

Зміст

Застосування середовища статистичних обчислень R до спектроскопічних даних Апунович С.Є., Апунович С.В., Дендебера М.П., Марковский А.А.....	3
Бестіарій Великих Даних, або про екологію проектів навколо Apache Hadoop Апунович С.Є.....	5
Використання контейнерної віртуалізації в організації навчального процесу студентів комп'ютерних спеціальностей Бойко Я. В.....	8
Досвід створення репозитарію Інституту післядипломної педагогічної освіти на основі вільно-поширюваного програмного забезпечення DSPACE Буган Ю.В., Олексюк О.Р.....	9
Використання мови R для ідентифікації параметрів ARFIMA-моделі Рабик В.Г., Болеста В.І., Червінський М.І.....	11
E-learning в ПТО, або не тільки Moodle-м живе сучасна освіта Чоповський С. С.....	14
Розпрацювання програмного забезпечення для генерації грануламетричного складу матеріялау на основі розміркування Вейбулла Дзівінец А.А., Дзераченнік С.С., Разумейчук В.С., Лапіч С.В....	18
VyOS introduction Дєєв К.....	22
Автоматизація роботи адміністратора онлайн-олімпіад на основі платформи EJUDGE Дуч М.С., Цимбал Ю.В.....	24
Програмне забезпечення для аналізу протоколів комп'ютерної мережі для ОС Ubuntu Федевич О.	27
Вплив похибки інтерполяції степеневих спектрів з нецілими показниками степеня на точність чисельного обчислення інтегрального перетворення Фур'є Флюнт О. Є.....	29
Створення інформаційно-аналітичної системи самооцінювання освітньої діяльності педагогічного університету Франчук В.М., Франчук Н.П.....	32
Побудова захищеної системи реалізації транзакцій з використанням бібліотеки JcardSim Гладкий Р.В, Желізняк Г.О.....	34
Вільнопоширюване програмне забезпечення в контексті варіативності моделей комп'ютерно-орієнтованого середовища навчання предметів природничо-математичного циклу Гриб'юк О.О., Хошаба О.М.....	36
Проектно-дослідницька діяльність в процесі навчання математики з використанням вільнопоширюваних програмних засобів Гриб'юк О.О., Юнчик В. Л.....	40
Особливості моделювання ринку програмного забезпечення Грицюк Ю.І., Грицюк П.Ю.....	44
Розробка мікросоціальної мережі з використанням мови програмування Python і програмного каркасу Django Ханас Д.....	49
До питання використання можливостей відкритої програмної бібліотеки компонентів при вивченні роботи із сокетом Харченко В.М., Харченко М.В.....	52
Системно-концептуальний підхід щодо побудови та функціонування комп'ютерно-орієнтованого середовища навчання Хошаба О.М., Гриб'юк О.О.....	54
Захист зображень у друкованих документах Кліц Ю.І.....	58
Статистичний пакет R Кобильник Т.П.....	61

Software system for parallel usability testing Kostiuk D.A., Latiy O.O., Markina A.A.....	63
Аб выкарыстанні мабільных платформаў у добраахвотных вылічэннях Каваленка У.Ю., Касцюк Д.А., Чацвѣркіна Г.А.....	67
Використання вільного програмного забезпечення у Львівському університеті імені Івана Франка Кухарський В., Злобін Г.Г., Рикалюк Р.Є., Шувар Р.....	71
Використання Qt Creator для побудови системи опитування Кушнір В.....	74
Використання вільно поширюваного програмного забезпечення при підготовці обдарованих дітей до олімпіад з комп'ютерної графіки та анімації Кузьменко А.....	76
Noise signal simulation with Octave Lapich S.V., Derechennik S.S., Menshih T.J., Divinets A.A.....	79
Використання rust для реалізації розширень інтерпретованих мов І. Лаврів.....	81
Використання вільного програмного та апаратного забезпечення для точної реєстрації часових міток подій. Мартинюк-Лотоцький К.П., Сергеев О.В.....	83
Месенджер для платформи Android на основі протоколу WebRTC Муха Б.М., Шпак З.Я.....	85
WxPython – вільна бібліотека для створення повнофункціональних кросплатформних графічних інтерфейсів мовою Python Миронюк Д.....	88
Досвід проектування складених мереж у академічній хмарі, розгорнутій на основі Apache CLOUDSTACK Олексюк В.П.....	90
Використання системи GEOGEBRA у процесі навчання математики в контексті впровадження ІКТ в освіту Рафальська М.В., Лященко Г.М.....	94
Можливості верстання математичних книжок за допомогою системи Luch Репецкий В.С.....	97
Універсальний конвертер на базі ОС Андроїд Рухолла О., Павлюк О.М.....	100
Вивчення природничих дисциплін з використанням PhET-моделювання Дмитрів М.В., Твердохліб І.А.....	103
Використання вільного програмного забезпечення для вивчення технічних дисциплін майбутніми учителями інформатики Ткачук Г.В.....	106
Створення електронних освітніх ресурсів засобами вільного програмного забезпечення Величко В.Є.....	109
Впровадження вільного програмного забезпечення у систему підвищення кваліфікації науково-педагогічних працівників як системна проблема Воронкін О. С.....	112
Розробка Android-додатків на Android Studio під Ubuntu на Kotlin Яхненко О.В.....	114
Розробка програмного забезпечення для оцінки та зміни параметрів якості растрових зображень Яворівський Б.....	119
Вольная альтэрнатыва Dropbox і сэрвісам GOOGLE Захарэвіч А.Л.....	122
Програмне забезпечення ІТ-компанії та фахова підготовка студентів напрямку “Комп’ютерні науки” на факультетах електроніки та прикладної математики ЛНУ ім. Івана Франка Злобін Г.Г., Скоропад О., Рикалюк Р.....	125

Застосування середовища статистичних обчислень R до спектроскопічних даних

Апуневич С.Є., Апуневич С.В., Дендебера М.П., Марковський А.А.

*Львівський національний університет імені Івана Франка,
н.сп. астрономічної обсерваторії, доцент кафедри експериментальної фізики,
студенти 4-го та 3-го курсу фізичного факультету,
вул. Кирила і Мефодія, 8, Львів 79005*

In this report we attempt to clarify how R statistical software can be applied to process and analyze the spectroscopical data. It appears that R Comprehensive Archive directly advertise quite a number of packages and tools for such application. We have tested one of them (prospectr) on Raman spectroscopy data obtained at Experimental Physics department. We find the usage of R very perspective for applying to real experiment data, as well as for learning students statistical methods.

Ускладнення експериментів, покращення точності та збільшення об'ємів даних, отриманих в експериментах, непрямий характер багатьох вимірювань кидають виклик фізикам-експериментаторам. Дедалі голосніше перед сучасним фізиком стає потреба в адекватному розумінні сучасних статистичних методів, в умінні їх застосувати до опрацювання даних, візуалізації даних та видобутку із них важливих фізичних результатів, обґрунтування їх статистичної значущості. Логічним у такому випадку виглядає не перевинайдення велосипеда, а використання наробків прикладної статистики у вигляді бібліотек до мов програмування або пакетів (середовищ).

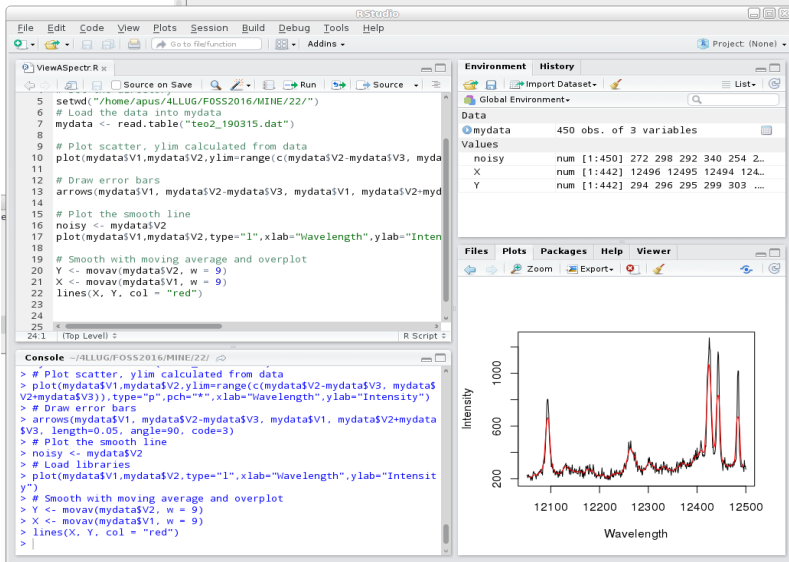
Однозначним лідером серед статистичних пакетів виглядає система (середовище) для статистичних обчислень R (R-project) [1]. Проект має тривалу історію понад 30 років, і, завдяки відкритості свого вихідного коду та потужній спільноті розробників, останнім часом переживає ренесанс не лише у середовищі професійних статистиків, але й серед спеціалістів інших галузей, від психологів до інженерів, у міру того як статистика дедалі більше входить у всі галузі життя.

До позитивних особливостей пакету R слід зарахувати: це вільне, відкрите ПЗ, один із проектів GNU [2]; це гнучка мова інтерактивних обчислень та програмування; це багатофункціональне середовище, не залишаючи котре можна виконати повний цикл роботи із даними; це архів-бібліотека CRAN (Comprehensive R Archive Network) [3] із понад 7000 пакетів; це сучасні засоби для високоякісної візуалізації даних, як інтерактивна так і для підготовки до друку; це векторизовані обчислення; це платформо-незалежність (є для Linux, Windows, MacOS); активна спільнота із професійних програмістів, академічних статистиків, активно просувається в Data Science; величезний обшир довідників/книжок/курсів/блогів.

Серед недоліків варто зауважити такі як: доволі складна і специфічна семантика функціонального програмування; деяка технологічна “несучасність”, мова мало змінилася за 30 років; доволі повільний інтерпретатор;

всі дані розміщуються в оперативній пам'яті. Здебільшого, ці недоліки або активно усуваються, або стають несуттєвими із зростанням потужності обчислювальної техніки.

Насправді, важко знайти статистичний метод чи алгоритм, який не втілено у пакеті R чи не опубліковано у вигляді рецептури або зразку. Навіть більше, R вже є дечим значно більшим ніж просто статистичним пакетом чи засобом візуалізації, він вже може потіснити MATLAB (Octave) як система наукових обчислень та моделювання.



Знімок екрану середовища розробки R-Studio із кодом, виконанням та графіком даних.

Отже, ми маємо такі завдання: є дані вимірювань; знайти що вже є в CRAN; адаптувати за мінімальної кількості кроків і застосувати для попередньої обробки даних.

Джерело [4] подає нам в одному місці аналіз наявних пакетів. Стисло подамо опис найбільш активних проектів: “hyperSpec” дає можливість візуалізувати/аналізувати гіперспектральні дані, тобто спектри із додатковою інформацією про розподіл у просторі, часі, концентрацію тощо; “ChemoSpec” є збіркою функцій для побудови графіків та загального аналізу спектральних даних; “prospectr” містить функції для попередньої обробки та формування вибірки для дифузних спектрів відбивання; “resemble” включає функції аналізу відмінностей, нелінійне моделювання спектральних даних; “TIMP” надає середовище розв’язування задач для апроксимації, розділення моделей, для часово-впорядкованих спектрів; “sraq” втілює засоби

калібровки спектрів Cluster-based Peak Alignment (CluPA); “Peaks” надає функції для маніпуляцій із спектром, перенесений із ROOT/TSpectrum.

Ми використали дані комбінаційного розсіяння сполуки TeO_2 , отримані за допомогою спектрографа ДФС-52, джерело – твердотільний лазер працював на довжині хвилі 532 нм. Ми скористались пакетом `prospectr` для згладжування даних та усунення шуму методом біжучого середнього, а також для побудови графіків.

Висновок: Пакет статистичних обчислень R має все необхідне для застосування до опрацювання даних спектроскопічних вимірювань, і ми вважаємо доцільним внести R разом із сучасними методами статистики до програми навчання фізиків-експериментаторів.

Джерела:

1. <http://www.r-project.org/>
2. GNU Project: <http://www.gnu.org>
3. Бібліотека CRAN: <http://cran.r-project.org/>
4. Journal of Statistical Software <http://www.jstatsoft.org/> January 2007, Volume 18, Issue 1. An Introduction to the Special Volume “Spectroscopy and Chemometrics in R”

Бестіарій Великих Даних, або про екологію проектів навколо

Apache Hadoop

Апуневич С.Є.

*Астрономічна обсерваторія Львівського національного університету
імені Івана Франка,
EPAM Systems,
вул. Кирила і Мефодія, 8, Львів 79005*

This short report is an attempt of holistic analysis of Hadoop framework for Big Data computations in terms of ecology and complexity, to expose the beauty and ugliness of this system, imagine its future.

Бестіарій [1] – це особливий жанр старовинної літератури, щедро ілюстровані збірки описів та оповідей, певною мірою предтечі енциклопедій. В них описували тварин, комах, потвор, і навіть камені, часто безоглядно до реальності описуваного, бо автори здебільшого самі не бачили цих сутностей і покладалися на непевні джерела. Бестіарії були популярні у Середньовіччі, коли різниця між реальністю та вигадкою була нечіткою, і ніхто не ставив під сумнів правдивість відомостей викладених у книжці, а істинність потверджувалася моральним висновком, що супроводжував кожен допис. До чого тут Apache Hadoop? Я висуваю тезу, що така ж суб'єктивність часто панує у сфері сучасних високих інформаційних технологій.

Apache Hadoop [2] – це каркас, що надає змогу розподілено опрацювати великі набори даних на кластерах комп'ютерів за допомогою простих моделей програмування. Він спроектований із метою масштабування від кількох серверів до тисяч обчислювальних машин, використовуючи кожен вузол для зберігання даних та обчислень; також у системі закладено стійкість до несправностей, що дозволяє знизити вимоги до надійності кожного окремого вузла. Очевидно, що ці властивості досягнуто за рахунок компромісу, звуження функціональності таких систем у порівнянні із класичними реляційними базами даними.

Можна стверджувати, що за 10 років існування проект виконав свої головні обіцянки. Однак, на основі цього проекту як платформи виникло багато, навіть дуже багато супутніх проектів, які взаємодіють нетривіальним чином між собою та із платформою. Дедалі частіше ці проекти мають на увазі компенсувати компроміси закладені у Hadoop під час проектування, та наблизити операції над великими даними для користувача до оперативності, цілісності, надійності “класичних” систем опрацювання та зберігання даних. Чи це можливо і яку ціну доведеться за це заплатити?

Проекти Apache Hbase, Cassandra, Hive, Spark, ZooKeeper, Drill, Impala, Oozie та інші переслідують різні цілі, часто конкурують між собою, деколи доповнюють одне одного. Кожен із них має дивну екзотичну назву та талісман часто у вигляді доволі дивних істот. Спроби оглядів всієї ніші продуктів для Big Data вже стануть бестіаріями, оскільки жоден із авторів вже не може мати досвід роботи із всіма продуктами. Тобто, очевидна головна ціна, яку сплатили – це складність.

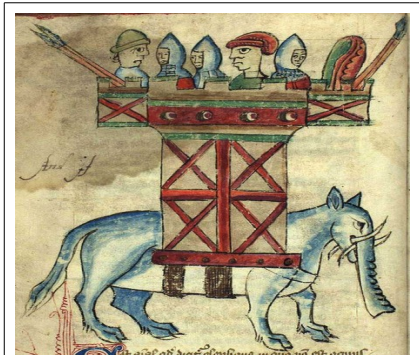
Який головний принцип інтеграції таких складних систем, як вони взагалі працюють? На згадку приходить “теорема” відомого науковця, винахідника ідеї підпрограми, Дейвіда Вілера,[3] яка звучить наступним чином:

“Всі проблеми у комп'ютерних науках можна розв'язати за допомогою ще одного рівня індирекції (непрямоті) ... окрім проблеми надто багатьох шарів індирекції.

Аналогічна думка закладена у Інтернет-меморандумі RFC 1925 [4]:

“...(6) Легше перемістити проблему деінде (наприклад, перемістити проблему в іншу частину загальної архітектури мережі), ніж її розв'язати.

(6а) (висновок). Завжди можна додати ще один рівень індирекції. ...”



*Зображення слона із бестіарія,
жовтий плюшевий слон є
символом проекту Hadoop*

Саме за таким принципом і надбудовується складна, багатоплатформова структура над низовою платформою Hadoop. Численні виклики програмних інтерфейсів (як через RPC, так і HTTP, або засоби серіалізації) пов'язують архітектуру до купи, хай і не надто елегантним чином.

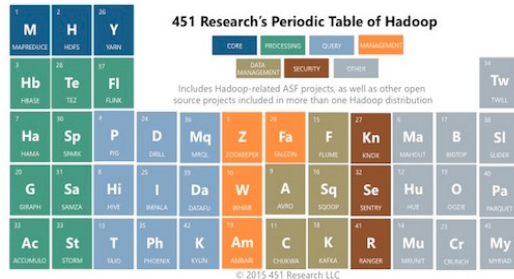
Інша аналогія, що напрошується для порівняння – це екологія в

первинному значенні цього слова, не як щось добре чи здорове у стосунку до людини, а як наука про виживання та пристосування в складних біологічних системах. Не слід забувати, що відкрите/вільне ПЗ – це світ жорсткої конкуренції. Очевидно, що світ Великих Даних – відомий як джерело надприбутків, і проекти Apache Software Foundation зазнають сильного впливу комерціалізації, коли великі гравці на ринку намагаються диктувати власні правила, часто виходячи із маркетингових, а не інженерних міркувань. Спостереження таке: екологічна ніша росте явно швидше, ніж заповнюється, отже боротьба за виживання грає у майбутньому. Мають вижити тільки ті проекти, хто запропонує найбільш ефективні рішення. Швидке зростання проявляється вже зараз через брак інтеграції, технічні вимоги до продуктів змінюються, а складність стає проблемою, яку намагаються вирішити за допомогою ускладнення системи.

А поки ми всі чекатимемо неможливої дії екологічних законів, вже зараз можна тішитися із того, що розподілені обчислення стали надзвичайно доступними для широких мас, а майбутнє їх спрощення тільки сприятиме поширенню високонавантажених обчислень.

Джерела:

1. <https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D1%81%D1%82%D1%96%D0%B0%D1%80%D1%96%D0%B9>
2. <http://hadoop.apache.org>
3. [https://en.wikipedia.org/wiki/David_Wheeler_\(British_computer_scientist\)](https://en.wikipedia.org/wiki/David_Wheeler_(British_computer_scientist))
4. <http://www.rfc-base.org/rfc-1925.html>



Проекти екологічної ніші, укладені в подоби періодичної таблиці.

Використання контейнерної віртуалізації в організації навчального процесу студентів комп'ютерних спеціальностей

Бойко Я. В.

Львівський національний університет імені Івана Франка, факультет електроніки, кафедра радіоелектронних і комп'ютерних систем, вул. Драгоманова, 50, Львів 79005, j_boyko@lnu.edu.ua

In the present report the features and benefits of using container virtualization in Linux operating system to organize the process of studying a number of courses by students of computer specialties were highlighted. The basic features of the LXD system and results of testing at a particular hardware and software environment were described.

У процесі вивчення низки дисциплін для студентів комп'ютерних спеціальностей виникає необхідність надання студентам повноцінного доступу до клієнтських і серверних систем із правами адміністратора з метою набуття ними стійких навичок адміністрування різного роду програмного забезпечення та сервісів. За стандартних підходів до організації робочих місць студентів забезпечення такої можливості пов'язане із проблемами безпеки та цілісності системи. Одним із можливих варіантів є поєднання систем віртуалізації із термінальним доступом до віртуальних машин. Проте використання з цією метою поширених гіпервізорів 2 типу надто не-ефективне через значні затрати серверних ресурсів. З огляду на вищезазначені проблеми нами було випробувано клієнт-серверну систему на основі контейнерних гіпервізорів типу LXC [1] -- системи віртуалізації на рівні операційної системи для запуску декількох ізольованих примірників ОС Linux на одному комп'ютері. LXC заснована на технології ядра Linux cgroups (control groups) та ізоляції на основі просторів імен. З міркувань зручності адміністрування нами використано надбудову над LXC — проект LXD (LXC Daemon) [2,3]. Це розробка фірми Canonical, що поширюється за ліцензією Apache 2. Вона складається з трьох компонентів: загальносистемний демон (LXD), клієнт командного рядка (LXC), плагін OpenStack Nova. Демон експортує REST API як локально, так і, якщо ця функція увімкнена, по мережі.

Процес встановлення LXD в дистрибутиві Ubuntu полягає у використанні штатних пакетних менеджерів. Базові образи операційних систем (різних дистрибутивів Linux) постачає ресурс images.linuxcontainers.org. Додавання репозиторію можна реалізувати командою

```
lxc remote add images images.linuxcontainers.org
```

Вивід переліку доступних дистрибутивів:

```
lxc image list images:
```


Копіювання образу на локальну машину і створення контейнера:

```
lxc image copy images:/ubuntu/trusty/amd64 local: --alias=trusty-  
amd64  
lxc launch trusty-amd64 container01
```

Таким способом на тестовій конфігурації (сервер на базі AMD Phenom 8650 X3, RAM 6 GB, HDD 1 TB, Kubuntu 14.04 LTS) було створено 10 контейнерів із 512 МБ оперативної пам'яті з можливістю доступу за протоколом SSH. Систему випробувано на лабораторних заняттях з курсів “Операційні системи”, “Адміністрування програмних систем і комплексів” та “Адміністрування ОС UNIX”, на яких студенти мали можливість працювати в індивідуальних середовищах в режимі адміністратора. Навіть в умовах обмежених обчислювальних і мережних ресурсів система демонструвала стабільну роботу без відчутних часових затримок. Таким чином, досвід використання контейнерної віртуалізації можна вважати позитивним, а сферу його застосування поширювати на інші навчальні дисципліни.

Джерела:

1. <https://uk.wikipedia.org/wiki/LXC>
2. <http://www.ubuntu.com/cloud/lxd>
3. <https://linuxcontainers.org/lxd/>

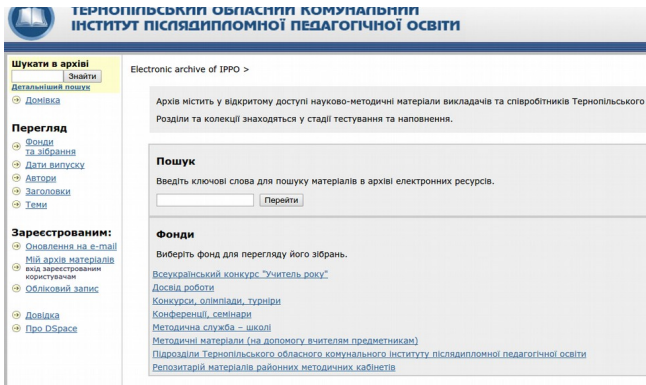
Досвід створення репозитарію Інституту післядипломної педагогічної освіти на основі вільно-поширюваного програмного забезпечення DSPACE

Буган Ю.В., Олексюк О.Р.

*Тернопільський обласний Інститут післядипломної педагогічної освіти,
yu.bugan@ippo.edu.te.ua, o.oleksyuk@ippo.edu.te.ua*

The main stages of implementing open electronic archive (repository) Ternopil Regional Institute of Postgraduate Education based on freely distributed software and open source (DSpace).

Упродовж останніх років в Інституті накопичилася велика кількість публікацій на CD-дисках. Проте значна частина розробок педагогів, цінний педагогічний досвід залишалися невідомими широкому колу учителів та науковців. Традиційні бібліотеки сьогодні не завжди можуть адекватно відповісти на виклики часу та ефективно організувати усі інформаційні джерела наявні у їх фондах. Перед нами виникла потреба створити систематизовану колекцію електронних документів, навчальних та методичних матеріалів та організувати різнорівневий доступ. Інституційні репозитарії допомагають успішно розв'язати цю проблему, тому сьогодні активно використовуються в навчальних закладах.



Мета створення ресурсу конкретизована у завданнях:

- надійне зберігання навчальних, методичних, наукових матеріалів педагогів області;
- поширення та популяризація передового педагогічного досвіду;
- створення середовищ для публікування навчальних, методичних, наукових матеріалів педагогів;
- сприяння інтеграції цих матеріалів в науково-інформаційний простір області.

Різноманітність систем для розгортання репозитаріїв, зумовили необхідність порівняльного аналізу відповідних програмних продуктів. Критеріями добору були: відповідність функціональних характеристик, потребам користувачів; умови поширення систем; кількість успішних інсталяцій; географія застосування; багатомовність та можливість локалізації; для вільно поширюваних систем важлива підтримка співтовариства розробників та якість документації.

Нині в усьому світі чимало компаній і організацій розробляють програмне забезпечення для розгортання інституційних репозитаріїв. За даними реєстру репозитаріїв OpenDoar [2] знаходимо понад 100 одиниць таких платформ. Комерційні програми досить вартісні, тому для розгортання репозитарію інституту обрана найпопулярніша у всьому світі і в Україні вільно поширювана система DSpace. Серед її переваг виділимо безкоштовність, кросплатформність, підтримку стандартизованого формату метаданих (Дублінське ядро), можливість обміну метаданими за протоколом OAI-PMH, індексування матеріалів популярними пошуковими системами тощо. Система має активну підтримку зі сторони постійно зростаючого співтовариства розробників. У липні 2015 року опублікована версія DSpace 5.5. Програмне забезпечення розробники постійно удосконалюють та розвивають.

Система Dspace [1] дає можливість створити окремі сторінки для кожної колекції електронного архіву. Крім цього, доцільно налаштувати колекції у

такий спосіб, щоб у разі завантаження матеріалів деякі типи метаданих були визначені за замовчуванням для даної колекції. Таке конфігурування сприяє зменшенню помилок під час додавання матеріалів, а отже підвищує якість матеріалів та заощаджує час технічних редакторів колекції.

Значну увагу було приділено спільно з працівниками бібліотеки до підготовки метаданих матеріалів. У системі додали нові поля, специфічні для опису навчальних ресурсів навчального процесу. Наприклад, такими полями були: предмет, клас для якого підготовлено ресурс. Відповідно до нових метаданих змінено поля, за якими буде здійснюватися розширений пошук у репозитарії.

Для науково-педагогічних працівників інституту та методистів-завідувачів районними методичними проведено ряд тренінгів та консультацій щодо самостійного розміщення публікацій у репозитарій.

Щодня репозитарій поповнюється новими матеріалами, на сьогодні містить 3100 документів. Серед них – статті та презентації працівників інституту та розробки педагогів області

Література:

1. DSpace– Available from: <http://www.dspace.org/>
2. The Directory Of Open Access Repositories – Opendoar / University Of Nottingham. – Available from: <http://www.openoar.org/>.

Використання мови R для ідентифікації параметрів ARFIMA-моделі

Рабик В.Г., Болеста В.І., Червінський М.І.

*Факультет електроніки Львівського національного університету
імені Івана Франка, RabykV@ukr.net*

This work is devoted to the identification of the parameters of ARFIMA(p,d,q) time series model. The algorithm of the parameters estimation using R programming language functions from fracdiff and forecast packages is discussed in details.

У разі аналізу часових рядів зі складною структурою часто використовують ARIMA(p,d,q) моделі. З їх допомогою моделюються різні випадки, які зустрічаються при аналізі стаціонарних і нестаціонарних рядів. При цьому першим кроком є, як правило, визначення порядку інтегрованості ряду. Переважно при цьому обмежуються вибором між d рівним 0 і 1. Випадок d=0 відповідає короткій пам'яті ряду, тоді як d=1 – нескінченній пам'яті. При цьому не враховується проміжна ситуація, коли пам'ять процесу є довгою. Для вирішення цієї проблеми в роботі [1] було запропоновано новий клас ARFIMA(p,d,q)-моделей, які допускають можливість дробового параметру d.

Процес $\{X_t\}$ описується ARFIMA(p,d,q)-моделлю

$$\varphi(L)\Delta^d(x_t) = \Theta(L)\varepsilon_t, \quad (1)$$

де L – оператор зсуву, $\Delta = (1 - L)$, p і q – степені поліномів відповідно авторегресивної моделі (AR) і моделі ковзного середнього (MA), d – параметр фрактальної інтегрованості,

$$\Phi(L) = 1 - \varphi_1 * L - \varphi_2 * L^2 - \dots - \varphi_p * L^p, \quad (2)$$

$$\Theta(L) = 1 + \theta_1 * L + \theta_2 * L^2 + \dots + \theta_q * L^q, \quad (3)$$

ε_t – білий шум з нульовим середнім значенням і деякою постійною дисперсією σ^2 ,

$$\Delta^d = (1 - L)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k L^k = \sum_{k=0}^{\infty} \frac{\Gamma(k-d)}{\Gamma(-d)\Gamma(k+1)} L^k, \quad (4)$$

де $-0,5 < d < 0,5$ і $\Gamma()$ – гамма-функція.

При значеннях параметру $|d| < 0,5$ ARFIMA(p,d,q) процес є стаціонарним. При $0 < d < 0,5$ вважається, що процес має «довгу пам'ять», а при $-0,5 < d < 0$ – «коротку пам'ять».

ARFIMA(p,d,q) є зручним інструментом аналізу часових рядів, оскільки дає можливість одночасного моделювання ефектів довгої та короткої пам'яті.

Основні кроки алгоритму оцінки параметрів ARFIMA(p,d,q)-моделі [2]:

1. Оцінка параметру d ARFIMA(p,d,q)-моделі (\hat{d}).
2. Обчислення $\hat{u}_t = (1 - L)^{\hat{d}} x_t$, де $u_t \in \text{ARMA}(p, q)$ процесом.
3. Оцінка параметрів p, q (\hat{p}, \hat{q}) і $\varphi, \Theta(\hat{\varphi}, \hat{\Theta})$ ARMA(p,q) процесу $\Phi(L)\hat{u}_t = \Theta(L)\varepsilon_t$.
4. Обчислення $\hat{y}_t = \frac{\hat{\Phi}(L)}{\hat{\Theta}(L)} (1 - L)^{\hat{d}} x_t$.
5. Оцінка параметру d ARFIMA(0, d, 0)-моделі $(1 - L)^d \hat{y}_t = \varepsilon_t$.
6. Значення параметру d , отримане на цьому кроці є новою оцінкою. Повторення кроків 2-5 виконується до тих пір, поки параметри \hat{d} , $\hat{\varphi}$ і $\hat{\Theta}$ не будуть отримані із заданою точністю.

Для оцінки параметру d ARFIMA(p,d,q)-моделі мовою R використовувалися функції `fdGPH` і `fdSperio`, які входять до складу пакету `fracdiff` [3]. Функція `fdGPH` оцінює параметр методом GPH. Звертання до функції має вигляд:

`dfGPH(xt, bandw.exp=0.5)`,
де $\{xt\}$ – часовий ряд, а `bandw.exp` – діапазон частот, який використовується в рівнянні регресії. Діапазон частот обчислюється як `bw=trunc(n^bandw.exp)`, де $0 < \text{bandw.exp} < 1$ і n – розмір вибірки часового ряду. Функція також знаходить асимптотичне стандартне відхилення і стандартне відхилення похибки оцінки параметра d .

Функція `fdSperio` використовує для оцінки параметру d ARFIMA(p, d, q)-моделі метод Reisen'a. Він базується на рівнянні регресії з використанням згладженої функції періодограми для оцінки спектральної густини. Звертання до цієї функції:

```
fdSperio(xt, bandw.exp = 0.5, beta = 0.9),
```

де $\{xt\}$ – часовий ряд, а `bandw.exp` – діапазон частот, який використовується в рівнянні регресії, а `beta` – діапазон частот, який використовується в вікні затримки Reisen'a. Для знаходження діапазонів частот використовуються вирази $bw = \text{trunc}(n^{\wedge} \text{bandw.exp})$, де $0 < \text{bandw.exp} < 1$ і $bw2 = \text{trunc}(n^{\wedge} \text{beta})$, $0 < \text{beta} < 1$, n – розмір вибірки часового ряду.

Для автоматичного вибору оптимальної ARFIMA(p, d, q)-моделі і оцінки її параметрів використовується функція `arfima` пакету `forecast` [4]. Звертання до функції `arfima`:

```
arfima(x, drange=c(0, 0.5), estim=c("mle", "ls"), lambda=NULL, ...),
```

де $\{xt\}$ – одномірний часовий ряд, `drange` – допустимий діапазон значення d . (за замовчуванням величина d визначається в діапазоні $(0, 0.5)$, що забезпечує стаціонарну ARFIMA(p, d, q)-модель); `estim` – якщо `estim=="ls"`, то тоді параметри ARMA розраховуються за допомогою алгоритму Haslett і Raftery, якщо ж `estim=="mle"`, то тоді параметри ARMA уточнюються з допомогою функції `arima`; `lambda` – параметр трансформації Вох-Сох, який ігнорується, якщо задано `NULL`.

Мовою програмування R реалізована програма `TS_ARFIMA`, яка дає змогу ідентифікувати параметри ARFIMA(p, d, q)-моделі. Програма використовує функції пакетів `fracdiff` і `forecast`. Для її тестування використовувалося моделювання часового ряду у вигляді ARFIMA-моделі з заданими параметрами p , d , q і коефіцієнтами φ , θ з допомогою функції `arfima.sim` пакету `arfima` [5].

Джерела:

1. Hosking J. R. M. Fractional Differencing. *Biometrika*, Vol. 68, N. 1, 1981, P. 165-176. [Електронний ресурс]. Режим доступу: <http://www.imperial.ac.uk/~ejm/M3S8/Problems/hosking81.pdf>
2. Reisen V., Abraham B., Lopes S. Estimation of Parameters in ARFIMA Processes: A Simulation Study. [Електронний ресурс]. Режим доступу: <http://mat.ufrgs.br/~slopes/artigos/verfinal.pdf>
3. Package 'fracdiff'. [Електронний ресурс]. Режим доступу: <https://cran.r-project.org/web/packages/fracdiff/fracdiff.pdf>
4. Package 'forecast'. [Електронний ресурс]. Режим доступу: <https://cran.r-project.org/web/packages/forecast/forecast.pdf>
5. Package 'arfima'. [Електронний ресурс]. Режим доступу: <https://cran.r-project.org/web/packages/arfima/arfima.pdf>

E-learning в ПТО, або не тільки Moodle-м живе сучасна освіта **Чоповський С. С.**

Державний навчальний заклад “Львівський професійний ліцей залізничного транспорту” auslemborg@meta.ua

З розвитком інформаційних технологій виникають і нові форми навчання – дистанційні, які стали вже відомими і необхідними у сучасному навчальному процесі не тільки серед ВНЗ, но й в ПТНЗ і школі. Існуючий термін E-learning визначає, що електронне навчання – це передача знань та управління процесом навчання за допомогою нових інформаційних і телекомунікаційних технологій. У E-learning використовуються інтерактивні електронні засоби доставки інформації Інтернет та інтранет, а також флеш-карти та компакт-диски. На теперішній час можна виділити основні засоби організації E-learning:

- системи управління контентом (Content Management Systems - CMS),
- системи управління навчанням (Learning Management Systems - LMS),
- системи управління навчальним контентом (Learning Content Management Systems - LCMS)

Повномасштабна система електронного навчання, вже традиційно, має три стандартні модулі: 1) Система управління навчанням (LMS - learning management system); 2) Навчальний контент (наповнення електронних курсів); 3) Авторські засоби створення і редагування навчального контенту (authoring tools). В галузі розроблені спеціальні стандарти технічних специфікацій для створення навчального Web-контенту. Стандарт SCORM вважається більш сучасною версією стандартів обміну навчальними матеріалами, побудованими на інших технологічних принципах, і використовується у більшості LMS (LCMS). Приклади створення сучасних LMS: IBM Lotus Workplace Collaborative Learning (LWCL); learnXact; Elearning Now та Manage Now; Sitos, компанія Bitmedia; Competentum; RedClass; eLearningServer; Naumen і.т.п. Всіх їх об'єднує одне — вони пропонують електронні курси власного виробництва. В першу чергу, ці курси розраховані на сферу корпоративного навчання та управління знаннями працівників в області бізнесу та інформаційних технологій. Можуть бути застосованими у приватних навчальних закладах та тренінгових центрах. Навчальний контент – це електронні курси, з допомогою яких навчаються учні. Авторські засоби (authoring tools) – це засоби розробки навчального контенту. Існує кілька різновидів авторських засобів: 1) Редактори навчальних курсів; 2) Засоби для створення презентацій; 3) Засоби для створення тестів і анкет; 4) Засоби для захоплення зображення з монітора; 5) Засоби для проведення онлайн-семінарів.

В основі LCMS лежить концепція подання змісту навчання як сукупності багаторазово використовуваних навчальних об'єктів зі своєю цільовою аудиторією і певним контекстом використання. LCMS повинна містити

наступні ключові компоненти: -Репозиторій навчальних об'єктів; -Програмне забезпечення автоматизованого ауторинга; -Інтерфейс відображення (прогресування контенту); -Засоби адміністрування.

На основі аналізу існуючих систем LMS/LCMS можуть бути виділені наступні найбільш популярні системи дистанційного навчання: ATutor, Claroline LMS, Dokeos, eFront, ILIAS, Moodle, OLAT, Open Elms, OpenACS, Sakai, TrainingWare Class, WebTutor. Основними критеріями відбору були ступінь підтримки системи і багатомовний супровід. Коротко розглянемо деякі з них. Atutor (<http://www.atutor.ca/>) є вільно поширюваною web-орієнтовану систему управління навчальним контентом, розробленою з урахуванням ідеї доступності та адаптованості. Адміністратори можуть оновити або інсталювати Atutor за кілька хвилин, розробити власні шаблони оформлення системи. Викладачі можуть швидко збирати, структурувати зміст навчального матеріалу для проведення занять on-line. Учні працюють з гнучким, адаптивним середовищем навчання.

Claroline LMS (<http://www.claroline.net/>) (Classroom Online) – платформа побудови СДН, створена з урахуванням побажань викладачів, має ліцензію GNU/GPL. Claroline дає змогу створювати уроки, редагувати їх вміст, керувати ними. Система містить генератор вікторин, форуми, календар, функцію розмежування доступу до документів, каталог посилань, систему контролю за успіхами учня, модуль авторизації.

Dokeos (<http://www.dokeos.com/>) – платформа побудови СДН, заснована на гілці (fork) Claroline (версії 1.4.2.) та підійде скоріше організаціям, ніж університетам, орієнтована на професійну клієнтуру (персонал підприємства). Dokeos має ліцензію GNU/GPL.

eFront LMS – має ліцензію GNU/GPL Безсумнівно, ця одна з найбільш якісних вільних систем, що заслуговують на увагу. eFront має весь основний необхідний функціонал, і навіть генерацію сертифікатів, одна з небагатьох систем, в якій стандарт SCORM має сертифіковану підтримку. Для функціонування системи, потрібна база даних MySQL, а також сервер з підтримкою PHP. Це може бути Apache або IIS. Якщо у разі встановлення більшості вільних систем необхідні відповідні технічні навички, то eFront в цьому сенсі значно відрізняється в кращу сторону. Система цілком вдало встановлюється і в Linux, і на CentOS, і на Windows. Причому всі процеси детально задокументовані і описані кирилицею. Тобто, у будь-якого користувача, навіть технічно слабко підготовленого, навряд чи виникнуть які-небудь проблеми під час встановлення. До системи можна під'єднати вебінари використовуючи відкриту платформу bigbluebutton. Для учня система виглядає дуже просто і звично як будь-яка соціальна мережа. Система eFront, в першу чергу, призначена для академічного сектору. Однак є і комерційні локалізації, які спеціально розроблені для організації, там вже наявний такий функціонал, як управління компетенціями та навичками, облік штатного розкладу, відстеження переміщення по службі тощо. Система

повністю кирилізована включно з документацією. В цілому, eFront LMS – досить якісна і функціональна система, яка, безсумнівно, заслуговує на велику увагу. Офіційний сайт проекту: www.efrontlearning.net Завантажити систему можна звідси: www.efrontlearning.net/download .

ILIAS (від скорочення Integriertes Lern-, Informations — und Arbeitskooperations-System) – система має ліцензію GNU GPL, з дуже широким функціоналом, підтримуються основні міжнародні стандарти (SCORM 1.2, SCORM 2004, AICC). Є конструктор тестів, підтримує різні типи питань. Також є можливість імпорту в систему зовнішніх тестів або їх експорту в форматі IMS QTI. ILIAS потребує сервера Apache з підтримкою PHP і бази даних MySQL. Є переклад на багато мов світу. Більш детально познайомитися з можливостями системи і завантажити дистрибутив можна з офіційного сайту: www.ilias.de Спробувати систему в демо-режимі можна там же: www.demo.ilias.de .

Moodle Про систему Moodle говорять дуже давно і багато, тому не будемо повторюватись, але зазначимо що система підтримує показ будь-якого електронного формату документів, що є корисним у разі створення курсів. До недоліків системи можна віднести певну складність встановлення та налаштування, монстроподібність. Так, наприклад, Moodle є здебільша тільки системою дистанційного навчання. У той же час, більшість сучасних LMS/LCMS вже давно вийшли за рамки функціоналу дистанційного навчання, маючи у своєму арсеналі безліч інших різноманітних модулів. Перевагою системи Moodle є її популярність – це, мабуть, найбільш популярна вільна система дистанційного навчання на сьогоднішній день. Серед аналогічного програмного забезпечення, у Moodle найбільша спільнота користувачів, а отже і можливостей знайти помічників, радників, соратників. З десяти найпопулярніших вишів України, які мають сайти дистанційного навчання, саме базуються на системі Moodle. Офіційний сайт продукту Moodle: www.moodle.org (тут можна завантажити програмне забезпечення, завантажити додаткові компоненти, отримати документацію тощо).

OLAT (скорочення від Open Learning And Training) – ще одна вільна система дистанційного навчання з відкритим вихідним кодом. На сьогоднішній день система підтримує всі основні міжнародний стандарти, такі як SCORM та IMS QTI, а також дає змогу розміщувати всі види медіа-контенту. Функціонал системи досить великий і містить всі основні інструменти, необхідні для такого програмного забезпечення. Система повністю кирилізована і підтримує кодування UTF-8. Офіційний сайт проекту, де також можна завантажити саму систему: www.olat.org Спробувати демо-версію на сайті виробника: www.demo.olat.org

OpenACS (<http://openacs.org>) Open Architecture Community System - є основою для багатьох компаній і університетів, що займаються використанням технологій електронного навчання. Має потужний інструментарій для побудови масштабованих, спільнота-орієнтованих веб-

додатків. OpenACS доступна на умовах GNU GPL, працює на AOLserver і Naviserver та використовує Oracle або PostgreSQL реляційних баз даних. Система повністю кирилізована.

Sakai (<http://sakaiproject.org/>) є одна з найпопулярніших СДН на умовах GNU GPL, яка підтримується спільнотою розробників. Інтегрована підтримка стандартів та специфікацій IMS, SCORM. Ця система повністю написана мовою Java, що робить її дуже надійною, а головне, кросплатформною. Sakai підтримує роботу з різними базами даних – (вбудовану базу даних, MySQL або Oracle). Якщо якогось функціоналу буде недостатньо, його завжди можна додати. Зробити це можна власними силами або із залученням зовнішніх розробників. Добре, що на теренах СНД є розробники, які можуть дописати потрібний функціонал в системі Sakai, а також зайнятися її кастомізацією. В першу чергу Sakai призначена для академічного сектору. В цілому, Sakai – досить якісна і функціональна система, яка, безсумнівно, заслуговує на увагу.

TrainingWare Class (<http://www.ksob.ru/kso>) – це перша СДН з відкритим кодом від компанії "Корпоративні Системи Навчання" (BCC Group). TWClass дає змогу формувати індивідуальний підхід до навчання та автоматизувати рутинну роботу вчителя, і на сьогоднішній день є єдиною СДН, що розроблена, на відміну від зарубіжних аналогів, з урахуванням вимог освіти на теренах СНД. Створені з допомогою TrainingWare навчальні центри забезпечують весь цикл дистанційного навчання.

WebTutor – найпопулярніша система дистанційного навчання від компанії ВебСофт, з недавнього часу стала доступна для безкоштовного використання на теренах СНГ. Система Webtutor – це більше, ніж просто система дистанційного навчання. Ця система призначена для автоматизації більшості HR-процесів компанії. Система WebTutor має досить потужну вбудовану систему управління сайтами. Зрозуміло, безкоштовна версія системи WebTutor має ряд обмежень на відміну від комерційної, оскільки безкоштовна версія призначена в першу чергу для малого і середнього бізнесу. У будь-якому разі, той факт, що одна з найпопулярніших і функціональних систем такого рівня зараз доступна для безкоштовного використання (хоч і з обмеженнями), безсумнівно, величезний плюс. Заповіти безкоштовну версію WebTutor можна тут: www.kursor.com.ru/webtutor.

Висновок: Для забезпечення вдалого проектування, розвитку та впровадження СДН, необхідно створити інфраструктуру відповідних освітніх закладів, провести підготовку викладачів та розробити відповідні навчальні програми. Важливою складовою частиною дистанційного навчання є його реалізація за допомогою використання інформаційних технологій, які створені для розробки, управління та поширення навчальних матеріалів онлайн із забезпеченням спільного доступу багатьох користувачів. Система e-Learning, в ідеалі, повинна надавати кожному учню (студенту) персональні можливості для найбільш ефективного вивчення матеріалу. Учень отримує

можливості доступу до навчального порталу, вибору відповідних навчальних треків на основі попереднього і проміжних тестувань, використання додаткових матеріалів з допомогою спеціальних посилань. Перехід створення E-learning систем управління навчанням на принцип open source не тільки дає змогу знизити сукупні витрати навчальних закладів на володіння системами автоматизації навчання, але й надає можливість фахівцям навчальних закладів самостійно розвивати систему, швидко і з мінімальними витратами адаптувати її під постійні зміни процесу навчання, впроваджувати і підтримувати за рахунок перерозподілу ресурсів, а не шляхом створення нових посад, що вкрай важливо в сьогоdnішній ситуації, що відрізняється високим ступенем мінливості в економіці і низькою передбачуваністю. Кожна з перерахованих систем E-learning є досить якісними і функціональними системами, які, безсумнівно, заслуговують на повагу, вивчення та використання для організації сучасного навчального процесу.

Література:

1. Наказ Міністерства освіти і науки України “Про затвердження Положення про дистанційне навчання” // Режим доступу: <http://zakon5.rada.gov.ua/laws/show/z0703-13>
2. eFront // Режим доступу: <http://www.free-elearning.ru/tags/eFront/>
3. П.Безяев. Выбор бесплатной LMS или «Почему не moodle?» // Режим доступу: <http://i-elearning.ru/wordpress/vybor-besplatnoj-lms-ili-pochemu-ne-moodle.html>
4. Бекеш Ю.Р., Матієшин Л.М., Серов Ю.О., 2013 “Огляд систем дистанційного навчання популярних ВНЗ України”, Національний університет “Львівська політехніка”, УДК 378.14.004, С. // Режим доступу: <http://ena.lp.edu.ua:8080/bitstream/ntb/22735/1/11-44-48.pdf>
4. Б. Демида, С. Сагайдак, І. Копил, Національний університет “Львівська політехніка”, кафедра автоматизованих систем управління СДН: огляд, аналіз, вибір. , УДК 621.391, 2011 С. // Режим доступу: <http://ena.lp.edu.ua:8080/bitstream/ntb/10662/1/14.pdf>

Распрацоўка праграмнага забеспячэння для генерацыі грануламетрычнага складу матэрыялаў на аснове размеркавання Вейбулла

Дзівінец А.А., Дзерачэннік С.С., Разумейчык В.С., Лапіч С.В.

Брэсцкі дзяржаўны тэхнічны універсітэт

A GPL licensed granulometrics modeling project targeted at building industry is presented. Module to generate material composition based on Weibull distribution is discussed.

Рознага роду дысперсныя матэрыялы шырока выкарыстоўваюцца ў разнастайных галінах прамысловасці. Пры гэтым падобныя матэрыялы ўяўляюць сабой прыватны выпадак гетэрагеннай сістэмы – гэта значыць, яны з’яўляюцца аб’ектамі, якія складаюцца з часціц двух або больш розных тыпаў

(двух або больш фаз), што запаўняюць агульную сераду. У выніку адна з састаўных частак такога аб'ект фаз ўтварае бесперапынную дысперсійную сераду, у аб'ёме якой размеркавана дысперсная фаза (або некалькі фаз) у выглядзе дробных крышталёў, цвёрдых часціц, бурбалак і інш. [1].

Распаўсюджаныя спосабы даследавання дысперсных матэрыялаў грунтуюцца ў асноўным на эмпірычных вымярэннях і не прадстаўляюць магчымасці прааналізаваць уплыў параметраў іх складу і структуры на агульныя якасці матэрыялу (напрыклад, у мэтах аптымізацыі складу). Уплыў складу матэрыялу на яго структуру і, апасродкавана, на якасці - важная і да гэтага часу не да канца даследаваная задача матэрыялазнаўства [2].

Камп'ютэрнае мадэляванне істотна зніжае выдаткі часу і сродкаў на стварэнне вопытных узораў, у параўнанні з падборам аптымальнага складу шляхам эксперыментальнага змешвання кампанентаў, выключаючы на ранняй стадыі варыянты з «сумніўнымі» ўласцівасцямі.

Адным з ключавых элементаў пабудовы мадэляў дысперсных сістэм з'яўляецца выбар функцыі размеркавання памераў часціц пры драбненні [3]. Ён залежыць ад вялікай колькасці фактараў, звязаных з метадам асколкаўтварэння, выкарыстаным пры падрыхтоўцы кампанентаў сумесі. Найбольш папулярна ўжывенне здрабняючых матэрыялаў у рамках лагарыфмічнага або логнармальнага закона размеркавання часціц [4]. Аднак гэты падыход далёка не адзіны, а ў многіх выпадках яшчэ і не занадта дакладны; таму ў канкрэтных задачах часта альтэрнатыўныя падыходы даюць лепшы вынік.

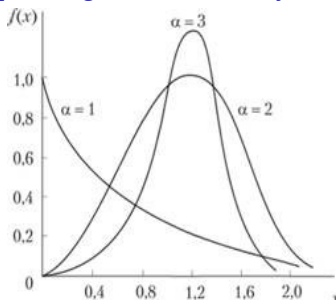
Для апісання грануламетрычнага складу дысперснага матэрыялу даволі часта выкарыстоўваюць размеркаванне Вейбулла. Упершыню эксперыментальная праверка прымянімасці дадзенага выгляду размеркавання была выканана Л.І. Баронам [5]. Шчыльнасць і інтэгральная функцыя размеркавання Вейбулла ў галіне фізічнага вызначэння (неадмоўнага аргументу) маюць выгляд:

$$\begin{cases} f(x) = \alpha \beta x^{\beta-1} \exp(-\alpha x^\beta); \\ F(x) = 1 - \exp(-\alpha x^\beta). \end{cases}$$

Вялікай перавагай такога размеркавання з'яўляецца тое, што яно вельмі шырока і ўтрымлівае ў сабе, як прыватныя выпадкі, экспанентнае размеркаванне і размеркаванне Рэлея, а таксама блізка да гама-размеркавання і логнармальнага (мал. 1). Пры пэўных значэннях параметраў размеркаванне Вейбулла становіцца нармальным размеркаваннем.

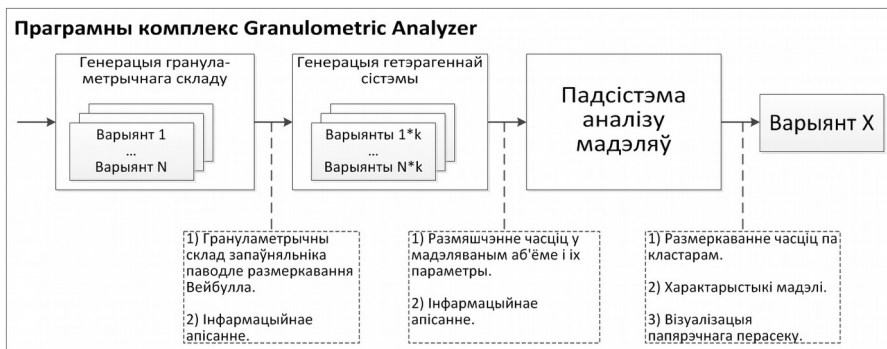
Рынак праграмнага забеспячэння для пабудовы і аналізу геаметрычных дысперсных мадэляў уяўляе сабой практычна абсалютны вакуум. У ходзе выкананага раней пошуку [6] нам не ўдалося знайсці ні адну спецыялізаваную сістэму, прызначаную для гэтых мэтаў; магчымасці універсальных пакетаў-монстраў, якія прадстаўляюць напрацоўкі на ўсе выпадкі жыцця, таксама нельга назваць арыентаванымі на дадзеную

канкрэтную задачу. Акрамя таго, у апошні час імкліваю папулярнасць набіраюць Open Source праекты [7]. У выніку быў пачаты праект Granulometric Analyzer, які распрацоўваецца ў асяроддзі Monodevelop, код якога распаўсюджваецца пад ліцэнзіяй GPL і даступны па адрасе <https://github.com/alexcapricorn/granulometricanalyzer>.



Малюнак 1 – Прыклады размеркавання Вейбула

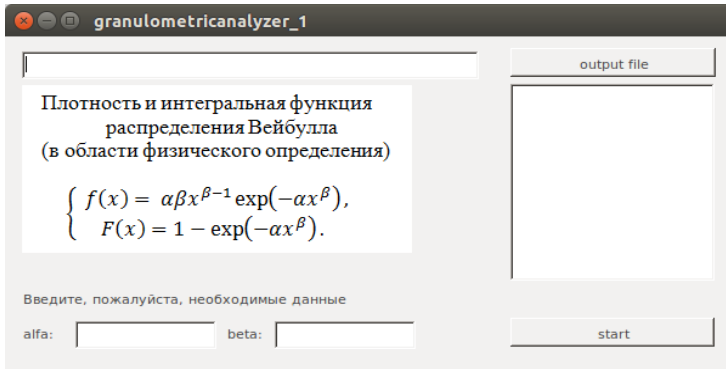
У сукупнасці, праект уключае тры праграмных модуля (мал. 2): першы для атрымання размеркавання грануламетрычнага складу дысперснага матэрыялу (графічны інтэрфейс прадстаўлены на мал. 3), другі для генерацыі структуры гетэрагеннай сістэмы на аснове стыхастычных метадаў, трэці - для вызначэння інтэгральных характарыстык мадэльнага аб'ёму. Сувязь паміж модулямі ажыццяўляецца праз файлы. Праца другога і трэцяга модуляў была апісана ў [6], у той час як грануламетрычны склад ў папярэдніх версіях праекта жорстка задаваўся таблічнымі значэннямі (меркавана змадэляванымі альбо запазычанымі з даведчнай літаратуры). Аднак пры выкарыстанні праекта ўзнікла неабходнасць аўтаматызаваць гэтую частку працы – напрыклад, для масавай генерацыі варыянтаў складу, нацэленай на атрыманне зададзеных уласцівасцяў метадам перабору варыянтаў.



Малюнак 2 – Канцэпцыя мадэлявання

Статыстычнай праверкай гіпотэз было ўстаноўлена, што размеркаванне

Вейбулла забезпечує добру збіжність з експериментальними даними або розмірним (грануламетричним) складом різних заповнювачів [8]. Тому, навіть дані типу розмірвання і будуть абрані для генерації грануламетричного складу дисперсних матеріалів.



Малюнок 3 - Графічний інтерфейс першого модуля

Розроблені програмні модулі складаються з декількох функціональних часток. Вхідними даними з'являються: β - так звані параметр форми, які характеризують здольність дисперсної системи до дроблення; α - параметр масштабу, які показує ступінь дроблення того чи іншого матеріалу. Крім того, для коректної роботи необхідно вказати шлях вихідного файлу. У випадку успішного виконання програми будуть створені: файл з грануламетричним складом заповнювача і інформаційне описання.

Після завершення роботи з документом з інформаційним описанням можна ознайомитися з основними етапами роботи модуля, а також дивитися або магчимо пам'ятати. Файл з грануламетричним складом заповнювача служить у якості одного з вхідних даних для наступної підсистеми.

Як згадувалося вище, чакана вобласть застосування проекту Granulometric Analyzer зв'язана перш за все з будівничою галузю. Наприклад, така розробка може бути використана при прогнозуванні тривалості і довговічності бетону, а також для оптимізації складу бетонної суміші.

Список литературы:

1. Дивинец, А.А. Имитационное моделирование транзитной зоны в гетерогенных системах / А.А. Дивинец // Сборник тезисов научной студенческой конференции «Неделя науки – 2014». – Брест, БрГТУ, 2014. – С. 62–63.
2. Разумейчик, В.С. Структурно-химическое моделирование гидратации цементного камня / В.С.Разумейчик // Вестник БрГТУ. – 2006. – №1(38): Строительство и архитектура. – С. 91-96.
3. Большая техническая энциклопедия [Электронный ресурс] / Технический словарь Том VII. – М., 2015. – Режим доступа: <http://www.ai08.org/index.php/term/7->

tehnicheskii-slovar-tom-vii,6481-raspredelenie-razmer-chastitsa.xhtml. – Дата доступу: 10.07.2015.

4. Большая техническая энциклопедия [Электронный ресурс] / Технический словарь Том VII. – М., 2015. – Режим доступа: <http://www.ai08.org/index.php/term/7-tehnicheskii-slovar-tom-vii,6481-raspredelenie-razmer-chastitsa.xhtml>. – Дата доступу: 10.07.2015.

5. Барон, Л.И., Сиротюк Г.М. Проверка применимости уравнения Розина-Рамлера для исчисления диаметра среднего куса при взрывной отбойке горных пород / Л.И. Барон, Г.М. Сиротюк // Взрывное дело. – М.: Недра, 1978.

6. Дзівінець, А.А. Распрацоўка праграмага забеспячэння для мадэлявання дысперсных матэрыялаў / А.А. Дзівінець, С.С. Дзерачэннік, В.С. Разумейчык // П'ята навуково–практична конференція FOSS LVIV 2015: Збірник наукових праць, Львів, 23–26 квітня 2015 р. – Львів: Львівський національний університет імені Івана Франка, 2015. – С. 23–26.

7. Костюк, Д.А. Порівняльний аналіз використання ВПЗ у вищих закладах освіти Білорусі, Російської Федерації та України / Д. Костюк, Г. Злобін, Е. Алексеев // Електроніка та інформаційні технології. – 2012. – Випуск 2. – С. 197-205.

8. Дивинец, А.А. Выбор вероятностного закона распределения для модельного описания дисперсности заполнителя бетонного композита / А.А. Дивинец, С.С. Дереченник, В.С. Разумейчик // Вестник БрГТУ. – 2015. – №5(95): Строительство и архитектура. – С. 54-57.

VyOS introduction

Дєєв К.

КНУ ім. Т. Г. Шевченка, факультет радіофізики, електроніки та комп'ютерних систем maintainers@vyos.net

VyOS – це зручна мережева операційна система, побудована на GNU / Linux і об'єднує різні додатки з відкритим вихідним кодом, які широко використовуються незалежно один від одного. Вона дозволить вам використовувати ваш комп'ютер або сервер в якості маршрутизатора, мережевого екрану, VPN-концентратора чи будь яку іншу їх комбінацію. VyOS може працювати в середовищі KVM, Xen, VMWare чи Hyper-V, і поширюється під ліцензією GNU GPL v2 . Ви можете використовувати її для будь-яких цілей, ніколи не піклуючись про вартість ліцензій.

Основні переваги

VyOS це мережева операційна система з повною реалізацією CLI, яка надається в якості відкритого вихідного коду. Це дозволить вам перетворити ваш сервер або настільний комп'ютер в повноцінну платформу маршрутизації. VyOS в даний час активно оновлюється і підтримується розробниками. Сервери на базі Intel з сучасним процесором і відповідними мережевими інтерфейсами (Gigabit Ethernet і 10 Гбіт Ethernet в даний час) надають чудову і дуже високу пропускну здатність у порівнянні з традиційними SMB маршрутизаторами Cisco, але з більшою гнучкістю для

модернізації в разі потреби.

Навіщо вам потрібен цей продукт

Коли ви плануєте рухатися в напрямку IT-інфраструктури для центру обробки даних сторонніх виробників, ви повинні точно вирішити, як всі інформаційні ресурси організації будуть об'єднані в єдину мережу. Рішення від провідних виробників часто є дорогими, компанії малого та середнього розміру не можуть прийняти їх з багатьох причин. В цьому випадку збільшується інтерес до безкоштовних продуктів з відкритим вихідним кодом, що дає багато можливостей, які не поступаються платним аналогам, а іноді навіть випереджають їх. Важливим елементом такої мережі є маршрутизатор - спеціалізований мережевий пристрій, який є об'єднанням частин мережі і переадресації пакетів між ними. Маршрутизатор можуть працювати як на апаратній так і на програмній основі. У разі, коли вам потрібно побудувати IT-інфраструктуру з мінімальними витратами, використання програмного маршрутизатора може бути кращим рішенням і VyOS є одним з основних представників такої ніші.

Ви можете задати питання, чому VyOS? І коротка відповідь вже тут: тому що VyOS це все, що вам потрібно, щоб створити інтелектуальну маршрутизацію у вашій організації. VyOS є платформа маршрутизації, яка підтримує велику кількість протоколів і функціональних можливостей: брандмауер, NAT, IPsec ... Якщо ви плануєте створити ідеальну і надійну мережу ви на правильному шляху використання VyOS, тому що вона дозволяє швидко і легко керувати, захищати, підтримувати і контролювати вашу мережу. VyOS має підтримку протоколу IPsec, реалізовану у вигляді системи, яка дозволить не турбуватися про вашу безпеку. У вас також є багато можливостей в написанні власних програм для VyOS, щоб реалізувати свої власні ідеї, тому що VyOS це продукт з відкритим вихідним кодом.

Використання

VyOS побудований на ядрі Debian і є представником програмного забезпечення для мережевого маршрутизатора. Вона може розширити свої можливості за рахунок установки додаткових Dev-пакетів. В даний час VyOS є мультиплатформна операційна система, так як вона може працювати на: KVM, Xen, VMWare, Hyper-V, навіть на VirtualBox як в 32-bit так і в 64-bit. VyOS також може працювати з багатьма мережевими інтерфейсами: 802.11, бездротовими модемами, Ethernet з 802.1Q VLAN, STP... Оскільки VyOS є мережевою ОС, ми повинні згадати про NAT. NAT є однією з найсильніших особливостей VyOS. Всі громіздкі і обтяжуючі процедури спрощені шляхом уточненої та оптимізованої CLI. Щоб уникнути перевантаження мережі, програмне забезпечення VyOS має функції балансування навантаження у каналах зв'язку. Для захисту інформації VyOS використовує протокол IPsec і інфраструктуру відкритих ключів PKI, що дозволяє захистити вашу інформацію від крадіжки. Система VyOS також підтримує реалізацію OpenVPN, що дозволяє створювати захищені канали зв'язку до користувача,

що знаходиться за межами позаду NAT.

Швидко і як і раніше зручно

Розродка VyOS почалася з останнього випуску Vyatta з метою забезпечення системних адміністраторів і мережових інженерів набором інструментів, який включає в себе програмне забезпечення лише з відкритим вихідним кодом для перетворення будь-якого комп'ютера в життєздатний і надійний мережовий маршрутизатор або між мережовий екран. Новий VyOS є набагато кращим, ніж старий Vyatta, він має стабільніше ядро, інтерактивний та продуманий CLI і готовий до автоматизації. VyOS також набагато більш ефективний з точки зору споживання пам'яті. VyOS може обробляти приблизно в 3 рази більше станів мережових з'єднань у брандмауері в тому ж обсязі пам'яті.

Готовність до експлуатації підприємства

VyOS в своїй історії має десятки успішних впроваджень в VoIP телекомунікаціях і невеликих ISP. VyOS прекрасно справляються із завданням створення і підтримки неперервної роботи мережі. Зростаюче співтовариство кваліфікованих розробників VyOS ядра готові надати різні види підтримки, від відповідей на запитання до повного аутсорсингу і кадрового забезпечення вашої внутрішньої мережі, як особисто так і з допомогою інтеграторів - компаній управління IT-послугами, якщо вам потрібен такий договір підтримки з такою організацією, приєднуйтеся до зростаючого SDN з VyOS.

Автоматизація роботи адміністратора онлайн-олімпіад на основі платформи EJUDGE

Дуч М.С., Цимбал Ю.В.

Національний університет "Львівська політехніка", Інститут комп'ютерних наук та інформаційних технологій mclex.one@gmail.com

Розглядається використання системи проведення учнівських предметних інтернет-олімпіад та тестувань знань на основі платформи ejudge, можливості подальшої автоматизації роботи адміністратора такої системи

Вступ. Останнім часом стрімко зростає використання вільного програмного забезпечення у освіті. Це дає змогу ощадити кошти, а також полегшувати роботу вчителів та викладачів, завдяки доступності вихідного коду вивчати, як воно працює, вільно копіювати таке програмне забезпечення для домашнього використання [1].

Платформа ejudge і її використання. Яскравим прикладом використання вільного програмного забезпечення в освіті є платформа для створення систем проведення онлайн-тестувань ejudge, яка була розроблена у 2002 році і вдосконалюється з кожним роком. Її переваги – відкритий код, а також

публічна безкоштовна ліцензія [2].

За допомогою системи на цій основі є можливість проводити турніри різних олімпіад, контрольні заходи, а також пробні підготовчі тестування для учнів і студентів.

Можливість дистанційного проведення дає змогу вирішити такі проблеми очних олімпіад як:

- необхідність пошуку місця для проведення змагань;
- обмеження кількості учасників олімпіади;
- транспортні витрати учасників та організаторів.

Завдяки відкритості коду і публічній ліцензії платформи ejudge з'являється можливість модифікувати, доповнювати, або переробляти її під певний турнір чи тестування. Університет "Львівська політехніка" активно використовує цю платформу для підготовки та проведення олімпіад, як для заочних, так і для очних турів. Упродовж декількох років спільною працею викладачів та студентів університету і вчителів шкіл проводяться Львівські міські учнівські Інтернет-олімпіади з різних предметів (інформаційних технологій, математики, англійської мови тощо) [3]. Використання цієї системи не обмежується лише олімпіадами, деякі вчителі проводять з її допомогою підготовчі або передконтрольні опитування зі свого предмету.

Автоматизація роботи адміністратора. Ця система кожного року доповнюється сценаріями автоматизації роботи адміністратора турніру, а також іншими додатковими модулями, в розробці яких беруть участь студенти університету.

Прикладами "неавтоматизованих" задач у базовій реалізації ejudge є:

- перевірка анкетних даних;
- розсилання e-mail повідомлень для учасників;
- випадковий вибір питань за вказаними темами і розділами, формування декількох варіантів завдання;
- генерування логінів із імені/прізвища та паролів;
- формування PDF-файлу з результатами турніру.

Розглянемо детальніше деякі ідеї щодо автоматизації цих задач.

1. Перевірка анкетних даних.

Щоб забезпечити формалізацію вигляду даних у турнірних таблицях, необхідно редагувати введені анкетні дані учасників, наприклад, враховувати, що учасники по-різному можуть вводити назви шкіл та гімназій тощо. Вирішенням цієї проблеми може бути сценарій, який забезпечить формування переліку назв шкіл та гімназій, і учасник під час реєстрації матиме можливість вибрати повну назву своєї школи або гімназії.

2. Розсилання e-mail повідомлень для учасників.

Створюється програмний модуль, який виконує функцію одночасного розсилання повідомлень для учасників, список яких береться із файлу або бази даних. Цей модуль дає змогу надсилати повідомлення з однаковим текстом для усіх, а також з різними текстами і наповненнями для різних груп

користувачів.

3. Випадковий вибір питань за вказаними темами і розділами, формування декількох варіантів завдання.

Під час проведення контрольних і самостійних робіт необхідно сформулювати якнайбільше варіантів питань, що досить часто стає ручною і рутинною роботою адміністратора. За допомогою відповідного програмного модуля, можна буде вибирати задану кількість питань за певною темою і автоматично формувати вказану кількість варіантів.

4. Генерування логінів із імені/прізвища та паролів.

Під час реєстрації переможців 1-го (заочного) туру на 2-й (очний) тур для зручності ідентифікації учасників логіни формуються із імені та прізвища і транслітеруються латиницею. У разі ручної роботи це займає багато часу, тому доцільно розробити програмний модуль для автоматичного генерування, вхідні дані для якого будуть взяті із файлу або бази анкетних даних. Також важливим аспектом є генерування паролів для очного туру і формування карток з необхідною інформацією, які видаються учасникам перед стартом.

5. Формування PDF-файлу з результатами турніру.

Такий модуль також буде корисним під час підготовки 2-го (очного) туру олімпіади. З таблиці результатів 1-го туру береться логін учасника-переможця, програмно виконується пошук необхідної інформації про нього у базі даних ejudge, наприклад, імені та прізвища, школи, класу, кількості набраних балів. Далі формується PDF-файл з вибраними даними про переможців, який використовується для публікації в Інтернеті, або розміщення в друкованому вигляді на стенді.

Висновок. Система проведення онлайн-олімпіад і тестувань на основі платформи ejudge підтримує можливості використання сучасних Інтернет-технологій в освіті. Завдяки спільній роботі викладачів і студентів Національного університету “Львівська політехніка” з кожним роком ця система стає все більше автоматизованою і зручною у використанні.

Джерела

1. [Електронний ресурс] Перспективи та можливості впровадження вільного програмного забезпечення в навчальних закладах та державних установах України <http://old.niss.gov.ua/monitor/june2009/15.htm>
2. [Електронний ресурс] Офіційний сайт платформи ejudge <http://ejudge.ru/>
3. [Електронний ресурс] Офіційний сайт системи проведення Львівських міських учнівських Інтернет-олімпіад <http://ejudge.lp.edu.ua/>

Програмне забезпечення для аналізу протоколів комп'ютерної мережі для ОС Ubuntu

Федевич О.

Інститут комп'ютерних наук та інформаційних технологій Національного університету “Львівська політехніка”, olhafedevych@gmail.com

The article describes designed computer network protocol analyzer from the standpoint of traffic flow analysis for the operating system Ubuntu. Developed software for traffic monitoring was proposed to use to collect data about the network traffic load with the purpose of forecasting the traffic flow. The developed software also has the modified Dijkstra's algorithm for calculating of the minimum road in the network. Based on forecasting and optimal way, adaptive control of traffic flow for minimization of packets delay was implemented.

Удосконалення алгоритмів маршрутизації може суттєво покращити продуктивність роботи вузлів комп'ютерної мережі. Всі інші вимоги, такі як надійність, сумісність, керованість, захищеність, розширюваність і масштабованість [1] безпосередньо пов'язані з якістю виконання цього основного завдання.

У даній роботі розглянуто створений аналізатор мережевих протоколів для моніторингу мережі в реальному часі, на основі розробленої раніше [2] математичної моделі прогнозування трафіку потоку. На основі отриманих результатів досліджень запропоновано приймати ефективні рішення адаптивного управління для розподілу завантаження вузлового обладнання. Також розглянуто модифікований алгоритм маршрутизації, оскільки він має суттєвий вплив на ефективність функціонування комп'ютерної мережі та його зв'язок з такими основними параметрами трафіку в комп'ютерній мережі як її пропускна здатність; затримка передавання даних та варіація затримки передавання (джиттер).

Метою даної роботи було створити програмне забезпечення для аналізу протоколів комп'ютерної мережі для операційної системи Ubuntu, експериментально дослідити та проаналізувати трафік потоку комп'ютерної мережі на основі збережених зразків даних спостережень над комп'ютерною мережею. Дані для дослідження взяті з проекту The Opte Project [3]. Інструментом дослідження став розроблений аналізатор мережевих протоколів з прогнозуванням поведінки трафіку потоку, реалізований у середовищі Qt Creator мовою програмування C++. Моделювання проводилось для декількох випадків. Зокрема для ста, чотирьохсот та двох тисяч вузлів. На рисунку показано приклад візуалізації даних для всесвітнього інтернет трафіку у 2015 році з врахуванням двох тисяч вузлів мережі. Здійснений аналіз трафіку використовується для прогнозування трафіку потоку на наступні декілька хвилин, що використовується для

прийняття коректних рішень адаптивного управління вузловим обладнанням комп'ютерної мережі.

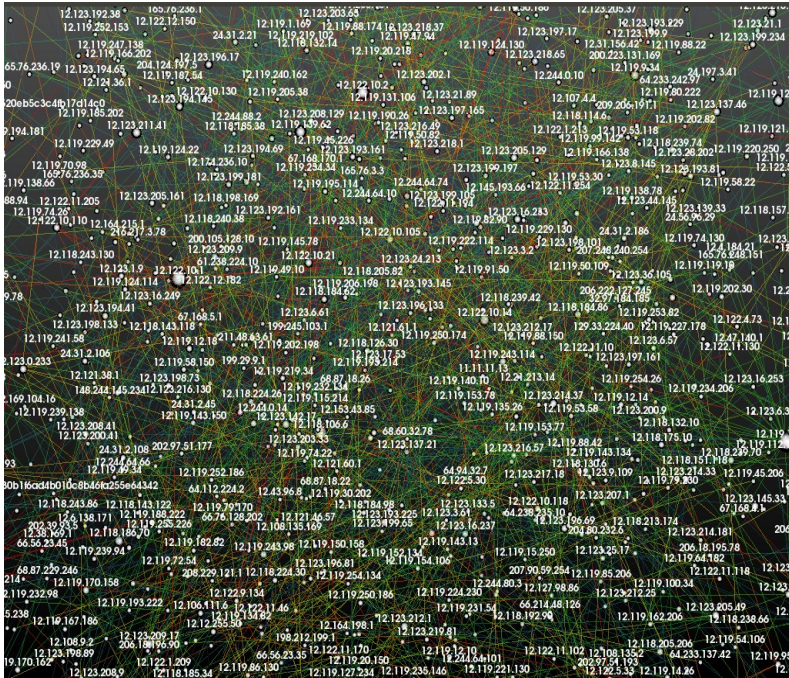


Рисунок. Візуалізація аналізатором Інтернет-графу для 2000 вузлів.

Авторська розробка для даних з [3].

Використання розробленого аналізатора мережевих протоколів у комп'ютерній мережі дає змогу значно економити час, автоматично та цілодобово збирати дані трафіку потоку з вузлів мережі, та в режимі реального часу слідкувати за роботою комп'ютерної мережі.

Література

1. Гончарук С.В. Администрирование ОС Linux – М.: Национальный Открытый Университет «ИНТУИТ», 2016
2. Ivanna Dronjuk, Maria Nazarkevych, Olga Fedevych. Asymptotic method of traffic simulation (Distributed Computer and Communication Networks) // Communications in Computer and Information Science. Springer 2014, Vol. 279, pp.136-144.
3. The Opte Project, The Internet 2015, [Електронний ресурс]. – Режим доступу: <http://www.opte.org/the-internet/>

Вплив похибки інтерполяції степеневих спектрів з нецілими показниками степеня на точність чисельного обчислення інтегрального перетворення Фур'є

Флюнт О. Є.

*Факультет електроніки Львівського національного університету імені Івана Франка,
flunt@electronics.lnu.edu.ua*

Effect of an interpolation error on numerical calculation of integral Fourier transform of fractional power spectra has been analyzed. For appreciation of accuracy interval it is proposed to calculate of Fourier integral for two interpolation dependencies drawn below and above the frequency spectra. Appreciation of confidence ranges of Fourier transform of fractional power frequency spectra $\omega^{-(1-n)}$ has been carried out as for small, so for large exponent n ($0 < n < 1$).

Задача чисельного обчислення інтегрального перетворення Фур'є важлива як у сучасній фізиці конденсованого стану, так і для аналізу степеневих залежностей у комп'ютерних системах. Зокрема у діелектричній спектроскопії твердих тіл і рідин часто потрібно переходити від імпульсних характеристик, отриманих як залежності від часу, до частотних спектрів, отриманих у частотному просторі, і навпаки. Інтерполяція залежностей, які відомі в окремих точках, за допомогою кубічних сплайнів є одним з найкращих методів отримання значень функцій між відомими точками, оскільки кубічні сплайни дозволяють згладжувати не лише значення функції у точках дотику між сусідніми відрізками, але і її першу та другу похідну. До того ж, їхнє використання дозволяє проводити обчислення перетворення Фур'є спектрів, відстані між точками яких неоднакові за величиною [1, 2]. Це важлива перевага методу, оскільки відстані між точками реальних спектрів переважно зростають з підвищенням частоти відповідно до експоненціального закону. Метод приводить до виразів, які є точними

результатами аналітичного обчислення інтегралів виду $a_i \int_{\omega_{m-1}}^{\omega_m} \omega^i \cos(\omega t) d\omega$ з

$i = 0, 1, 2, 3$. Проте для реалізації аналітичного методу обчислення інтегрального перетворення Фур'є за допомогою інтерполяції експериментальних частотних залежностей кубічними сплайнами необхідне використання чисел з високою розрядністю: більшою ніж 18-19 значущих цифр [1, 2]. Високу точність обчислень можна реалізувати за допомогою бібліотеки високоточних обчислень MPFR [1, 3], яка поширюється за ліцензією GNU Lesser GPL.

Оскільки частотні спектри часто підлягають степеневим залежностям від частоти з нецілими значеннями показника степеня, то сплайнів третього порядку повинно бути цілком достатньо для максимально адекватного

відтворення форми спектра. Проте, зрозуміло, що завжди буде присутня певна чисельна похибка точності відтворення значення спектра на певній частоті, яка в свою чергу може приводити до похибки форми інтегрального перетворення Фур'є. Тому важливою є задача оцінки точності перетворення Фур'є, яка зумовлена неточністю інтерполяції експериментальних залежностей заданих тільки у певних точках. Надалі для оцінки можливої похибки, зумовленої неточністю інтерполяції, будемо вважати, що значення частотних залежностей у заданих точках є точними; оцінювати будемо як можливі похибки інтерполяції спектра між заданими точками можуть вплинути на форму інтегрального перетворення Фур'є.

Спектри дійсної частини діелектричної проникності твердих тіл в діапазоні низьких та радіочастот є переважно монотонно спадними функціями. Тому похідна в будь якій відомій точці не перевищує значення, обчислене як середнє на наступному відрізку від f_i до f_{i+1} . Тому лінійна апроксимація на певному відрізку буде мати у кожній точці значення, яке не менше за фактичне значення спектра, а інтеграл від неї буде не меншим за значення інтеграла від справжньої спектральної залежності.

Для знаходження нижньої межі інтеграла використаємо ту властивість спектра, що похідна записана як $(C_i - C_{i-1})/(f_i - f_{i-1})$, де C_i і C_{i-1} – значення спектра на частотах f_i і f_{i-1} , відповідно, буде перевищувати фактичне значення похідної у точці f_i . Середня похідна на відрізку від f_{i+1} до f_{i+2} буде дещо меншою за фактичне її значення у точці f_{i+1} . Тому інтерполяція за допомогою кубічного сплайна з накладання умов прирівнювання похідних до їхніх середніх значень на сусідніх відрізках для монотонно спадної функції дасть інтерполяційну залежність, значення якої у кожній з точок не буде перевищувати справжні значення спектра.

Такий самий підхід можна застосувати для зростаючих ділянок спектра, які часто можна зустріти на залежностях діелектричних втрат від частоти. У цьому разі верхню і нижню межі інтеграла на кожному з відрізків треба поміняти місцями.

Надалі розглянемо два граничні випадки: коли підінтегральний вираз інтегрального перетворення Фур'є є неосцилюючим (коли $\omega t \ll 1$) і осцилюючим ($\omega t \gg 1$). Результати обчислення показують, що кусково-лінійна апроксимація приводить до незначного відхилення (порядку 1-2 %) значення інтегралів на проміжках для відрізків без осциляцій (рис. 1), великі відносні похибки можуть виникати за умови однієї осциляції на відрізок, які значно зменшуються за умови більшої кількості осциляцій.

З рис. 2 видно, що інтеграл Фур'є, обчислений у широкому частотному діапазоні (наприклад, 1 Гц – 1 МГц) буде мати значно менші межі можливої сумарної похибки зумовленої неточністю інтерполяції спектра, що можна пояснити сумарним вкладом як сильно-чутливих до точності інтерполяції фрагментів (з порядком однієї осциляції на відрізок), так і слабо-чутливих (без осциляцій або з великою кількістю осциляцій на відрізок).

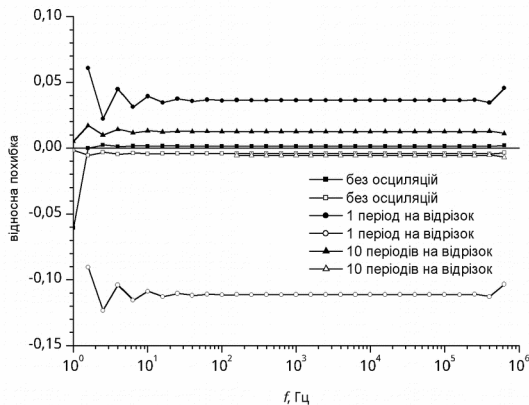


Рис. 1. Верхня та нижня допустимі межі значень для косинус інтеграла Фур'є від неосцилюючих та осцилюючих підінтегральних виразів на кожному з відрізків від f_{i-1} до f_i для степеневого спектра з $1-p = 0,2$

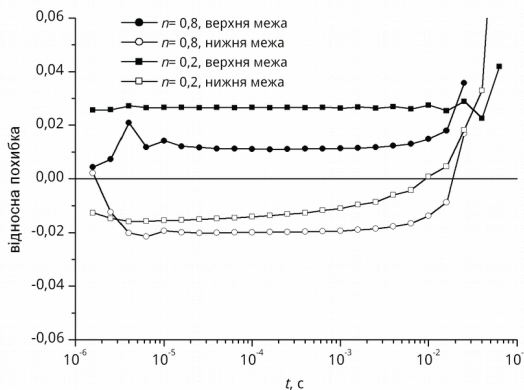


Рис. 2. Межі відносних похибок значень для косинус інтеграла Фур'є від степеневих спектрів з $1-p = 0,2$ та $0,8$ заданих на частотному проміжку від 1 Гц до 1 МГц

Отже запропонований метод можна використовувати для встановлення максимальних меж можливих відносних похибок чисельного перетворення Фур'є методом апроксимації кубічними сплайнами частотних степеневих спектрів, зумовлених неточністю інтерполяції. Також можна стверджувати, що можлива похибка, зумовлена неточностями інтерполяції частотних залежностей, значно менша, від можливих похибок, зумовлених відсутністю вкладів інтеграла Фур'є на нижчих та вищих частотах поза межами

частотного діапазону, на якому форма спектра відома [1, 2].

Джерела:

1. Флюнт О. Вплив розрядності чисел на правильність та точність чисельного розрахунку перехідних діелектричних характеристик / О. Флюнт // Вісник Львів. ун-ту. Серія фізична. – 2013. – Вип. 48. – С. 270–278.
2. Флюнт О. Розрахунок перехідної характеристики низькоомних шаруватих кристалів GaSe // Вісник Львів. ун-ту. Сер. фіз. 2009. – Вип. 44. – С. 226–233.
3. The Multiple Precision Floating-Point Reliable Library // The MPFR team. – 2013. – 56 p. – <http://www.mpfr.org/mpfr-current/mpfr.pdf>.

Створення інформаційно-аналітичної системи самооцінювання освітньої діяльності педагогічного університету **Франчук В.М., Франчук Н.П.**

*Національний педагогічний університет імені М.П. Драгоманова,
v.m.franchuk@npu.edu.ua, n.p.franchuk@npu.edu.ua*

The research is devoted to the creation of information and analytical system of self-assessment pedagogical university. For creation of the system used free distributed software for the developing web applications.

Вступ. Під самооцінюванням освітньої діяльності педагогічного університету розуміється всебічне (або сегментне) обстеження освітньої установи (або її структурного підрозділу), підсумком якого є формування інформаційної бази для подальшого аналізу та удосконалення процесів освітньої діяльності в установі. Самооцінювання використовується педагогічним університетом також для порівняння своєї діяльності з кращими в певному класі досягненнями інших педагогічних університетів чи показниками світового рівня в цій галузі (бенчмаркінг), а також може бути корисним у разі порівняння з поставленими раніше цілями при повторному оцінюванні ступеня досягнення цих цілей.

Постановка задачі. Проблема полягає в необхідності створення інформаційно-аналітичної системи самооцінювання діяльності педагогічного університету, яка б відображала реальний стан освітніх процесів у педагогічному університеті з використанням об'єктивних показників, критеріїв якості освіти.

Мета. Розроблення методологічного підходу застосування інформаційно-комунікаційних технологій для самооцінювання освітньої діяльності педагогічного університету.

Основна частина. В основу самооцінювання діяльності педагогічних університетів потрібно покласти модель, яка надасть можливість, використовуючи критерії і підкритерії, сформулювати перелік вимог до рівня організації навчально-виховного процесу у педагогічному університеті.

Для реалізації цієї моделі було розроблено критерії оцінювання для:

викладачів:

Показники досягнутого рівня кваліфікації та організаційної роботи.

Показники навчально-методичної роботи.

Показники науково-дослідної роботи.

Показники виховної роботи і навчально-дослідної роботи студентів.

Показники досягнутого рівня кваліфікації та організаційної роботи.

Показники навчальних досягнень.

Показники науково-дослідної роботи.

навчальних підрозділів педагогічного університету:

Загально кафедральні показники.

Загальноінститутські показники

Розроблення інформаційно-аналітичної системи самооцінювання освітньої діяльності педагогічного університету передбачало створення таких компонентів:

система оцінювання діяльності викладачів (Рис. 1);

система оцінювання діяльності студентів;

система оцінювання діяльності навчальних підрозділів педагогічного університету.

← → ↻ www.cmsrating.npu.edu.ua/profile/measure/1		🔍 📄 📁 📌 📧 ☰
Мій Профіль	Особова картка	Кафедри
Адміністративна панель		Вийти
2015-2016 ▼		
Факультет інформатики - Кафедра комп'ютерної інженерії		87
➤ Показники досягнутого рівня кваліфікації та організаційної роботи		87
➤ Показники навчально-методичної роботи		0
➤ Показники науково-дослідної роботи		0
➤ Показники виховної роботи і навчально-дослідної роботи студентів		0
Інститут розвитку дитини - Кафедра менеджменту та інноваційних технологій дошкільної освіти		0
➤ Показники досягнутого рівня кваліфікації та організаційної роботи		0
➤ Показники навчально-методичної роботи		0
➤ Показники науково-дослідної роботи		0
➤ Показники виховної роботи і навчально-дослідної роботи студентів		0

Рис.1

Під час розроблення інформаційно-аналітичної системи використовувалися такі засоби: інтерпретатор мови PHP; Symfony2 (PHP-фреймворк), використання якого надає зручні інструменти для розроблення веб-додатків; Doctrine2 – об'єктно-реляційне відображення (ORM) для PHP; TWIG PHP – шаблонізатор; MySQL – реляційна база даних; HTML5 – мова розмітки гіпертексту; CSS3 – мова опису зовнішнього вигляду документа; Bootstrap4 – CSS-фреймворк; JavaScript – клієнтська мова програмування; jQuery – бібліотека JavaScript.

Висновки: Важливо, щоб навчальні заклади мали в своєму

розпорядженні засоби збирання та аналізу даних про власну діяльність. За відсутності даних засобів навчальні заклади не будуть знати, що в їхній системі працює добре, а що потребує уваги, також невідомі результати нововведень. Інформаційні системи, що стосуються моніторингу якості, деякою мірою залежать від місцевих умов, але вони, принаймні, повинні охоплювати: навчальні та наукові досягнення студентів та показники їх успішності; попит на випускників на ринку праці; задоволеність студентів програмами підготовки за освітньо-кваліфікаційними рівнями «бакалавр», «магістр»; ефективність організації навчального процесу; склад студентів і його аналіз; професорсько-викладацький склад та його аналіз; доступні навчальні ресурси; головні показники освітньої діяльності даного навчального закладу.

Джерела:

1. Сергієнко В.П., Франчук В.М., Микитенко П.В. Використання інформаційно-комунікаційних технологій управління якістю освіти в педагогічному університеті. // Інформатика та інформаційні технології. - 2012. - №4. - С. 2 – 5.
2. Франчук В.М. Автоматизована система рейтингового оцінювання діяльності студентів і викладачів університету. IV International Summer School "Education Measurement: Teaching, Research, and Practice" is held under the support of the EU within the framework of the EU Project 145029-TEMPUS-2008-SE-JPCR "Educational Measurements Adapted to EU Standards".

Побудова захищеної системи реалізації транзакцій з використанням бібліотеки JcardSim

Гладкий Р.В, Желізняк Г.О.

Національний університет «Львівська політехніка», Львівський коледж Державного університету телекомунікацій, roman.gladkuy@gmail.com, zhelizniak_yosyf@ukr.net

In this article were analyzed ways to provide reliable and security solution, using smart-card based systems. As platform implementation has been used open source library JCardSim. Additionally, described open source used technologies, standards and specifications for communication with smart cards, methods to provide possibility of update and enhance mentioned solution.

Швидкий розвиток інтернету і безпроводних засобів цифрового зв'язку викликали великі зміни в способі використання електронної комерції. Із цим прогресом змінились також способи виконання дій, пов'язаних із покупкою товарів в інтернет магазинах, оплатою послуг. Виникла потреба у створенні систем безпеки для надійного використання електронної комерції. При розробці таких інформаційних систем є необхідність в реалізації безпеки та захисту даних, як на програмному, так і на фізичному рівні. Для захисту інформаційної системи на фізичному рівні ефективним інструментом є смарт-картки з вбудованим елементом безпеки, що надає доступ тільки

авторизованому користувачеві через протокол передачі даних. На відміну від захисту на програмному рівні, відсутня можливість будь-якої прямої взаємодії із програмою, зокрема пошук дефектів користувачем, через які можна оминати безпеку системи. Розробнику не обов'язково знати принципи роботи елемента безпеки чи протоколу передавання даних, для розробки програмного продукту достатньо мати загальні навички програмування та роботи із документацією.

Основною метою роботи є створення безпечної і захищеної системи, що дає змогу проводити транзакції із грошима за допомогою смарт-карток, а також терміналу для взаємодії із розробленим рішенням і підсистеми для автоматизованої валідації за допомогою тестових випадків. Можна виділити такі компоненти (див. рис.) розробленої системи 1)ужиток для смарт-картки із набором функціоналу для проведення транзакції; 2)термінал, для роботи із смарт-картками; 3)система для автоматичної валідації роботи ужитку за допомогою тестів; 4)емулятор смарт-картки для тестування і валідації роботи ужитку. Для реалізації завдання використано бібліотеку з відкритим вихідним кодом JCardSim (Java Card Runtime Enviroment Simulator) - методи і засоби для розробки Java Card застосунків і роботи із смарт-картками [1], а також документацію для смарт-карток, що описана в специфікаціях ISO-7816 [2], Open Card [3], Global Platform [4], та інших.

Аплет для смарт-картки є основною складовою частиною системи і містить набір функціоналів для захищених операцій: аутентифікації користувача, отримання приватної інформації, видачі валюти користувачеві та інші.

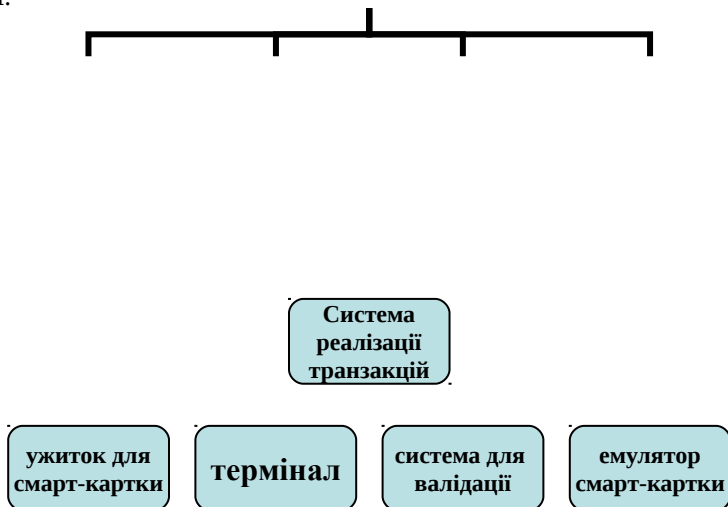


Рисунок. Структурна схема системи реалізації транзакцій

Термінал використовується для надання зручного інтерфейсу доступу до смарт-картки: надсилання запитів на проведення операцій та отримання і опрацювання відповідей. Взаємодія із картою відбувається через APDU команди (Application Protocol Data Unit – протокол що використовується для обміну даними із смарт-картками) [4]. Програмна складова що виконує валідацію, дозволяє виконувати завантаження тестів на розробленому аплеті, та аналізує результати їх виконання за критерієм відповідності параметрам безпеки. Це дозволяє проводити автоматизоване тестування в процесі розробки проміжних версій та відразу виправляти знайдені помилки. Емулятор смарт-картки виконує ті ж дії, що й чіп на картці. Використовується для прискорення завантаження тестів, а також для можливості розробки застосунків без використання апаратних пристроїв.

При користуванні системою можна виділити два основні потоки даних:

- запит користувача на транзакцію, тут користувач повинен надати необхідні дані в терміналі для авторизації чи виконання конкретної дії;
- результат виконання транзакції у вигляді повідомлення про її успішне завершення чи помилку.

Проведено експериментальні дослідження надійності даної системи, тестування та аналіз розробленого рішення. Приведено приклади тестових результатів та показана можливість подальшої оптимізації та оновлення системи.

Практична цінність розробленої системи полягає в можливості використання її як захищеної системи для операцій із валютою. А також передбачено можливості функціонального розширення, оновлення реалізації, покривання нового функціоналу додатковими тестами та автоматизованого тестування для забезпечення захищеності та надійності даної системи.

Джерела

1. Java Card Runtime Enviroment Simulator 2.2.2. - <https://jcardsim.org/>
2. ISO/IEC 7816, http://www.iso.org/iso/catalogue_detail.htm?csnumber=45989
3. OpenCard, <http://www.openscdp.org/>
4. GlobalPlatform, <http://www.globalplatform.org/>

***Вільнопоширюване програмне забезпечення в контексті
варіативності моделей комп'ютерно-орієнтованого середовища
навчання предметів природничо-математичного циклу
Гриб'юк О.О., Хошаба О.М.***

*Інститут інформаційних технологій і засобів навчання НАПН України, Вінницький
національний технічний університет, olenagrybyuk@gmail.com*

Taking into account the specific technical characteristics of dissonance on computer equipment used in schools, in the design of the individual components of computer-

based learning environment objects natural mathematical cycle used distributions of xUbuntu and Ubuntu. The proposed solution is the budget alternative to expensive as software, meet all the necessary requirements and promote the efficient organization of educational process at school. Thoroughly considered solutions to problems of designing learning environment and building models of varied disciplines of study natural and mathematical cycle using separate components of a computer based system for learning.

Із врахуванням специфіки наявного дисонансу щодо технічних характеристик комп'ютерної техніки, що використовується в загальноосвітніх навчальних закладах, в процесі проектування окремих компонентів комп'ютерно орієнтованого середовища навчання предметів природничо-математичного циклу рекомендується використовувати дистрибутиви систем xUbuntu і Ubuntu. Пропоновані рішення є бюджетною альтернативою дороговартісному програмному забезпеченню, відповідають усім необхідним вимогам та сприяють ефективній організації навчально-виховного процесу в школі. Дотепер актуальними є постановка і пошук шляхів вирішення проблем щодо проектування середовища навчання та побудови варіативних моделей навчання дисциплін природничо-математичних циклу з використанням окремих компонентів комп'ютерної орієнтованої системи навчання. Удосконалення системи освіти можливе за умови виявлення взаємозв'язків та ґрунтового тлумачення таких понять, як «знання», «мислення», «розвивальне навчання» та педагогічно виваженого поєднання традиційної системи навчання з окремими компонентами комп'ютерно орієнтованої системи навчання в школі [1]. У результаті ґрунтового аналізу добиралися візуальні середовища для мов програмування C++, C#; контроль комп'ютерів учнів викладачем; система управління та навчання в режимі он-лайн, в тому числі групового навчання, дослідження і тестування; створення розкладів навчальних занять та розподілу навантаження вчителів; комплект адміністративних програм для навчального закладу; програми для навчання алгебри, геометрії, фізики, хімії, географії, астрономії тощо. Здійснювався детальний аналіз та було виокремлено програми для тренування слуху, пам'яті, зору учнів тощо; робота з електронними картами (GIS); системи комп'ютерної математики, включно з системами динамічної математики (GeoGebra), Cantor, для побудови графіків математичних функцій використовуємо KAlgebra, OpenSource аналог MatLab, wxMaxima – графічний інтерфейс для системи комп'ютерної математики Maxima, система комп'ютерної алгебри Mathomatic, програма Cabri, геометричне середовище The Geometer's Sketchpad, GEONExT, Cinderella, TracenPoche та ін.; пакет прикладних математичних програм задля здійснення інженерних та наукових розрахунків; створення і розрахунок хімічних структур, перегляд макромолекул та підготовка до публікації їх зображень; редактори хімічних схем та реакцій; схематичне створення електронних схем, графічні стимулятори комп'ютерних мереж і електричних схем; цифрові осцилографи,

проектування електронних пристроїв і необхідних плат.

Побудова навчальної програми як індивідуального навчального проекту можлива завдяки ґрунтовно осмисленим траєкторіям індивідуального навчання учнів із врахуванням можливостей дедуктивного проектування навчального процесу шкільних дисциплін природничо-математичного циклу. Безперечно, для успішного виконання такої роботи з учнями реконструюється зміст навчання в контексті логічності його побудови та використання технології (педагогічного інструментарію та способів впровадження) в процес навчання дисциплін, в тому числі природничо-математичного циклу. Основні акценти розставляються на проектній діяльності [4], а відповідними цілями предметних дослідницьких проектів є створення умов для самореалізації та становлення авторської позиції учнів шляхом активної їх участі в предметному позакласному проекті; уточнення та диференціація понятійного апарату, систематизація знань учнів, встановлення міжпредметних зв'язків та підготовка олімпіадних завдань, підготовка до вступу у вищі навчальні заклади; корекція рівня сформованості різних учбових дій з використанням проектної діяльності [2].

Варіативна модель проектування представлена на основі компетентнісного підходу в сучасній освіті із врахуванням основних етапів проектування (цільового, методологічного, факторного, структурного, функціонального, ресурсного, дефіцитарного, процесуального, прогностичного та результативного) [3]. В процесі конструювання комп'ютерно-орієнтованого навчального середовища суб'єкти виконують наступні функції: формування відповідних компетентностей, оцінювання факторів впливу, визначення стратегії діяльності навчального закладу, оцінка ризиків та освітніх ресурсів, добір освітнього маршруту, добір варіативного змісту освітнього процесу; розроблення технологій та методик засвоєння необхідних компетентностей, експертиза навчальних програм та оцінювання компетентнісного результату [1].

У навчально-виховному процесі рекомендується використовувати утиліти FlashQard, GNU it Flashcard Trainer, симулятори tkgate Circuit Simulator, середовище програмування Kturtle, математичні функції KmPlot, інструмент для роботи з математичними поверхнями K3DSurf, альбоми геометрії Евкліда KSeg Geometry Sketchpad, інтерактивне математичне середовище програмування Euler. Під час навчання географії використовуються KGeography, географічну інформаційну систему Quantum GIS, редактор карт Emerillon Map Viewer.

Учням подобається здійснювати дослідження з використанням графічного симулятора комп'ютерних мереж gns3, Kumir Language Implementation, Quite universal circuit simulator, схематично створювати елементарні електронні схеми Oregano electrical engineering tool. Відповідно, для аналізу візуалізації даних використовується QtiPilot, візуалізація даних на основі MathGL, рекурсивний фрактальний редактор Qosmic, пакет наукової

графіки Veusz та інструмент для візуалізації зв'язків View Your Mind. Розвиток математичного мислення та відповідних навичок здійснюється за допомогою програм GCompris, MathWar Tux, of Math Command!; пам'ять розвивається за допомогою програмних засобів Mnemosyne, PySyCache, Gamine, LMemory, Linux Letters, Childsplay, Blinken, відповідно, слух тренується завдяки GNU Solfege та ін. У навчальному процесі для розвитку образного мислення і перспективи використовуються програми Tux!, KLetres.

Дослідження Всесвіту учням подобається здійснювати із використанням засобів космічного симулятора Celestia. Для симуляцій фізичних експериментів використовується також Step. Використання на уроках та в процесі проектно-дослідницької діяльності настільного планетарію Kstars, глобусу Marble та карт Pauker, планетарію Stellarium, програм для роботи з електронними картами Merkaator сприяє активізації пізнавальної діяльності учнів та підвищенню ефективності навчального процесу. В процесі навчання хімії використовуються періодична таблиця gElemental, GPeriodic, система Kalzium, системи молекулярного проектування та системи моделювання Avogado, а для зображення та обчислення хімічних формул – програмний засіб Chemtool, редактори хімічних схем та реакцій BKchem і xdrawchem Chemistry Editor. Для виконання досліджень у процесі навчання біології рекомендується пакет Unipro UGENE, для перегляду макро-молекул та підготовки до публікації їх зображень використовується RasMol та цифровий оцилограф Xoscope.

Проектування в навчальному процесі набуватиме розвивального характеру лише за умови унеможливлення використання репродуктивних форм діяльності, що спонукатиме школярів до творчого пошуку відповідей на проблемні питання у наставників, або літературних джерелах. В процесі проектування процесу навчання учнями дисциплін доцільно враховувати контрольну рефлексію з метою коригування термінів виконання роботи на кожному з етапів та покращення результату проектної діяльності. З використанням комп'ютерно-орієнтованого середовища навчання забезпечується можливість концентрації навчальних ресурсів; багатогранність траєкторій [2] та результатів формування необхідних компетентностей; доступність та рівність можливостей учнів в навчанні; поліфункціональність взаємодії суб'єктів навчального процесу (вчителів, учнів, батьків, адміністрації навчального закладу); орієнтацію змісту, форм та технологій підготовки учнів на інтеграцію освітню, наукову, дослідницьку, виробничу в умовах навчально-виховного процесу.

На підставі проведених досліджень можна стверджувати, що організовані варіативні моделі навчання за допомогою ґрунтовно педагогічно продуманих компонентів комп'ютерно-орієнтованого середовища навчання є перспективним напрямком щодо модернізації процесів навчання дисциплін природничо-математичного циклу у загальноосвітніх навчальних закладах.

Продовжуються дослідження, спрямовані на створення оптимальних умов [4] для перманентного підвищення рівня фахової майстерності вчителів, в тому числі математики, фізики, хімії, біології та ін., в контексті використання інформаційно-комунікаційних технологій у навчальному процесі загальноосвітніх навчальних закладів.

Джерела:

1. Гриб'юк О.О. Педагогічне проектування комп'ютерно орієнтованого середовища навчання дисциплін природничо-математичного циклу. / Гриб'юк О.О.// Наукові записки. – Випуск 7. – Серія: Проблеми методики фізико-математичної і технологічної освіти. Частина 3. – Кіровоград.: РВВ КДПУ ім. В. Винниченка, 2015. – С. 38–50.
2. Гриб'юк О.О. Когнітивна теорія комп'ютерно орієнтованої системи навчання природничо-математичних дисциплін та взаємозв'язки вербальної і візуальної компонент / Гриб'юк О.О. // Гуманітарний вісник ДВНЗ «Переяслав-Хмельницький державний педагогічний університет імені Григорія Сковороди» – Додаток 1 до Вип.36, Том IV (64): Тематичний випуск «Вища освіта України у контексті інтеграції до європейського освітнього простору». – Київ: Гнозис, 2015. – С. 158-175.
3. Гриб'юк О.О. Розв'язування евристичних задач в контексті STEM-освіти з використанням системи динамічної математики GeoGebra / О.О.Гриб'юк, В.Л.Юнчик // Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми. // Збірник наукових праць – Випуск 27 / – Київ-Вінниця: Планер, 2015. – С. 138-155.
4. Grybyuk O. Mathematical modelling as a means and method of problem solving in teaching subjects of branches of mathematics, biology and chemistry // Proceedings of the First International conference on Eurasian scientific development. «East West» Association for Advanced Studies and Higher Education GmbH. Vienna. 2014. P. 46-53.

Проектно-дослідницька діяльність в процесі навчання математики з використанням вільнопоширюваних програмних засобів

Гриб'юк О.О., Юнчик В. Л.

*Інститут інформаційних технологій та засобів навчання НАПН України,
olenagrybyuk@gmail.com, uynchik@gmail.com*

Research demonstrates effectiveness using the system GeoGebra in process solving mathematical problems for the purpose of enhance teaching and learning of students, basic functions Dynamic Mathematics system GeoGebra. Special give attention to the possibility of forming research competence of students In process solving mathematical problems. Provide examples using computer models, created using the system GeoGebra In process teaching students school of Mathematics.

У процесі навчання природничо-математичних дисциплін доцільно використовувати окремі компоненти комп'ютерно-орієнтованої системи навчання для розвитку проектно-дослідницької діяльності учнів. Система

динамічної математики GeoGebra є універсальним програмним засобом, що використовується для підтримки навчання геометрії, алгебри, математичного аналізу, теорії ймовірності, математичної статистики та інших розділів математики [4].

Система динамічної математики GeoGebra постійно оновлюється та вдосконалюється. Нещодавно з'явився новий інструмент, режим іспиту GeoGebra, що сприяє проведенню іспитів, не маючи доступу до Інтернету, GeoGebraTube або іншого програмного забезпечення, встановленого на комп'ютері. В процесі роботи учні з даним модулем всі дії документуються в журналі іспиту. Graphing Calculator Released GeoGebra використовується для телефонів і планшетів Android та для iPhone і Windows та сприяє роботі в процесі навчання природничо-математичних дисциплін та проектування динамічних графічних об'єктів, має доступ до GeoGebraTube. Напрацювання співтовариства GeoGebra становлять понад 300000 вільних і динамічних робочих листів і книг. Для зручної співпраці та колаборації між учнями та вчителями створено GeoGebra групи (Collaboration for Everyone), де є можливість опрацьовувати поштові тексти, зображення, відео, PDFs і робочі листи. В системі GeoGebra розроблено модуль, де можна задавати домашні завдання для учнів та прослідковувати їх роботу, оскільки зберігається оцінка, дата, тривалість і побудова кожної із спроб виконання. Учні можуть зберігати поточний стан виконаного завдання для доопрацювання аплета [3].

Значущою особливістю системи динамічної математики GeoGebra є інтеграція її у систему дистанційного навчання Moodle. Система GeoGebra інтегрується через фільтр GeoGebra Filter (Math Applets), отриманий із сайту moodle.org. З використанням GeoGebra Filter можна вбудовувати файли системи GeoGebra у лекційний матеріал, лабораторні та практичні заняття, тести, повідомлення форуму, блоги та інші складові системи. Врахування типу використовуваного апаратного засобу можливе у налаштуваннях фільтру з зазначенням ширини, висоти та параметрів аплетів GeoGebra. Використання файлів аплетів можливе як з сайту GeoGebra, так з сайту системи дистанційного навчання Moodle.

Ще один спосіб доступу до системи GeoGebra можливий з мобільного пристрою, з використанням експериментальної версії geogebmobile (<http://www.geogebra.org/mobile/>). використання даної версії мобільний пристрій потребує повної реалізації стандарту мови javascript та підтримки HTML5 у Web-браузері та

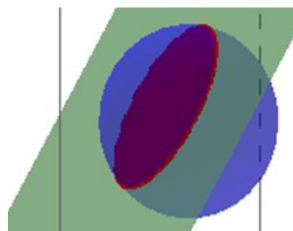


Рис.1. Перетин сфери площиною

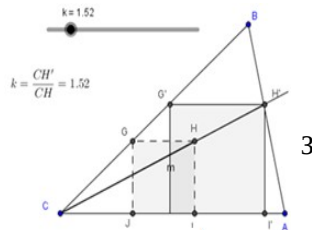


Рис. 2. Гомотетія

не потребує Java ME.

Функціонал системи GeoGebra включає ряд цікавих і затребуваних функцій, що використовуються в процесі навчання математики. З використанням функції `Intersect[<Object>, <Object>]` можна створити перетин об'єктів або функцій, вказуючи інтервал перетину та порядковий номер точки перетину. В залежності від об'єктів, що перетинаються, результатом перетину може бути відрізок, точка, площина, полігон, окіл та ін.

Приклад. Перетин сфери $a: x^2 + y^2 + z^2 = 3$ площиною $b: x + 6y + 4z = 4$ виконується з використанням функції `IntersectPath[b,a]` (Рис.1).

Команди трансформації використовують під час навчання геометричних перетворень, задач на доведення та на побудову. Однією з таких функцій є гомотетія, що задається функціями `Dilate[<Object>, <Dilation Factor>]` розширення об'єкта з початкової точки на коефіцієнт гомотетії. `Dilate[<Object>, <Dilation Factor>, <Dilation Center Point>]` розширення об'єкту з початкової точки, центру гомотетії, використовуючи коефіцієнт гомотетії.

Приклад. Дано трикутник СВА. Вписати в нього квадрат так, щоб дві вершини лежали на бічних сторонах трикутника, а інші дві вершини – на основі.

Розв'язання. Побудовано допоміжний квадрат GHIJ та промінь СН. В процесі виконання гомотетії квадрата відносно точки С, коефіцієнт гомотетії $k = \frac{CH'}{CH} = 1,52$ отримано потрібний квадрат (Рис. 2.).

Наступні функції `Reflect[<Object>, <Point>]` симетричне відображення геометричного об'єкта через задану точку (рис. 3. а). `Reflect[<Object>, <Line>]` симетричне відображення геометричного об'єкта через задану пряму (рис. 3. б). `Reflect[<Object>, <Circle>]` інвертування геометричного об'єкту відносно кола. `Reflect[<Object>, <Plane>]` симетричне відображення геометричного об'єкта відносно площини (рис. 3. в).

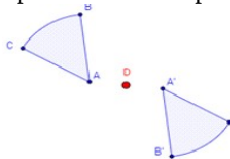


Рис. 3. а).



Рис. 3. б).

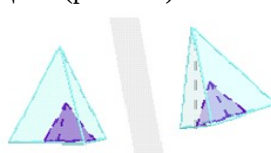


Рис. 3. в).

Методична система навчання з використанням комп'ютерно-орієнтованої системи будується на концепціях теорії проблемного навчання та теорії поетапного формування розумових дій, що забезпечує можливість управління навчальною діяльністю і створення орієнтувальної основи дій для розвитку творчих здібностей [5]. Водночас



Рис. 4. Побудова зображень з використанням команд трансформації

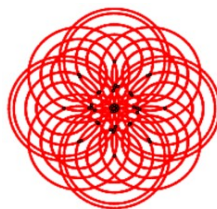
набуття навичок побудови і дослідження моделей сприяє розв'язуванню задач (<http://www.geogebra.org/m/w3KpBQPR>), що має самостійну значущість – воно створює передумови для розвитку системного і логічного мислення. Таке навчання забезпечує формування наукового світогляду [1].

Команди трансформації використовують під час побудови різних зображень (рис. 4), що забезпечує пізнавальний інтерес до навчання (<https://tube.geogebra.org/material/simple/id/2993293>).

Здійснення поворотів навколо точки, прямої та осі необхідне в процесі побудов фігур обертання, розв'язування задач на побудову та доведення, а також для створення різних зображень. (рис.5). Нижче наведено такі функції.

Rotate[<Object>, <Angle>] здійснення повороту геометричного об'єкта навколо осі на певний кут. Rotate[<Object>, <Angle>, <Point>] здійснення повороту геометричного об'єкта навколо точки на певний кут. Rotate[<Object>, <Angle>, <Axis of Rotation>] здійснення повороту геометричного об'єкта навколо осі обертання на певний кут.

Rotate[<Object>, <Angle>, <Point on Axis>, <Axis Direction or Plane>] здійснення повороту навколо початкової точки вектора.



Розроблені варіативні моделі комп'ютерно орієнтованого середовища навчання дисциплін природничо-математичного циклу урізноманіт-

нюються створеними з використанням систем комп'ютерної математики, в тому числі GeoGebra [2].

Отже, педагогічно доцільне та виважене впровадження систем комп'ютерної математики в процес навчання сприяє підвищенню ефективності навчання, а також активізує пізнавальну діяльність майбутніх фахівців. Інтеграція системи динамічної математики GeoGebra з системою Moodle надасть можливість організовувати та підтримувати процес навчання математики за допомогою вільно поширюваних програмних засобів, що створює можливості для реалізації в системі освіти різних моделей навчання, зокрема моделі змішаного навчання.

Література

1. Гриб'юк О. Розв'язування евристичних задач в контексті STEM-освіти з використанням системи динамічної математики GeoGebra / О. Гриб'юк, В. Юнчик // Сучасні інформаційні технології та інноваційні методи навчання у підготовці фахівців: методологія, теорія, досвід, проблеми // Зб. наук. пр. – Випуск 43 / Редкол. – Київ-Вінниця: ТОВ фірма «Планер», 2015. – С. 206 - 218.
2. Гриб'юк О.О. Педагогічне проектування комп'ютерно орієнтованого середовища навчання дисциплін природничо-математичного циклу. / Гриб'юк О.О.// Наукові записки. – Випуск 7. – Серія: Проблеми методики фізико-математичної і технологічної освіти. Частина 3. – Кіровоград.: РВВ КДПУ ім. В. Винниченка, 2015. – С. 38–50.
3. Гриб'юк О. Формування дослідницьких компетентностей учнів в процесі навчання математики з використанням системи динамічної математики GeoGebra / О. Гриб'юк,

В. Юнчик // Інноваційні технології навчання обдарованої молоді: матеріали VI-ї Міжнародної науково-практичної конференції, 3-4 грудня 2015 року, м. Київ. – Київ: Інститут обдарованої дитини, 2015 – С. 420–428.

4. Grybyuk O. Mathematical modelling as a means and method of problem solving in teaching subjects of branches of mathematics, biology and chemistry // Proceedings of the First International conference on Eurasian scientific development. «East West» Association for Advanced Studies and Higher Education GmbH. Vienna. 2014. P. 46-53.

5. Юнчик В. Модель змішаного навчання математики з використанням системи GeoGebra / В. Юнчик // Гуманітарний відділ ДВНЗ «Переяслав-Хмельницький державний педагогічний університет імені Григорія Сковороди» - Додаток 1 до Вип. 36, Том IV (64): Тематичний випуск «Вища освіта України у контексті інтеграції до європейського освітнього простору». – К.: Гнозис, 2015. – С. 559-568.

Особливості моделювання ринку програмного забезпечення

Грицюк Ю.І., Грицюк П.Ю.

НУ "Львівська політехніка", yurii.i.hrytsiuk@lpnu.ua

The features of the construction of the software models market regarding of using its legal and pirated versions has been considered. It was found that in the computer piracy conditions and possible reorientation between users of different origins of the application software, share of total market sales of software should be evaluated not by the number of licenses sold, but by the available number of its users.

На сьогодні індустрія ПЗ є однією з найбільш високотехнологічних і прибуткових сфер економіки України. За даними дослідницької компанії IDC [6], в Україні станом на кінець 2015 р. на ринку ПЗ функцінувало біля 2,3 тис. компаній, в яких було задіяно понад 180 тис. осіб з річним валовим доходом до 2 млрд. дол. США. Щорічно на ринку праці з'являється до 30 тис. випускників ВНЗ України – фахівців у галузі ІТ. Індустрія ПЗ добре розвинена в Києві, де працевлаштовані більше 40% усіх фахівців цієї сфери, вона швидко розвивається у Львові, Харкові, Дніпропетровську, Одесі, Миколаєві та інших містах України.

Однак, останнім часом в індустрії ПЗ спостерігається деяке зменшення інтенсивності його розроблення та прагнення постачальників використати більш ефективні канали розширення своєї присутності на ринку ПЗ. Це викликає підвищений інтерес закордонних ІТ-компаній до оптимізації системи маркетингу, зокрема, впровадження прикладного ПЗ у наукових, освітніх і комерційних установах України.

Основною причиною гальмування процесу розповсюдження прикладного ПЗ в Україні є піратство. Згідно з дослідженням Асоціації виробників ПЗ (Business Software Alliance), що охопило 32 країни, 69% користувачів комп'ютерів в Україні використовують прикладне ПЗ, отримане з нелегальних джерел [9]. Водночас, у більшості країнах, навіть далеких від галузі ІТ, частка нелегальних користувачів ПЗ становить в середньому 47%.

Це означає, що в Україні потенційні користувачі ПЗ купують на ринках піратські диски або завантажують програми з мережі Інтернет. Також в Україні приблизно 26% усіх торговельних фірм активно пропонують саме піратське ПЗ.

Не секрет, що розповсюдження неліцензійного ПЗ призводить до економічного відставання України від провідних ІТ-країн, значно знижується кількість робочих місць, скорочується рівень інвестицій і зменшується бюджет країни. За даними компанії IDC Українська економіка через піратський ринок ПЗ щороку втрачає більше 100 млн дол. США.

Ще в 1995 р. у Великобританії на реальних даних [2] було доведено, що пірати домінують в перетворенні потенційних шанувальників ліцензійного ПЗ на дієвих користувачів його копій. Група науковців за допомогою узагальнення фундаментальної моделі проникнення інновацій досліджували вплив тіншового (піратського) розповсюдження текстових редакторів і електронних таблиць на легальну продажу їх ліцензій. Виявилось, що інформація, яка передавалася розповсюджувачами піратських копій потенційним користувачам, мала практично таку ж саму дію, як інформація, яка передавалася легальними продавцями відповідних продуктів. Понад це, для цих двох типів ПЗ пірати забезпечили 80 % продажі!

Згодом, в 1997 р., ці ж самі науковці розглянули не тільки особливості законного і незаконного використання ПЗ, але й основні причини переорієнтації користувачів між різними торговими мережами – легальними і тіншовими. Отримані науковцями результати дали підставу зробити висновок про те, що в умовах існування комп'ютерного піратства і можливої переорієнтації користувачів між різними брендами (оригіналами і заміниками прикладного ПЗ), частку загального ринку ПЗ варто оцінювати не за кількістю проданих ліцензій, а за наявною кількістю його користувачів.

У роботі [4] автори сформулювали таку умову вигідності тіншової діяльності для піратів:

$$n \nabla Vл >> Vз + n \nabla Vп + (1 - P(n)) \nabla Vш(n),$$

де: n – кількість поширених піратських копій ПЗ; $Vл$ – вартість легальної версії ПЗ; $Vз$ – витрати на зламування системи захисту легального ПЗ; $Vп$ – вартість піратської копії ПЗ; $P(n)$ – ймовірність виявлення порушника, що розповсюдив n піратських копій ПЗ; $Vш(n)$ – величина штрафу за розповсюдження n піратських копій ПЗ. Зазвичай, витрати на порушення авторських прав за розповсюдження неліцензійного ПЗ є значно більшими від вартості придбання його піратської копії навіть з урахуванням можливого покарання.

Табл. 1. Причини придбання легального і піратського ПЗ

Причини придбання	
нових версій легального ПЗ:	піратських копій ПЗ:
– потреба використання ПЗ для навчання або роботи;	– дорожнеча легального ПЗ; – бажання поспробувати можливості

<ul style="list-style-type: none"> – вдале та тривале використання попередньої версії легального ПЗ; – наявність легальної документації; – дотримання чинного законодавства; – технічна підтримка легального ПЗ дистриб'ютором чи його виробником; – політика навчального закладу, державної чи приватної організації; – неможливість знайти ПЗ у знайомих; – гарантія захисту ПЗ від вірусів; – потенційна можливість оновлення поточної версії легального ПЗ; – престижність володіння новою версією легального ПЗ. 	<ul style="list-style-type: none"> – нової версії легального ПЗ через його копію; – недостатні доходи для придбання нової версії легального ПЗ; – короткотермінове використання ПЗ; – легкість зламування системи захисту і копіювання легального ПЗ; – тривале очікування нової версії легального ПЗ; – низька ймовірність викривання використання піратського ПЗ; – використання піратського ПЗ більшістю знайомих; – дещо жорсткі вимоги щодо отримання ліцензії на нову версію ПЗ.
--	--

У працях С.А. Середи [1, 7, 8] проведено детальне дослідження проблеми успішного функціонування ринку ПЗ, внаслідок чого було отримано основні причини використання легальних версій і піратських копій ПЗ (табл. 1). Також була розглянута графічна модель попиту і пропозиції на ринку ПЗ з урахуванням продажі його легальних версій і піратських копій (рис. 1), побудовані відповідні моделі поведінки і взаємодії агентів ринку ПЗ.

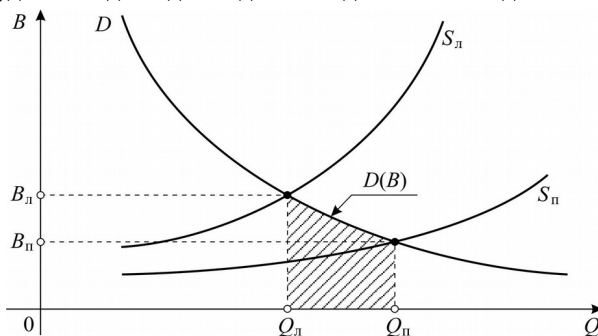


Рис. 1. Графічна модель ринку ПЗ з врахуванням його легальних версій та піратських копій

У графічній моделі ринку легального і тінювого ПЗ (рис. 1) крива тінювої його пропозиції S_p дещо пологіша, ніж крива пропозиції його офіційних розповсюджувачів S_l через істотно нижчу вартість піратських копій, а також дещо більшу еластичність тінювої пропозиції за вартістю. Тут спостерігаються дві точки ринкової рівноваги: точка $(Q_l; B_l)$ – рівновага легального ПЗ і точка $(Q_p; B_p)$ – його рівновага на тінювому ринку. Точка Q_p визначає загальний обсяг продажі піратських копій ПЗ, точка Q_l – обсяг

продажі легального ПЗ, водночас як обсяг тіншового ринку становить $Q_p - Q_l$. Втрати держави від тіншового розповсюдження ПЗ при цьому становлять

$$V_T = \int_{B_n}^{B_p} D(B)dB - Q_n(B_p - B_n) + B_n(Q_p - Q_n) = \int_{Q_n}^{Q_p} D(Q)dQ,$$

де: $D(B)$ – попит на ПЗ при вартості B ; $D(Q)$ – попит на ПЗ при продажі Q .

Побудовані в працях [7, 8] моделі поведінки виробника ПЗ, користувача і пірата є моделями задач, які розв'язуються методами дискретного лінійного програмування з булевими змінними. Також у цих працях розглядаються дві матричні гри, що описують конфліктні ситуації між державою і тіншовим ринком ПЗ, а також між виробником ПЗ і піратами.

У праці [5] А.М. Козирев сформулював основні принципи, на яких ґрунтується економіка піратства при аналізі ринку ПЗ:

витрати на створення ПЗ мають фіксований характер;

витрати на тиражування та розповсюдження ПЗ близькі до нуля;

корисність, яку можуть отримувати потенційні користувачі при використанні ПЗ, має грошовий еквівалент;

одному користувачеві потрібна тільки одна легальна версія чи піратська копія прикладного ПЗ;

кількість користувачів, готових купити одне ПЗ за деякою фіксованою вартістю, обернено пропорційно до цієї вартості;

повна вартість ПЗ – максимальна сума, яку готові заплатити за отримання його одного примірника всі потенційні користувачі.

Виходячи з цих припущень А.М. Козирев вивів такі два твердження:

якщо вартість примірника ПЗ постійна для всіх його покупців, то прибуток виробника ПЗ не залежить від його вартості;

повну вартість ПЗ може отримати його виробник, якщо кожному користувачеві він продасть примірник ПЗ за найвищою вартістю, прийнятною для цього покупця.

Отже, користувачі, які купують ліцензійне ПЗ, закладають надійне підґрунтя свого бізнесу. Це інвестиція у надійне та безпечне сьогодення, а також у майбутній успіх. Піратські ж копії ПЗ, завантажені з файлообмінника мережі Інтернет, можуть не тільки погано функціонувати, але й містити троянські закладки, які нададуть зловмисникам доступ до важливих ваших документів, персональних і фінансових даних, а також до керування комп'ютером.

Багато з програмних аналітиків вважають, що війни між Microsoft і вільним ПЗ хоча і завершилися, однак запроваджувати вільне ПЗ в українській освіті й науці ніхто не поспішає. Наприклад, Ігор Полянський у своїй праці [3] дає розлогі коментарі щодо активного використання вільного ПЗ в освіті, у дослідницькій роботі, в адмініструванні. Воно допоможе позбутися залежності від ПЗ корпорації Microsoft, більшість якого має

піратське походження. А боротьба з піратством в Україні схожа на боротьбу з курінням. Адже методи боротьби так само неефективні, як і попередження про шкоду куріння. Хоча держава вживає ті чи інші заходи, говорить про важливість боротьби з піратством ПЗ, закриває файлообмінники, але результату немає. Окрім цього, в Україні за порушення авторських прав передбачена не тільки цивільна й адміністративна, а навіть кримінальна відповідальність.

Чому ж чинне законодавство і вжиті заходи не працюють?

Насамперед це пов'язано з ментальністю пересічного українця. Адже багато не розуміють, навіщо платити за музику, фільми чи ПЗ, якщо їх можна просто завантажити з мережі Інтернет. Більше цього, далеко не всі користувачі прикладного ПЗ розуміють, що автор або колектив авторів працював над своїм творінням і, відповідно, заслуговує хоча б на мінімальну винагороду.

Якщо йдеться про ПЗ корпорації Microsoft, то його вартість непомірно висока відносно доходів середньостатистичного українця. Адже не кожен українець може дозволити собі купити операційну систему Windows 8 за 1000 грн та MS Office 2016 для навчання за 2609,0 грн чи професійний за 10535 грн. Закордонні ж корпорації, які пропонують прикладне ПЗ, не розуміють того, що не кожному українцеві його вартість по кишені.

У багатьох країнах окрім пропрієтарного ПЗ популярності набирає так зване вільне ПЗ, яке для звичайного користувача здебільшого є безкоштовним. У країнах Західної Європи, Японії та Китаї в державному та освітньому секторах уже давно використовується вільне ПЗ. Навіть в РФ ще в 2007 році був прийнятий 3-х річний план переходу на вільне ПЗ в освітній сфері.

В Україні також були аналогічні спроби, які так і не увінчалися успіхом. Наприклад, ще в 2002 році народним депутатом Борисом Олійником було внесено проект Закону про використання відкритого (вільного) ПЗ в державних установах і державному секторі ведення господарства. Минуло понад 10 років, але наша Верховна Рада так і не спромоглася його прийняти, адже є і більш важливі завдання, наприклад, комунальна реформа чи приватизація державного майна. Тому вживати ті чи інші заходи без підтримки відповідного законодавства – недостатньо. Необхідно кардинально змінити підходи, тобто донести до свідомості українців про важливість дотримання прав на інтелектуальну власність і на державному рівні впроваджувати програми з популяризації вільного ПЗ, що значно зменшить піратство на ринку ПЗ.

Тим не менше, за статистичними даними компанії IDC [6], в Україні станом на кінець 2015 р. майже 72% користувачів були готові придбати ліцензійне ПЗ для своїх комп'ютерів, а 47% з них вже придбали нові його версії. Серед основних причин такого вибору українці відзначили надійність, стабільність і високу якість роботи ліцензійного ПЗ, бажання отримувати

його нові оновлення від виробника та технічну підтримку від дистриб'ютора, а також небажання порушувати чинне законодавство та інформаційну безпеку.

Література:

1. Благодатских В. А., Середа С. А., Посакалов К. Ф. Экономико-правовые основы рынка программного обеспечения. – М. : Финансы и статистика, 2007. – 240 с.
2. Гивон М., Махаджан В., Мюллер Е. (Givon M., Mahajan V., Muller E.) Software piracy: Estimation of lost sales and the impact on software diffusion // Journal of Marketing. – 1995. – V. 59. – P. 29-37.
3. Полянський Ігор. Війни між Microsoft і вільним програмним забезпеченням закінчилися. – Доступний з <http://kameniar.lnu.edu.ua/?p=410>
4. Деванбю П., Стаблбайн С. (Devanbu P. T., Stubblebine S.) Software engineering for security: A roadmap // Proceedings of the Conference on The Future of Software Engineering: International Conference on Software Engineering: Limerick, Ireland, June 04–11, 2000. – Piscataway, USA : IEEE, 2000. – P. 3–22.
5. Козырев А.Н. Экономика пиратства: Создание и уничтожение стоимости: Тезисы к обсуждению. – М. : ЦЭМИ РАН, 2008. – Доступный с http://www.labrate.ru/kozyrev/voprosy_dlya_obsuzhdeniya_cemi_25-06-2008.htm.
6. Світові витрати на ІТ, структура і прогноз ситуації до 2019 року. – Доступний з <http://parus.ua/ua/133/1620/> Станом на [18.02.2016]
7. Середа С. А. Анализ рисков и минимизация потерь от нелегального распространения программных продуктов : дисс. ... канд. экон. наук: спец. 08.00.13 – Математические и инструментальные методы экономики. – М., 2005. – 165 с.
8. Середа С. А. Экономический анализ поведения участников рынка программного обеспечения // ИНФОРМОСТ: Радиоэлектроника и телекоммуникации. – 2002. – № 6 (24). – С. 4-9.
9. Чому Україна вже не "головний у світі злодій"? – Доступний з <http://www.telekritika.ua/daidzhest/2015-05-06/106801>

Розробка мікросоціальної мережі з використанням мови програмування Python і програмного каркасу Django

Ханас Д.

Львівський національний університет імені Івана Франка, dkhanas@gmail.com

The report examined the development of micro-social networks using the Python programming language and application framework Django.

Python - потужна і проста для вивчення мова програмування. Вона дає змогу використовувати ефективні високорівневі структури даних і пропонує простий, але ефективний підхід до об'єктно-орієнтованого програмування. Поєднання витонченого синтаксису, динамічної типізації в інтерпретованій мові робить Python ідеальною мовою для написання сценаріїв та прискореної розробки додатків в різних сферах і на більшості платформ.

Django - вільний програмний каркас для Веб-додатків мовою Python, що

використовує шаблон проектування MVC. Проект підтримується організацією Django Software Foundation.

Веб-сторінка на Django будується з одного або декількох додатків, які рекомендується робити окремо. Це одна з істотних архітектурних відмінностей цього каркасу від деяких інших. Один з основних принципів каркасу – DRY(від англ. “Don't repeat yourself” – “Не повторюйте себе”)

Розглянемо основні можливості Django:

- вбудований інтерфейс адміністратора з уже наявними перекладами багатьма мовами;
- диспетчер URL на основі регулярних виразів;
- розширювана система шаблонів з тегами та наслідуванням;
- система кешування (процес запису даних в кеш-пам'ять для зберігання часто використовуваних команд і даних);
- Інтернаціоналізація (процес адаптації продукту, такого як програмне, або апаратне забезпечення, до мовних і культурних особливостей регіону (регіонів), відмінного від того, в якому розроблявся продукт);
- архітектура застосунків, що під'єднуються, які можна встановлювати на будь-які Django-сайти;
- «generic views» - шаблони функцій контролерів;
- авторизація та аутентифікація, під'єднання зовнішніх модулів аутентифікації таких як LDAP, OpenID тощо;
- система фільтрів («middleware») для побудови додаткових обробників запитів, наприклад внесені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій;
- бібліотека для роботи з формами (наслідування, побудова форм за існуючою моделлю БД);
- вбудована автоматична документація за тегами шаблонів та моделями даних, доступна через адміністративний застосунок.

Зручність Django дає змогу швидко і якісно розробляти Веб-додатки, прикладом є написано мною мікросоціальна мережа (рис.1.)

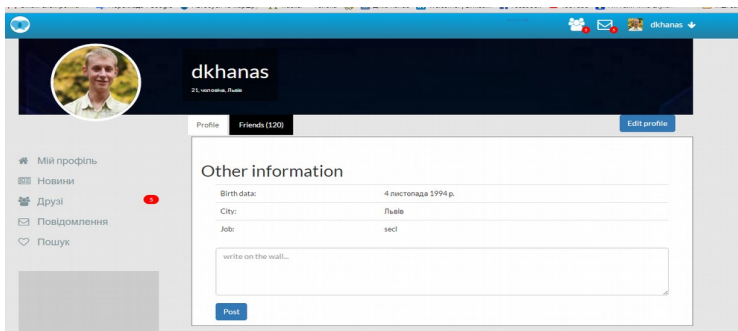


Рис. 1. Стіна користувача мікросоціальної мережі.

Оскільки Django є програмним каркасом з відкритим кодом, то код мого додатку я виклав в своєму репозиторії на Github

Dkhanas fix user from news page		Latest commit ae934d1 on 20 Jan
idea	changed video_admin	3 months ago
auths	fix user from news page	3 months ago
dialogs	Added new style in base	7 months ago
locale	fixed localization	6 months ago
media/avatars/01	last_changes	10 months ago
microsocial	fix user from news page	3 months ago
news	fix user from news page	3 months ago
templates/flatpages	refactor templates	6 months ago
users	added layout user profile	3 months ago
videomain	changed video_admin	3 months ago
gltignore	test_upload	9 months ago
Procfile	Added new style in base	7 months ago
README.md	Revert "Revert "Initial commit""	11 months ago
manage.py	last_changes	10 months ago
requirements.txt	added video on video url	3 months ago
README.md		

Рис. 2. Головний каталог проекту мікросоціальної мережі на Github

На основі простої і надійної платформи Django мовою Python можна розв'язувати поставлені задачі всього кількома рядками коду. Автори, досвідчені творці, описують всі методи, інструменти та концепції, які потрібно знати, щоб нормально використовувати Django 1.8, включно з усіма основними особливостями новоспеченої версії. Також можна додати, що вся документація по Python і Django є у відкритому вигляді за посиланнями <https://docs.python.org/2/> та <https://docs.djangoproject.com/en/1.8/> Це повне керівництво починається з вступу в Python, після цього докладно обговорюються основні компоненти Django (моделі, подання та шаблони) і порядок організації взаємодії між ними. Описуються методи розробки певних програм: мікроблог, фотогалерея, система керування вмістом, інструмент публікації фрагментів коду з підсвічуванням синтаксису.

Джерела:

1. Dzheff Forse, Pol Bisseks, Uesli Chan DJANGO. Developing Web Applications PYTHON / DJANGO.
2. Text Mining for Social Network/Ming Zhou, Xiaohua Liu, Furu Wei
3. The use of Network Analysis tools for dimensionality reduction in Text Mining / Agnieszka Stawinoga, Simona Balbi
4. Holovaty The Definitive Guide to Django
5. Mark Lutz Learning Python
6. Mark Lutz Programming Python

До питання використання можливостей відкритої програмної бібліотеки компонентів при вивченні роботи із сокетами
Харченко В.М., Харченко М.В.

*Ніжинський державний університет ім. Миколи Гоголя, volmkhar@gmail.com;
Київський національний університет ім. Тараса Шевченка*

The authors have considered a capability of using open components library Indy 10 in Lazarus. They have described the features which are using blocking sockets when sending files in the network. Advantages of usage of Synapse library in network programming are listed.

При вивченні мережевого програмування на базі сокетів автори пропонують використовувати різні мови програмування. Зокрема, в [1] пропонується використовувати C++, в [2] – Java, а в [4] – Delphi для програмування в мережі. Якщо ж студенти вивчали програмування на основі Lazarus, то в цьому середовищі можна виконувати і завдання робіт з мережевого програмування. Опис відкритої програмної бібліотеки компонентів Indy 9, 10 можна знайти в [3; 4]. Ця бібліотека підтримує практично всі популярні мережеві протоколи типу TCP, UDP, SMTP, POP3, NNTP, HTTP тощо, що дає змогу програмувати досить складні мережеві проекти.

Зауважимо, на відміну від Delphi, в Lazarus слід самостійно установити відкриту програмну бібліотеку Indy 10. Це можна вважати своєрідною платою за безкоштовність: розпакування бібліотеки, рознесення по каталогах `src` і `lazarus`, видозміна файла конфігурації. На відміну від Turbo Delphi 2006 чи Delphi 7, в Lazarus компоненти розбиті на більшу кількість закладок панелі інструментів: `Indy Clients`, `Indy Servers`, `Protocol` тощо. Це, на наш погляд, тільки полегшує пошук потрібної компоненти.

При виконанні такого роду лабораторних робіт до основних завдань відносять: пересилання одного або кількох файлів, пересилання повідомлень, пошук потрібної інформації в мережі Інтернет.

При розв'язуванні цих завдань з використанням Indy 10 доводиться наголошувати студентам, що у цій бібліотеці використовуються блокуючі сокети [3, с. 26] – на відміну від `Winsock`. При багатьох перевагах використання такого режиму є й недолік – виклик такого сокета не повертає управління до того часу, поки не буде виконана поставлена задача. А тому відбувається “заморожування” користувацького інтерфейсу. Це особливо часто трапляється під час пересилання файлів, розмір яких більший за 4 МВ. Щоб усунути дану проблему – необхідно використовувати в програмі компоненту `TIdAntiFreeze`.

Напевно, найскладніше дається програмування передачі кількох файлів. Обговорюючи написання цього алгоритму доводиться наголошувати на важливості використання конструкції обробки виняткових ситуацій типу `try/finally`, особливо у разі розробки сервера. Наголошувати доводиться і на

потребі перевірки відкритого потоку на атаку. Оскільки в цей момент деструктивні дії можуть привести до непередбачуваних наслідків. Під час написання серверної частини передачі файлу доречним буде шаблон:

```
procedure TForm1.IdTCPServer1Execute(AThread: TIdPeerThread);
var s:string;
    F: TFileStream;
begin
    s:=AThread.Connection.ReadLn;           // шлях, куди качати файл
    //перевірка s на атаку
    .....
    F:= TFileStream.Create(memo1.Lines.Strings[memo1.Lines.Count-1], fmCreate);
    // створюється файл
    try
    AThread.Connection.ReadStream(F, -1, true); // потік, що передається клієнтом
    finally
    end;
    AThread.Connection.Disconnect;
end;
```

Тоді досить легко здійснити програмування передачі кількох файлів від сервера до клієнта:

```
procedure TForm1.Button1Click(Sender: TObject);
var F: TFileStream;
    i: integer;
begin
    for i:=0 to memo1.Lines.Count-1 do
    begin
        IdTCPClient.Connect;
        IdTCPClient.WriteLn(memo2.Lines.Strings[i]); // шлях куди копіювати
        F:= TFileStream.Create(memo1.Lines.Strings[i], fmOpenRead);
        // що передавати
        try
        IdTCPClient.WriteStream(F);
        finally
        F.Free;
        end;
        IdTCPClient.Disconnect;
    end;
end;
```

Отже, використовуючи можливості більш ніж 100 компонентів бібліотеки Indy 10 можна створювати сучасні програми для роботи в мережах. Проте частина із даних компонентів функціонує ненадійно. Тому можна запропонувати студентам використовувати відкриту бібліотеку Synapse [5], яку в Lazarus можна використовувати без встановлення.

Джерела:

1. Григор'єв В.М. Лабораторний практикум із програмування у мережах TCP/IP за допомогою бібліотеки Winsock операційних систем Windows: Навч.-мет. посіб. [Текст] / В.М. Григор'єв. – Дніпропетровськ: ДНУ, 2009. – 148 с.
2. Методичні вказівки до лабораторних робіт з дисципліни «Проектування комп'ютерних мереж» для студентів фаху 7.05010201, 8.05010201 «Комп'ютерні системи та мережі» та 7.05010203, 8.05010203 «Спеціальні комп'ютерні системи», усіх форм навчання. Програмування. Розробка додатків. [Текст] / Укл.: Г.Г. Киричек, Т.М. Семерюк. – Запоріжжя: ЗНТУ, 2013. – 50 с.
3. Хувер Ч. Глубины Indy. [Текст] / Чад Хувер, Хади Харрири. – Atozed Computer Software Ltd, 2006. – 128 с.
4. Фленов М. Программирование в Delphi глазами хакера [Текст] / М. Фленов. – СПб.: БХВ-Петербург, 2003. – 368 с.
5. Synapse. [Електронний ресурс]. – Режим доступу: <http://synapse.ararat.cz/doku.php>

Системно-концептуальний підхід щодо побудови та функціонування комп'ютерно-орієнтованого середовища навчання
Хошаба О.М., Гриб'юк О.О.

Вінницький національний технічний університет, Інститут інформаційних технологій і засобів навчання НАПН України, khoshaba@mail.ru

It is possible to use the system-conceptual approach to construction and operation of computer-based learning environment. It is shown that the main factors of the implementation approach is to define the problems, goals and objectives of previous studies. The success of the implementation of system-conceptual approach is to use freeware licensed software.

Введення

Сучасний етап розвитку моделей побудови комп'ютерно-орієнтованого середовища навчання формує системний комплексний підхід що визначає взаємозв'язок навчального процесу, архітектури систем обробки даних, захисту інформації та технології їх функціонування. В цьому випадку, системність підходу щодо технології проектування та впровадження систем комп'ютерно-орієнтованого середовища навчання виступає як регулярний процес що здійснюється на певних етапах функціонування життєвого циклу інформаційної системи.

Сутність системно-концептуального підходу

Системно-концептуальний підхід визначає стратегію дій. Концепція побудови комп'ютерно-орієнтованого середовища навчання — це система поглядів на сутність, цілі, принципи та організацію навчального процесу на сучасному етапі розвитку суспільства.

Під системністю розуміється пояснювальний принцип наукового пізнання, що вимагає дослідження явищ у їхній залежності від внутрішньо зв'язаного цілого, яке вони утворюють [1-3]. Використання цього основного принципу в технології побудови комп'ютерно-орієнтованого середовища надає можливість наукового обґрунтування структуризації навчального процесу, враховуючи їх взаємні зв'язки та впливи.

Сутність системно-концептуального підходу щодо дослідження і вирішення складних проблем до яких відноситься побудови та функціонування комп'ютерно-орієнтованого середовища навчання полягає в наступному:

- системний розгляд сутності проблеми що розробляється;
- розробка моделі проблеми і обґрунтування повної і несуперечливої концепції її рішення, складання цілей і постановки завдань;
- системне використання методів моделювання процесів і явищ що досліджуються.

До основних вимог щодо побудови комп'ютерно-орієнтованого середовища

навчання відноситься [4,5]:

- використання ліцензійного та вільнопоширеного програмного забезпечення;
- наявність сучасних вільнопоширених ефективних методів захисту та визначення продуктивності інформаційних ресурсів;
- періодичне проведення аналізу безпеки та продуктивності інформаційних систем;
- ефективне використання серверного обладнання та робочих станцій навчальних класів, мережевого та мультимедійного обладнання;
- наявність файлового сховища що призначене для створення backup файлів серверу комп'ютерно-орієнтованого середовища навчання.

Задачі системно-концептуального підходу

До основних задач концепції побудови комп'ютерно-орієнтованого середовища навчання належать:

- надання користувачам якісної програмно-технічної підтримки за допомоги сучасних інформаційних технологій на основі корпоративних мереж, розподілених систем з дистанційного навчання, хмарних технологій, засобів захисту інформації [6];
- забезпечення надійності проведення дистанційного навчання внаслідок включення до складу платформи наявної розподіленої системи [7];
- ефективне супроводження програмно-технічних комплексів з визначенням появ "вузьких місць" у продуктивності обчислювальних систем [8];
- можливе дистанційне керування серверним та клієнтським обладнанням віддалених регіонів з використанням сучасних засобів криптографії;
- ефективне використання придбаної комп'ютерної техніки внаслідок вільнопоширених програмних продуктів [9];
- поточне проведення аналізу безпеки, якості та продуктивності інформаційної системи.

Реалізація комп'ютерно-орієнтованого середовища навчання

Побудова та забезпечення функціонування комп'ютерно-орієнтованого середовища навчання – це комплексна проблема, для вирішення якої потрібне поєднання заходів законодавчого, адміністративного, процедурного і програмно-технічного рівнів.

Законодавчий рівень є найважливішим для забезпечення функціонування комп'ютерно-орієнтованого середовища навчання. Розробка і прийняття правових питань покликані регулювати питання використання інформаційної структури і телекомунікацій, доступу до приватної та конфіденційної інформації, авторського право, захисту інформації від несанкціонованого

доступу та витоку технічними каналами, захисту інформації телекомунікаційних мереж від неправомірних дій, тощо.

До процедурного рівня відносяться заходи безпеки, що реалізуються адміністраторами системи, програмістами та користувачами.

Можна виділити наступні групи процедурних заходів:

- підтримання працездатності розробленої платформи;
- управління персоналом;
- реагування на порушення режиму безпеки функціонування системи;
- планування профілактичних та відновлювальних робіт.

Основою програмно-технічного рівня є такі механізми безпеки:

- забезпечення високої якості, продуктивності, доступності;
- управління доступом до інформаційних ресурсів;
- протоколювання і аудит (моніторинг);
- ідентифікація та аутентифікація користувачів.

Приклади архітектур (параметрів роботи) комп'ютерно-орієнтованого середовища

Наведемо приклади архітектур (параметрів роботи) комп'ютерно-орієнтованого середовища навчання на основі вільнопоширених ліцензійних програмних продуктів.

Приклад архітектури (параметрів роботи) сервера комп'ютерно-орієнтованого середовища навчання на основі вільнопоширених програмних продуктів:

1. Операційна система сервера: Linux Debian 8.0 (jessie), ядро - more than 4.0 файлова система ext4, віртуальна система KVM, метод розподілу простору жорсткого диска за логічними томами - LVM.

2. Операційна система віртуальних серверів: Linux CentOS 7.1, ядро - more than 4.0, робочий стіл Gnome, файлова система ext4, віртуальна система KVM, метод розподілу простору жорсткого диска за логічними томами - LVM.

3. Віддалений захищений (безпечний) доступ до сервера: консольний - на основі ssh, GUI - на основі Virtual Machine Manager або інших.

4. Система криптографії жорсткого диска, окремих файлів і розділів.

5. Системи безпечної роботи з потоками даних комп'ютерних мереж: Firewall, VPN-сервер.

6. Основний навчальний сервіс: на основі moodle;

7. Основні сервіси: web-сервер (використання відкритого - http і захищеного протоколу передачі даних https), ftp-сервер (використання відкритого - ftp і захищеного - sftp протоколу передачі даних), система управління базами даних mysql або postgresql, система управління зовнішніми і внутрішніми іменами доменів dns, dhcp-сервер (призначення ip адреси з прив'язкою на mac), поштовий сервіс, проксі сервіс.

8. Система резервного копіювання і відновлення даних, заснована на шифруванні інформації.

9. Відмовостійкість: час відновлення сервера - від 1:00 до 1 доби.

Приклад архітектури (параметрів) робочих станцій навчальної аудиторії на основі вільнопоширених програмних продуктів:

1. Операційна система: Linux Mint 17.2 (rafaela), ядро - more than 4.0, робочий стіл Cinnamon, файлова система ext4, метод розподілу простору жорсткого диска по логічним томам - LVM.

2. Віддалений захищений (безпечний) доступ до робочих станцій: консольний - на основі ssh, GUI - на основі Virtual Machine Manager.

3. Система криптографії жорсткого диска, окремих файлів і розділів.

4. Системи безпечної роботи з потоками даних комп'ютерних мереж: Firewall, VPN-клієнт.

5. Основні сервіси: web-клієнти (використання відкритого - http і захищеного протоколу передачі даних https), ftp-клієнти (використання відкритого - ftp і захищеного - sftp протоколу передачі даних), поштовий клієнт, офісні додатки - текстовий і табличний процесор (з перевіркою правопису англійської, російської та української мов).

6. Система резервного копіювання і відновлення даних, заснована на шифруванні інформації.

7. Відмовостійкість: час відновлення робочих станцій - від 1 до 4 діб.

Таким чином, існує важливість використання системно-концептуального підходу щодо побудови та функціонування комп'ютерно-орієнтованого середовища навчання. Головними чинниками впровадження підходу є визначення проблеми, мети та завдань чинних досліджень. Успіхом впровадження системно-концептуального підходу є використання вільнопоширених ліцензійних програмних продуктів.

Джерела:

1. Ракитов А.И. Философские проблемы науки: Системный подход. – М., 1977. – 270 с.
2. Карпович В.Н. Проблема, гипотеза, закон. – Новосибирск: Наука, 1980. – 176 с.
3. Кочергин А. Н. Научное познание: формы, методы, подходы. – М.: Изд-у МГУ, 1991. – 79с.
4. Хошаба А.М. Опыт использования облачных технологий в учебном процессе. // 8 міжнародна конференція "Інтернет – Освіта – Наука – 2012" - вересень-жовтень 2012. – Вінниця, ВНТУ. – С.229 – 232.
5. Хошаба А.М., Ткачук Л.В. Использование облачных вычислений как перспективного направления в дистанционном обучении. // Збірник наукових праць XVI Міжнародної науково-практичної конференції "Інформаційні технології в економіці, менеджменті і бізнесі.": Том 2. Наукові праці-Київ, 2010.-С. 129-131.
6. Хошаба А.М., Романюк А.Н. Проблемы и решения внедрения служб безопасности информационных ресурсов в компьютерных системах и сетях // Електронні інформаційні ресурси: створення, використання, доступ»: //Збірник матеріалів Міжнародної науково-практичної Інтернет-конференції. - Вінниця, 2015 р. - С. 397-401.
7. Хошаба О.М. Дослідження надійності функціонування систем з керування потоками даних у комп'ютерних мережах. // II Міжнародна науково-практична конференція

"Інформаційні технології в освіті, науці і техніці": Тези допов.-Черкаси, 2014.-С. 107-108.

8. Хошаба А.М. Построение информационной системы для компьютерного моделирования производительности программно-технического комплекса // Вестник Херсонского национального технического университета №3(56), 2015.-Херсон.-С.- 303-307.

9. Хошаба О.М., Романюк О.Н. Досвід використання OPEN SOURCE технологій у Вінницькому національному технічному університеті. // V Міжнародна науково-практична конференція "FOSS Lviv 2015": Тези допов.-Львів, 2015.-С. 102.

Захист зображень у друкованих документах

Кліщ Ю.І.

Інститут комп'ютерних наук та інформаційних технологій Національного університету "Львівська політехніка", chlichch@ukr.net

Developed software protects printed documents. Application is written using Qt framework with C++ language. Protection is based on modifying the bitmap and converting it into a grid which changes by offset, amplitude or width. Also this software can protect documents of strict accountability.

Одним з видів графічного способу захисту друкованої продукції є мікрографіка. Різновид методу базується на ефекті прихованого зображення на основі високої роздільної здатності ліній. Візуально мікрографіка сприймається як неперервна лінія, хоча складається зі штрихів, які можна побачити лише при значному збільшенні. Як захисний елемент під час формування зображення широко використовують захисні сітки, які багаторазово повторюються. Ці елементи можуть утворювати періодичні та аперіодичні захисні сітки. У разі формування площини зображення документу у вигляді аперіодичних захисних сіток рівень захисту від підробки значно вищий, ніж у разі використання звичайних періодичних захисних сіток [1].

Розроблене програмне забезпечення для захисту зображення. Програмне забезпечення вхідне зображення перетворює у захищене і готове для друку. Відповідно до параметрів одиничної лінії формується сітка ліній, яка, разом з параметрами зміни формує захист. На виході отримуємо растрове або векторне захищене зображення. Ужиток реалізовано за допомогою програмного каркасу Qt мовою C++. Схема роботи програми (для формування захисту) зображена на рис. 1 і показує основні функціональні частини захисту. Вхідними даними для ужитку є вхідне зображення, параметри зміни та параметри одиничної лінії.

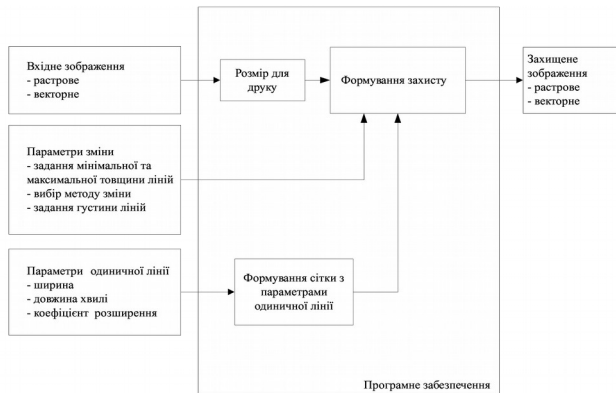


Рис. 1. Структурна схема захисту зображення

У програмному забезпеченні реалізовано три різні методи утворення захисту. Залежно від вибраного методу, лінія може як просто зміщуватися під час накладання на зображення, так і перетворюватись у деформовану синусоїду з різною амплітудою коливаль. Також є можливість потовщення лінії в місцях накладання її на зображення.

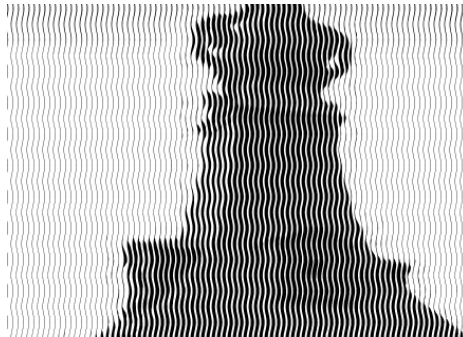


Рис. 2. Приклад роботи методу зміни товщини лінії

Методи утворення захисту

1. Метод зміни товщини

Одним із методів захисту зображень є використання сітки ліній, які для передачі градацій сірого використовують зміну своєї товщини (рис. 2). Лінія будується із базової і зміщеної кривих. Базова крива є графіком функції синус з такими параметрами як частота хвилі і амплітуда.

2. Метод зміщення

Цей метод дає змогу захистити зображення, використовуючи виділення контурів. Він найкраще підходить для контрастних фотографій, зображень що містять текст, логотипів. Лінія будується на основі функції синус та

використовує такі самі параметри як і метод зміни товщини.

Контур - лінія, що окреслює форму предмета. Для виділення контуру, лінія, у місці, де виникає високий контраст, зміщується, що дає змогу візуально побачити обрис об'єкта. Рівень зміщення залежить від різниці градацій сірого сусідніх пікселів на вхідному зображенні (рис. 3).

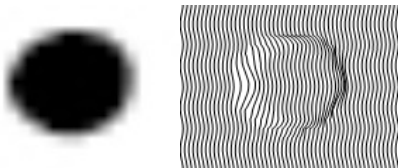


Рис. 3. Приклад методу виділення контуру у місці високого контрасту

Особливістю даного методу є те, що градація сірого вхідного зображення не передається, що дає змогу посилити захист яскравих зображень.

3. Метод зміни амплітуди

Відображення градацій сірого у цьому методі відбувається за рахунок зміни амплітуди тригонометричної функції синус.

Базова амплітуда задається для того, щоб крива мала вигляд синусоїди у разі передачі білого кольору. Коефіцієнт збільшення амплітуди дає змогу плавно змінювати градації сірого на захищеному зображенні. Це покращує відображення градієнтів, що показано на рис. 4.

Формування сітки

Алгоритм формування захисної сітки передбачає високу швидкодію, помірне використання ресурсів комп'ютера та універсальність вхідних зображень. Для побудови захисної сітки вхідне зображення дискредитується відповідно до розміру друку за допомогою методу білінійної інтерполяції, за допомогою якого пікселі додаються шляхом усереднення значень кольорів оточуючих пікселів.

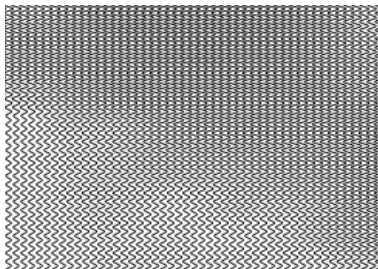


Рис. 4 Приклад відображення градієнту сірого

Цей метод інтерполяції є оптимальним за співвідношенням швидкодія/якість. Результатом є зображення, яке відповідає розміру вихідного і є джерелом для побудови ліній.

Після цього на основі дискретизованого зображення формується масив

перетворених у градації сірого пікселів. Потім, відповідно до параметрів формується сітка. Параметри формування сітки містять густину ліній (кількість вертикальних ліній оригінального зображення на одну лінію захищеного зображення), розмір зображення для друку і метод захисту.

Сформована сітка надалі використовується для відображення ліній на векторному або растровому зображенні. Під час відображення лінії згладжуються за допомогою кривих Без'є, що дає змогу уникнути кутів у векторних зображеннях.

Збереження

Захищене зображення можна зберегти як векторне або як растрове.

Збереження векторного зображення відбувається у форматі SVG (англ. Scalable Vector Graphics). SVG - універсальний відкритий формат векторних графічних файлів.

Растрове зображення за замовчуванням зберігається у форматі TIFF для високоякісного поліграфічного відтворення. Також є можливість збереження в інших популярних форматах як JPEG, PNG та BMP.

Джерела:

1. Назаркевич М. А. Методи підвищення ефективності поліграфічного захисту засобами Ateb-функцій. Назаркевич М.А. Монографія. Львів: Видавництво Львівської політехніки, 2011. 188 с.

Статистичний пакет R

Кобильник Т.П.

*Дрогобицький державний педагогічний університет імені Івана Франка,
taras2408@mail.ru*

R – вільне програмне середовище з відкритим кодом, що поширюється в рамках проекту GNU. Аналізується пакет R – мова і програмне середовище, орієнтоване на розв'язування задач статистичного аналізу. Наводяться переваги та недоліки R. Вказуються на можливості операції імпортування/експортування даних в різні програми та інтеграцію R з LaTeX та офісними програмами.

Для експериментальних досліджень (наприклад, психолого-педагогічних, соціологічних, економічних тощо) використовуються різні статистичні методи для перевірки висунутих гіпотез, побудови статистичних моделей об'єктів, явищ, закономірностей і процесів.

Стандартні статистичні методи опрацювання експериментальних даних реалізовані в електронних таблицях (Lotus, QuatroPro, MS Excel, OpenOffice.org Calc і ін.), спеціалізованих пакетах (R, SPSS, Statistica і ін.), системах комп'ютерної математики (Mathematica, Maple, Maxima та ін.).

Перед користувачами постає питання вибору програмного забезпечення

для дослідження статистичних даних. Очевидно, що найкращим є варіант, що поєднує в собі необхідні функціональні можливості, високу якість роботи і помірну ціну. Тому у разі вибору пакета необхідно враховувати:

1. відповідність характеру розв'язуваних задач;
2. обсяг експериментальних даних;
3. вимоги, що ставляться до кваліфікації користувача (рівень знань в галузі статистики і програмування);
4. наявне комп'ютерне забезпечення.

Найпростіші методи статистичного аналізу реалізовані в електронних таблицях, що належать до програмного забезпечення загального призначення. Тому значна частина (якщо не всі) користувачів здійснює опрацювання статистичних даних саме за допомогою них. Проте, незважаючи на переваги електронних таблиць, повноцінне статистичне опрацювання результатів неможливе: це програмне забезпечення загального призначення, а не спеціального (наукового). Набір статистичних тестів невеликий, багато методів, особливо багатовимірних, відсутні, немає спеціалізованої системи звітів. Це пояснюється тим, що статистика – не першочергова функція електронних таблиць. Цих недоліків позбавлені статистичні пакети, які значно переважають табличні процесори за об'ємом і якістю реалізованих в них статистичних методів.

Необхідно відзначити, що виділяють мінімальний набір статистичних методів аналізу, який внесений в усі статистичні пакети, зокрема описова статистика, непараметрична статистика, дисперсійний, кореляційно-регресійний, кластерний, факторний, дискримінантний аналізи.

Усім цим характеристикам відповідає статистичне середовище R [1]. Перш за все R – мова програмування для статистичного опрацювання даних та графічного їх подання. Це вільне програмне середовище з відкритим кодом, що поширюється в рамках проекту GNU. Додаткової популярності R дало створення системи зберігання і розповсюдження пакетів – CRAN (Comprehensive R Archive Network — <http://cran.r-project.org>). Статистичні алгоритми зазвичай виконуються у вигляді скриптів і зібрані у пакети (packages) R. Під час інсталяції разом з базовою програмою інсталиються й основні пакети, у яких реалізовано найбільш популярні методи статистичного аналізу. Частина цих пакетів автоматично завантажується під час запуску R. Решту можна завантажити, використовуючи функцію library.

Виокремлюють такі переваги пакета R як:

- R є вільнопоширюваним програмним забезпеченням: кожен може його безплатно скачати з сайту <http://www.r-project.org>;
- є реалізації під операційні системи сімейств Microsoft Windows, Mac OS X, Linux;
- базова комплектація R займає небагато місця на ЖМД і містить всі функції, необхідні для проведення статистичного аналізу;
- завжди можна додатково встановити допоміжні пакети з необхідними

функціями;

- хороша графічна візуалізація подання даних та результатів їхнього аналізу;
- можливість самостійного написання необхідних функцій;

Серед недоліків необхідно відзначити те, що на відміну від більшості комерційних програм, R має не графічний інтерфейс, а інтерфейс командного рядка, тому треба знати необхідні для роботи функції та синтаксис мови програмування (цей недолік дещо нівелюється встановленням пакету [Rcmdr](#)).

Дані, окрім введення з клавіатури, можна імпортувати з txt- та xml-файлів, Веб-сторінок, MS Excel, SPSS, SAS, Stata, систем керування базами даних, включно з Microsoft SQL Server, Microsoft Access, MySQL, Oracle, PostgreSQL. Також можна дані експортувати з R, зокрема в текстовий файл з розділювачами, таблицю Excel або файл іншого статистичного пакету (SPSS, SAS, Stata).

Зазвичай будь-яке дослідження завершується публікацією результатів. Для інтеграції програмного коду R та результатів дослідження в LaTeX використовується пакет `sweave`, а для експорту в документи формату ODF (Open Documents Format) – пакет `odfWeave`.

Джерела:

1. Роберт И. Кабаков R в действии. Анализ и визуализация данных в программе R / пер. с англ. Полины А. Волковой. – М.: ДМК Пресс, 2014. – 588 с.

Software system for parallel usability testing

Kostiuk D.A., Latiy O.O., Markina A.A.

Brest State Technical University, dmitriykostiuk@gmail.com

An open source software suite is presented targeted at evaluation of the effectiveness of human-computer interaction by monitoring physical state of the user. An internal architecture and capabilities of the system are reviewed.

Current approaches for evaluation of software usability (as ones based on cognitive schemes analysis, so ones based on expensive measuring tools like eye capture and medical-grade electroencephalography) can't be used without an expert's judgment after examination of collected data [1]. This way of evaluation is slow, prone to human errors and even to biased view in some cases. However, last years have brought into wide usage some computer-connected measuring devices, like pulsometers and simple electroencephalographs. Being targeted at sports and entertainment, these devices at the same time have enough precision to provide data of user's physical state during working with software. This provides the possibility to create totally new style of usability evaluations: fully automatic numerical estimation based common tests set and data acquired from popular measuring devices.

While carrying out some research in usability of GNU/Linux GUI, we have created specialized package of testing instruments and have used it in several experiments, which demonstrated proposed approach to be promising. These experiments have resulted in an integrated software suite for such tasks as running tests, acquiring measured data and accumulating them in central database, and what is more, providing parallel testing of several users to speed up experiments. It was implemented as an open source project titled “UXDump Suite” and is available at <https://bitbucket.org/AsyaAliset/uxdump>.

The architecture of the system is presented at figure 1. The main part of the system is written in C++ and Qt, and currently is compiled for two platforms, GNU/Linux and MS Windows, to compare usability of these platforms software.

Being targeted at parallel usage at several computers, it relies on MySQL database to store measurement results, which is supposedly residing on a separate server. Each computer participating in research has a copy of UXDump Suite, executed to carry on test with chosen measuring modules (for user-wearable sensors) chosen testing module and filled-up user's data (figures 2-b and 2-c). Same application is used to visualize portion of data (figure 2-a) with one of typical plot styles. Chosen data can be exported either as graphics (SVG and bunch of raster formats are supported) or as CSV table.

The architecture of the system is presented at figure 1. The main part of the system is written in C++ and Qt, and currently is compiled for two platforms, GNU/Linux and MS Windows, to compare usability of these platforms software.

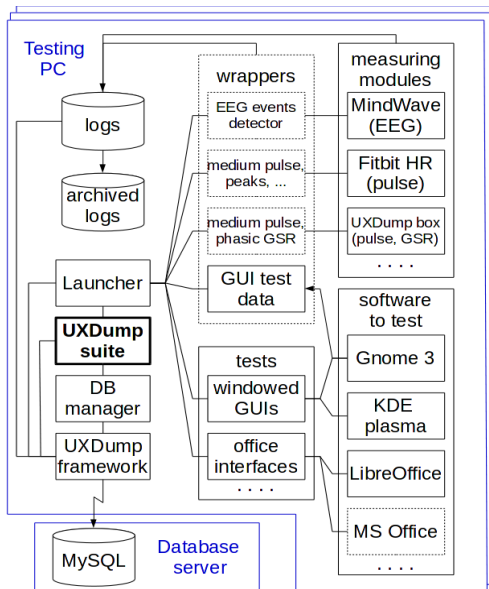


Figure 1 — Architecture of the system

Being targeted at parallel usage at several computers, it relies on MySQL database to store measurement results, which is supposedly residing on a separate server. Each computer participating in research has a copy of UXDump Suite, executed to carry on test with chosen measuring modules (for user-wearable sensors) chosen testing module and filled-up user's data (figures 2-b and 2-c). Same application is used to visualize portion of data (figure 2-a) with one of typical plot styles. Chosen data can be exported either as graphics (SVG and bunch of raster formats are supported) or as CSV table.

UXDump Suite relies on a specific structure of subfolders.

Testing module checks whether the software to test is running (optionally starts it), and supplies the user with tasks to do. In fact these “modules” are subfolders of the `Test` folder, and contain at least one test starting script (a script with “start_” prefix). Currently two groups of tests are implemented. `WindowedGUIs` group includes 3 test variants to evaluate effectiveness of windows management in GUI shells [2], and `OfficeInterfaces` group contains 8 variants to evaluate usability of office applications [4]. Tests from the first group depend on Gnome 3, KDE Plasma Workspace, and Ubuntu Unity, while tests from the second group rely on LibreOffice (Writer and Calc), Calligra Suite (Words and Sheets), Microsoft Office 2007+ (Word and Excel), and two graphical editors (Pinta and Microsoft Paint). Of course, Microsoft-depending tests are available only if UXDump Suite is compiled for Windows. Figure 2-c shows starting dialog of the chosen testing module with specified user login and software (in combobox filled up by scanning starting script names).

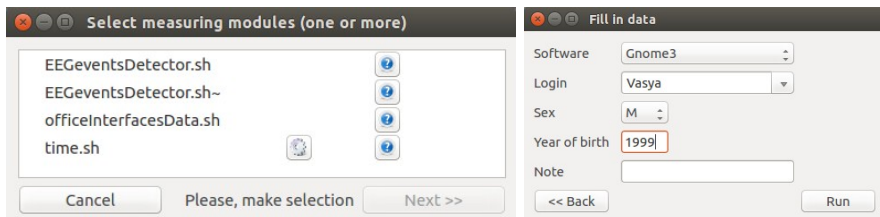
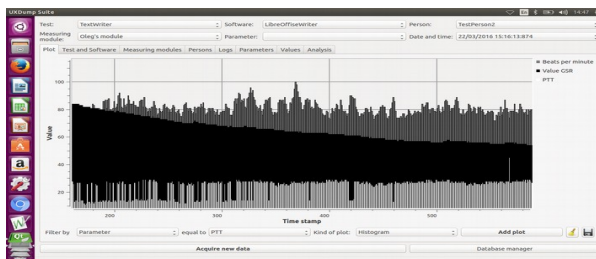


Figure 2. UXDump Suite: starting screen with plot (a), and test launcher asking to choose measuring modules (b) and to fill in user data before running test (c).

Measuring modules accept user's physical state data from wearable devices. In reality they are either executable files or scripts placed in the `Modules` folder. Moreover, some of them are not more than wrappers for real measuring modules, residing in deeper level subfolders. Such a cascade approach allows to separate acquiring raw data and its primary processing. For example, not yet written EEG MindWave module wrapper can be used to detect additional events (eye blinks), while `phasicGSR` wrapper is useful to extract phasic component of the skin galvanic response (GSR).

Currently following modules are designed to be used by the system: `fitbit` to measure heart rate (HR) with Fitbit Charge HR fitness tracker (based on open source `galileo` project), `mindwave` (based on the same-name open source library) to evaluate mind concentration with mass-market Neruosky Mindwave EEG headset, and `uxdumpbox` to acquire HR and GSR with our own open hardware project, the UXDump Box [3]. There are also two quasi-modules, `time` to register the experiment duration, and `gui_test_data` to get additional logs from the WindowedGUIs testing module.

Data are provided by measuring modules as log files in CSV format. File name is specified by the launcher component, which controls the process of testing: asks user to choose measuring modules and one testing module, runs all chosen modules, waits until testing module finishes its work and forces quit of measuring module processes. After data acquisition log files are stored in the local archive for further analysis if needed. CSV parsing and dealing with database is done through the set of libraries, specified on the figure 1 as UXDump framework. The same libraries are used by the DB Manager service application, targeted at administration of measurements stored in a remote database.

References

1. Paas F., Renkel A., Sweller J. Cognitive Load Theory: Instructional Implications of the Interaction between Information Structures and Cognitive Architecture // *Instructional Science*. No. 1-2(32), 2004. – pp. 1–8.
2. Kostiuk D.A., Derechennik S.S., Shitikov A.V., Latiy O.O. Approach to evaluate effectiveness of human-computer interaction with contemporary GUI // Третя міжнародна науково-практична конференція FOSS Lviv 2013: Збірник наукових праць, Львів, 18–21 квітня 2013 р. – Львів, 2013. – С. 85–87.
3. Костюк Д.А., Латий О.О. Оценка состояния пользователя с помощью платформы Arduino // Информационные технологии и системы 2014 (ИТС 2014): материалы МНК. Минск, БГУИР, 29 октября 2014 г. – С. 57–58.
4. Костюк Д. А., Латий О. О., Маркина А. А. Инструментальная оценка состояния пользователя в задаче сравнения интерфейсов офисных приложений // XII конференция разработчиков свободных программ. Тезисы докладов. – Калуга, 16-18 октября 2015г. – М.: Альт Линукс, 2015. – С.8–12.

Аб выкарыстанні мабільных платформаў у добраахвотных вылічэннях

Каваленка У.Ю., Касцюк Д.А., Чацвёркіна Г.А.

Брэсцкі дзяржаўны тэхнічны ўніверсітэт, volidik666@gmail.com

An overview of approaches to use mobile devices in volunteer computing related to free/libre projects is presented. Analysis of mobile platform applicability is done from the point of view of processing power and typical use cases, with brief discussion of possible directions to develop mobile-oriented GRID-like computing.

«Добраахвотныя вылічэнні» (ангельск. volunteer computing) гэта выкарыстанне асабістых рэсурсаў карыстальнікаў Internet нейкай GRID-сеткай (SETI@home, праекты distributed.net і інш.). Сёння карыстальніцкія GRID дэманструюць вылічальную магутнасць, параўнальную з топавымі суперкамп'ютэрамі, ахопліваюць задачы матэматыкі, крыптаграфіі, біялогіі, медыцыны, фізікі і інш. Праекты завабляюць удзельнікаў важнасцю вырашаемай задачы, упорам на пазіцыі ў рэйтынгах, электроннымі бэджамі, прывабнымі кліенцкімі праграмамі (напрыклад, экраннай застаўкай, якая падвышае сацыяльны статус ўладальніка кампутара).

У апошні час з'яўляюцца мабільныя кліенты volunteer computing, у першую чаргу для Android - дзякуючы як Linux-аснове, так і асаблівасцям самой платформы, якая змушае канцэнтравать ў мабільнай прыладзе шмат'ядравы працэсар, які мэтанакіравана прастойвае вялікую частку часу. Кліенты ствараюцца альбо мэтавым партаваннем (Android-кліент BOINC), альбо зборкай Linux-версіі: напрыклад, першапрацавіцкі жарт аб вылічальным кластары Google на Nexus One прывёў да з'яўлення некалькіх рэальных аналагаў [1].

Аднак ўключэнне ў GRID мабільных прылад, пры ўсёй іх шматлікасці і шмат'ядравасці, абмяжоўваюць асаблівасці архітэктуры і іх выкарыстання. Архітэктура ARM, якая ўзнікла ў 80-я гады для персанальных кампутараў пад кіраваннем RISCOS, у 90-я гады пераарыентавалася на наладонныя кампутары дзякуючы прастаце, эфектыўнаму выкарыстанню блокаў працэсара і меншаму, у параўнанні з x86, энергаспажыванню і цеплавывыдзяленню. Далейшая аптымізацыя ARM праходзіла з упорам на гэтыя два крытэрыі, і сёння карыстальнікі ведаюць, што «ў ARM і x86 розныя мегагерцы». Праілюстраваць гэтую розніцу можна наглядна, калі параўнаць працягласць выканання тыповых каманд у тактах працэсара (мал. 1-а, светлы фон адпавядае больш высокай прадукцыйнасці). Паказчыкі сямейства ARM зніжаюцца яшчэ больш, калі ўлічыць меншыя тактавыя частоты і разраднасць некаторых інструкцый, і сітуацыю не выправіць нават ўлік вялікага ліку ядраў. Безумоўна, калі дадаткова пранармаваць прадукцыйнасць па рассеяванай магутнасці (мал. 2-б), ARM апынецца безумоўным лідэрам, але гэтыя мела б значэнне, калі б нізкаватныя GRID-сеткі былі хоць колькі-

небудзь запатрабаванья.

	Data transfer	Logical	Integer +/	Integer *	SSE/NEON(VFP)	Special math
ARM Cortex A9	2,66	1	4,16	5,66	10,5	
ARM Cortex A7	3,66	1	5,16	6,66	16,5	
x86 Atom 330	2,66	1	4	19	7	
x86 AMD A10 (Trinity)	2,33	1	3,5	5	5,5	113,33
x86 I7 3xxx	2	1	3,5	3	3	73,33

а)

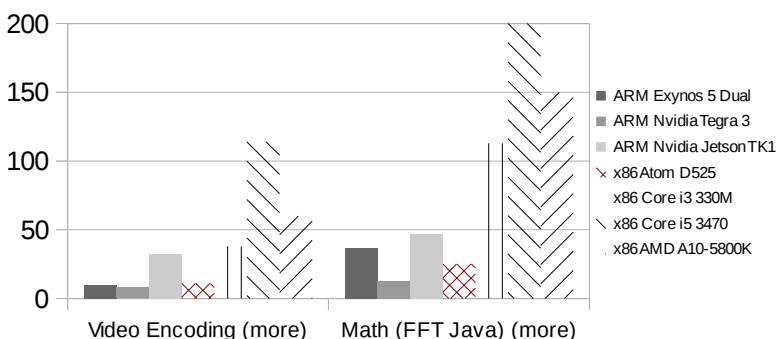
	Data transfer	Logical	Integer +/	Integer *	SSE/NEON(VFP)	Special math
ARM Cortex A9	98 932	263 158	63 259	46 494	25 063	
ARM Cortex A7	136 612	500 000	96 899	75 075	30 303	
x86 Atom 330	46 992	125 000	31 250	6 579	17 857	
x86 AMD A10 (Trinity)	4 292	10 000	2 857	2 000	1 818	88
x86 I7 3xxx	3 846	7 692	2 198	2 564	2 564	105

б)

Мал. 1. Працягласць выканання інструкцый у цыклах працэсара (а) і вынік нармавання па расейскай магутнасці (б)

Варта згадаць, што рэальны разрыў паміж ARM і x86 яшчэ больш з-за аптымізацыі выканання патокаў каманд ў x86 (спекулятыўнае выкананне, прадказанне пераходаў, зліццё і падзел інструкцый і г.д.), але гэта дрэнна паддаецца разлікам, і таму прадукцыйнасць даводзіцца ацэньваць на канкрэтных задачах, якія ўлічваюць, праўда, яшчэ і розніцу ў хуткадзейнасці памяці і дыскавай падсістэмы, як, напрыклад, бэнчмаркі, якія перыядычна публікуюцца рэсурсам phoronix.com (мал. 2).

Працоўны працэс мабільнай прылады, з другога боку, мяркуюе як мага больш доўгае функцыянаванне ад батарэі, і як вынік - мінімальнае выкарыстанне наяўных вылічальных магутнасцяў. Распрацоўнікі Android-кліентаў гэта ўлічваюць: так, кліент BOINC працуе толькі падчас зарадкі і пазбягае перадачы дадзеных праз мабільную сетку.



Мал. 2. Параўнанне архітэктур на рэальных задачах (сямейства x86 адзначана штрыхоўкай; найбольш высокія значэнні адпавядаюць большай прадукцыйнасці)

Такім чынам, апроч меншых вылічальных магутнасцяў, мабільны вузел

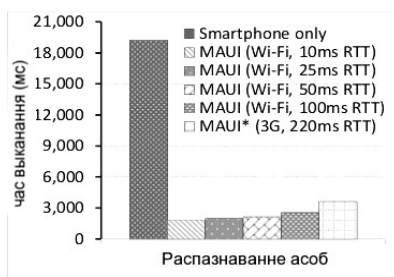
хутчэй за ўсё будзе заставацца ў валандэрскай GRID-сетцы толькі пакуль сетка практычна не нагружае яго вылічэннямі, і таму прымяненне мабільных прылад у складзе GRID носіць эпизадычны характар.

Найбольш падыходныя для мабільных вылічэнняў задачы павінны альбо выкарыстаць вылічальную магутнасць ў выглядзе кароткачасовых пікавых нагрузак (1), альбо здабываць карысць не столькі з вылічальнага рэсурсу, колькі з тэрытарыяльнага роскідку (2) і гетэрагеннасці вузлоў сеткі (3).

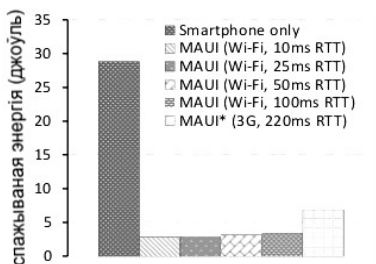
Да задач катэгорыі 1 ставіцца самадыягностыка GRID з мэтай вымярэння дасягальных пікавых нагрузак; аднак гэта ператварае пабудову сістэмы у спартовае мерапрыемства. Задачы тыпу 2 перыядычна абмяркоўваюцца ў прэсе (ужыванне смартфонаў для стварэння размеркаванай сеткі датчыкаў). На жаль, рэальныя прылады не маюць датчыкаў, сетка якіх апынулася б карыснай у рэальных даследаваннях або маніторынгу.

Ёсць прыклады «ня матэматычных» валандэрскіх вылічэнняў. Так, праект Majestic-12 выкарыстоўвае інтэрнэт-канал ўладальніка прылады для пабудовы размеркаванай сістэмы сеткавага пошуку, праект Surveill@Home ацэньвае прадукцыйнасць і адмоваўстойлівасць вэб-сайтаў, а адзін з чатырох метадаў узаемадзеяння, закладзеных у праекце SmartLab, тычыцца выдаленай адладкі праграм на мабільнай прыладзе [1].

Больш перспектыўна ў мабільных добраахвотных вылічэннях зваротная схема - напрыклад, апісаныя ў [3] і [4] спробы дэлегавання частцы нагрузкі мабільнай маламагутнай прылады выдаленаму серверу для памяншэння энэргаспажывання і павышэння прадукцыйнасці ў некаторых тыпах задач. Прыклады канкрэтных абласцей – гульні, складаная матэматыка з багаццем спецыяльных функцый, або задачы, якія цяжка дзяляцца на паралельныя патокі і ў якіх прадукцыйнасць аднаго ядра вельмі крытычна. Мал. 3 паказвае графік прадукцыйнасці пры распазнаванні асоб сіламі толькі прылады і пры выкарыстанні сервера, а мал. 4 - параўнанне спажыванай энэргіі пры розных сцэнарах [3].



а)

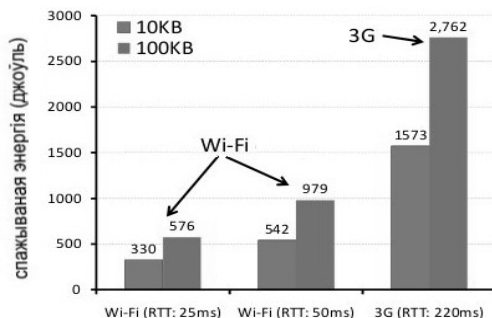


б)

Мал. 4. Час працы алгарытму (а) і спажыванне энэргіі (б) на мабільнай прыладзе пры розных затрымках сеткі (а)

Галоўныя праблемы у дадзеным выпадку - метадыка «падзелу» задачы на кавалкі іншым вузлам сеткі і эфектыўная перадача дадзеных па сетцы. Агульнае ва ўсіх прапанаваных рашэннях - запуск праграмы і на мабільнай прыладзе і на высокапрадукцыйным кампутары, з кіраваннем паслядоўнасцю апрацоўкі. На жаль, для кожнага тыпу задач трэба рэалізаваць асобную метадыку. Для задач рэндэрыngu, напрыклад, у тэставых мэтах прымяняўся падыход, калі кожны кадр аблічвалі на маламагутнай прыладзе але з нізкай колькасцю дэталей, а на высокапрадукцыйнай ў поўным дазволе і з усімі дэталямі. Па сетцы жа перадавалася розніца паміж кадрамі і выраблялася сумяшчэнне. Можна адзначыць і зваротны падыход, калі мабільная прылада аблічвае адзін кадр з поўнай якасцю, а кампутар у гэты час паспявае апрацаваць n кадраў.

Праца модуляў бесправадной сувязі гэтак жа звязана са значным выдаткам энергіі (мал. 5). Аднак з распаўсюджваннем высакахуткасных WiFi-сетак гэта перастае з'яўляцца непераадольнай праблемай.



Мал. 5. Спажыванне энергіі пры рознай якасці сеткі

Літаратура

1. G. Larkou et al. Managing Smartphone Testbeds with SmartLab // Proceedings of the 27th USENIX Large Installation System Administration Conference (LISA '13), Washington D.C., USA. Pp. 115-132. <https://www.usenix.org/conference/lisa13/technical-sessions/presentation/larkou>
2. C. Hu, I. Neamtiu. Automating GUI Testing for Android Applications / 3rd Int. conf. on Advanced Science and Technology, September 27--29, 2011, Seoul, Korea. - pp. 77--83. <http://www.cs.ucr.edu/~neamtiu/pubs/ast11hu.pdf>
3. E. Cuervo et al. MAUI: Making Smartphones Last Longer with Code Offload // MobiSys'10, June 15--18, 2010, San Francisco, California, USA. pp. 49-62. <http://research.microsoft.com/en-us/um/people/ranveer/docs/maui.pdf>
4. E. Cuervo et al. Kahawai: High-Quality Mobile Gaming Using GPU Offload // MobiSys'15, May 18--22, 2015, Florence, Italy. pp. 121-135. <http://homes.cs.washington.edu/~kklebeck/kahawai.pdf>

Використання вільного програмного забезпечення у Львівському університеті імені Івана Франка

Кухарський В., Злобін Г.Г., Рикалюк Р.Є., Шувар Р.

*Львівський національний університет імені Івана Франка,
vitaliy.kukharsky@gmail.com*

В доповіді проведено аналіз використання вільного програмного забезпечення у Львівському національному університеті імені Івана Франка на протязі останніх двадцяти років.

Львівський національний університет імені Івана Франка має вже понад двадцятирічний досвід впровадження та використання вільного програмного забезпечення, напрямки застосування якого дуже різноманітні [1- 6]:

серверні застосування — Linux (Debian, Open SuSE), Unix FBSD;

навчання — операційна система (Debian, Open SuSE), офісний пакет (OpenOffice), засоби програмування (gcc, Kuzya IDE, Qt Creator), математичні пакети (Octave, Labplot), дистанційне навчання (Moodle), технологія термінал-сервер;

студентська наукова робота — операційна система (Debian, Open SuSE), офісний пакет (OpenOffice), засоби програмування (gcc, python, Qt Creator), математичні пакети (Octave, Maxima), системи керування базами даних (MySQL), емулятори апаратних засобів і операційних систем;

наукові дослідження — операційна система (Debian, Open SuSE), офісний пакет (OpenOffice), засоби програмування (gcc, Qt Creator, R), математичні пакети (Octave, Maxima), організація обчислювальних кластерів (Scientific Linux, Kickstarter, Webmin, SGE, Ganglia, OpenMPI, MPICH2, BLAS, FFTW, NorduGrid ARC, Condor, **CUDA 5.0 production release**), засоби розробки (Apache, Ggithub, Java, Postgres, PgAdmin3, Maven3, Jenkins, Spring Framework, Hibernately).

На початках використання ПЕОМ у Львівському університеті питання вибору програмного забезпечення практично не було — як операційна система використовувалась MS-DOS, згодом Microsoft Windows і прикладне програмне забезпечення для них. Про ліцензійність програмного забезпечення практично не дбали, навпаки спостерігалась успадкована від СРСР “героїчна” традиція використання неліцензованих програм. Після підписання у 2003 р. меморандуму між Microsoft і міністерством освіти України в тендерних вимогах на закупівлю закладами освіти засобів обчислювальної техніки з’явилась обов’язкова вимога — програмне забезпечення. Ця обставина декларується кожного разу у вимогах до тендерних закупівель комп’ютерного обладнання. Переважно це ОС Microsoft Windows і (не завжди) Microsoft Office. Решта програмного забезпечення додавалося за потребою і майже завжди без врахування його ліцензійності.

Позаяк вартість мінімального набору ліцензійного ПЗ (Microsoft Windows та Microsoft Office) складала від 10 до 30 відсотків вартості ПЕОМ, розпочалися пошуки альтернативи з метою скорочення вартості одного робочого місця в результаті чого у 2015 р. закупівля ПЕОМ здійснена з ОС Linux. Ситуація з неліцензійним програмним забезпеченням у Львівському університеті загострилась після отримання ректором університету наприкінці 2012 р. листа від представництва фірми Microsoft в Україні з пропозицією придбати у 2012/2013 роках необхідну кількість ліцензій на ОС Microsoft Windows на всі робочі місця в університеті. Природничі факультети скористались передплатою MSDNAA (згодом Dream Spark) для отримання ліцензійних Microsoft Windows і Visual Studio, з іншим ПЗ виникали проблеми ліцензування. Схожа ситуація спостерігалась і на гуманітарних факультетах, де до придбаних ліцензійних Microsoft Windows і Microsoft Office додавалося решта ПЗ без особливого піклування про його ліцензійність. У 2013 р. на факультеті електроніки через брак коштів, виділених факультету на придбання засобів обчислювальної техніки, було прийнято рішення про переведення усіх навчальних лабораторій факультету на вільне програмне забезпечення. Згодом за бажанням деяких викладачів, Вчена Рада факультету прийняла рішення про оформлення дворічної передплати Dream Spark на факультет, однак через помилки в апікаційних документах факультет цієї передплати так і не отримав. Через кілька місяців після цього кафедри радіофізики і комп'ютерних технологій вдалося оформити безоплатну передплату Dream Spark на кафедру. Завдяки цьому в чотирьох навчальних лабораторіях факультету встановлено ОС Linux і прикладне програмне забезпечення для неї, в трьох навчальних лабораторіях встановлено ОС Linux та Microsoft Windows 7 за програмою Dream Spark. Прикладне ПЗ, яке встановлюється в навчальних лабораторіях під ОС Microsoft Windows 7 обов'язково має бути ліцензійним. Це призвело до появи в навчальних лабораторіях факультету вільного і безоплатно поширюваного ПЗ. Студенти факультету електроніки також мають змогу встановити ліцензійну ОС Microsoft Windows за програмою Dream Spark. На сьогодні у лабораторіях факультету забезпечено викладання близько 40 навчальних курсів винятково на платформі ОС Linux та 8 курсів на комбінованій платформі - Windows/Wine+Linux.

У 2015 році в Університеті при Інституті післядипломної освіти та доуніверситетської підготовки організовано курси "Інформаційні технології в освіті" у програмі яких основний акцент зроблено на використання вільного програмного забезпечення в навчальному процесі. Метою курсів є підвищення обізнаності академічного персоналу у сфері інформаційних технологій та формування належної культури використання програмного забезпечення.

Підводячи підсумки можна константувати, що:

1. Незважаючи на об'єктивні труднощі, які виникають під час вибору

програмного забезпечення, перехід на відкрите програмне забезпечення в цілому можна вважати вдалим;

2. Широкий спектр наявного вільного програмного забезпечення та відповідних користувацьких інтерфейсів показав можливість повного забезпечення лабораторних робіт з нормативних та вибіркових дисциплін. Це ставить під питання подальшу доцільність витрачання коштів на придбання платних програмних продуктів. У першу чергу актуальним є своєчасне оновлення і модернізація комп'ютерів, які використовуються в навчальному процесі.

3. Лабораторні роботи з деяких курсів виконуються в MS Windows не через доконечну потребу в цій ОС, а лише тому, що лектори, які читають ці лекційні курси, не подбали пошук вільних аналогів використовуваних програм або про запуск потрібних їм програм у системі Wine.

4. Розробка необхідного університету програмного забезпечення може здійснюватися виключно з використанням засобів у відкритому доступі не знижуючи, а може навіть підвищуючи її високу технологічність.

Джерела:

1. Апунович С. Є., Злобін Г. Г., Рикалюк Р. Є., Шувар Р. Використання вільного програмного забезпечення у навчанні і наукових дослідженнях у Львівському національному університеті імені Івана Франка // Матеріали міжнар. наук.-практ. конф. FOSS Lviv-2011. – Львів, 2011, с. 135.
2. Батюк А.Я., Злобін Г.Г. Використання ВПЗ для тестування апаратного забезпечення ПЕОМ в навчальному процесі факультету електроніки ЛНУ імені Івана Франка // Матеріали другої міжнар. наук.-практ. конф. FOSS Lviv-2012. – Львів, 2012. с.23
3. Бойко Я., Ванькевич Д., Злобін Г. Використання технології віртуалізації в навчальному процесі факультету електроніки ЛНУ імені Івана Франка // Матеріали другої міжнар. наук.-практ. конф. FOSS Lviv-2012. – Львів, 2012. с.24
4. Рудий М.Ф., Використання крос-платформного інструментарію розробки програмного забезпечення Qt для створення навчальних програм Матеріали другої міжнар. наук.-практ. конф. FOSS Lviv-2012. – Львів, 2012. с.101
5. Шийка Ю.А., Шувар Р.Я. Виконання завдань розподіленої обробки зображення під управлінням системи CONDOR // Матеріали другої міжнар. наук.-практ. конф. FOSS Lviv-2012. – Львів, 2012. с.114
6. Столярчук О.В., Шувар Р.Я., Продивус А.М. Виконання завдань розподіленої обробки зображення під управлінням системи CONDOR // Матеріали другої міжнар. наук.-практ. конф. FOSS Lviv-2012. – Львів, 2012. с.127

Використання Qt Creator для побудови системи опитування **Кушнір В.**

Львівський національний університет імені Івана Франка, vasylll95@gmail.com

Today, an important component in IT is input data, which we can operate. To do this, we create different profiles to collect data. Some companies automatize this process as quickly as possible, but it is very expensive or not optimal software frameworks and integrated programming environment. So I decided to try the free and optimal integrated development environment Qt Creator.

Qt Creator — інтегроване середовище розробки, призначене для створення кросплатформних застосунків з використанням бібліотеки Qt. Підтримується розробка як класичних програм мовою C++, так і використання мови QML для визначення сценаріїв, в яких використовується JavaScript, а структура і параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt Creator може використовувати GCC або Microsoft VC++ в якості компілятора і GDB як зневаджував. Для ОС Microsoft Windows версій бібліотека комплектується компілятором, заголовковими і об'єктними файлами MinGW.

Особливості MinGW:

Зроблений спеціально для розробки на Qt;

Вбудований редактор форм (Qt Designer) і довідкова система (Qt Assistant);

Контекстно-залежна система допомоги;

Розширюваність плагінами;

Є графічний фронтенд для GDB;

Підтримка зневадження за допомогою CDB;

Для створення проектів використовується qmake (планується підтримка Makefile та тестується підтримка Cmake);

Узагальнене підсвічування синтаксису, підтримується велика кількість мов програмування і розмітки. Є можливість створення своїх стилів підсвічування;

Можливість редагувати етапи складання проекту;

Підтримка розробки мовами C/C++, JavaScript, QML;

QML-дизайнер;

Можливість розробки для ОС Symbian і Maemo зі зневадженням в симуляторі або на пристрої.

На цій IDE я розробив сервер з використанням таких класів як:

- QTcpServer (клас, який забезпечує створення сервера на основі TCP);
- QTcpSocket (клас, який забезпечує створення сокета на основі TCP);
- QThreadPool(клас, який містить колекцію об'єктів потоків).

Після написання асинхронного сервера я здійснив перевірку на кількість можливих з'єднань за секунду (рис.1)



Рис.1 Перевірка сервера на кількість з'єднань за секунду.

Для зберігання даних на сервері використано базу даних Berkley DB.

Berkeley DB (BDB) - вбудована високопродуктивна СУБД, реалізована у вигляді бібліотеки. BDB є нереляційною базою даних - вона зберігає пари « ключ - значення » як масиви байтів і підтримує безліч значень для одного ключа . BDB може обслуговувати тисячі процесів або потоків, які одночасно маніпулюють базами даних розміром в 256 терабайт, на різноманітному обладнанні та під різними операційними системами , включно з більшістю UNIX-подібних систем та [Microsoft](#) Windows, а також в операційних системах реального часу.

Berkeley DB примітна своєю простою архітектурою в порівнянні з іншими системами баз даних, такими як, наприклад [Microsoft SQL Server](#) і [Oracle Database](#) . Наприклад, в ній відсутній мережевий доступ - програми використовують базу даних через виклики внутрішньопроцесного [API](#). Berkeley DB передбачає роботу з парами ключ-значення, де ключ і значення можуть мати фіксовану або змінну довжину, а функція порівняння ключів може бути написана і призначена прикладним програмістом. Програма, яка використовує БД, сама вирішує, як дані зберігаються в запису; БД не накладає обмежень на дані, що зберігаються в записах. Запис і її ключ можуть мати розмір до чотирьох гігабайт.

Джерела:

1. Qt Documentation <http://doc.qt.io/>
2. Berkeley DB https://ru.wikipedia.org/wiki/Berkeley_DB
3. Qt <https://ru.wikipedia.org/wiki/Qt>

Використання вільно поширюваного програмного забезпечення при підготовці обдарованих дітей до олімпіад з комп'ютерної графіки та анімації
Кузьменко А.

аспірантка НПУ імені М.П. Драгоманова, kuzmenko.dtl@gmail.com

The purpose of research is review free software that can be used in the preparation of children to competitions in computer graphics and animation. The object of research is the editor of vector and bitmap graphics, two- and three-dimensional animation. Subject of research is using these software tools in preparing students for competitions. Identification and development of creative abilities in children is one of the urgent problems in teaching practice. Using the free software for processing graphic data develops creativity and skills of independent scientific knowledge, self-identity.

Вступ. Виявлення здібностей (інтелектуальних і творчих) у дітей, їх розвиток є однією з актуальних проблем на сучасному етапі розвитку педагогічної теорії та практики.

Постановка задачі. Проблема розвитку творчого мислення школярів, розробка системи навчання обдарованих дітей була і залишається актуальною. Проведення олімпіад з веб-дизайну, комп'ютерної графіки, комп'ютерної анімації ставить перед вчителем питання вибору програмного засобу для досягнення високих результатів.

Мета роботи. Метою дослідження є аналіз використання вільно поширюваного програмного забезпечення при підготовці обдарованих дітей, зокрема учнів Технічного ліцею м.Києва, до олімпіад з комп'ютерної графіки та анімації.

Основна частина. Є учні, які мають певні здібності в різних галузях використання комп'ютера, володіють навичками краще за інших, мають хист до різного роду діяльності, що пов'язана із сучасними інформаційними технологіями. Основним завданням педагога є створення умов для самовизначення, самовираження учнів через залучення їх до творчої діяльності. Саме через змагання найкраще розкриваються здібності учнів [1].

Олімпіада – це конкурс, у якому переможцями стають найсильніші, а інші учасники здобувають необхідний досвід. Тільки зацікавленість і добровільний принцип допомагають залучати учнів до підготовки до олімпіад [2]. У даний час у місті Києві проводяться олімпіади з інформатики, офісних технологій, веб-дизайну, комп'ютерної графіки, комп'ютерної анімації.

Завданням III (міського) етапу учнівської олімпіади з інформаційних технологій у номінації «Комп'ютерна графіка» 2016 року для учнів необхідно було створити ілюстрацію до казки, макет вітальної листівки, макет поштової марки України тощо. Завданням III (міського) етапу учнівської олімпіади з

інформаційних технологій у номінації «комп'ютерна анімація» 2016 року було зобразити виступ циркових артистів.

Використання вільно поширюваного ПЗ стало доброю традицією для проведення олімпіад з КГ та КА та веб-дизайну до переліку рекомендованого програмного забезпечення входять Inkscape, GIMP, Paint.Net, Synfig Studio, Vectorian Giotto, Blender [3].

Inkscape – ПЗ для створення ілюстрацій у форматі векторної графіки. Передбачені інструменти для зручності малювання: Еліпс, Зірка, Спіраль, Каліграфічне перо, Текст, Градієнт, Піпетка; підтримується робота з контурами; розширені можливості роботи з клонами об'єктів.

Gimp - графічний редактор для створення і редагування зображень. До основних характеристик графічного редактора Gimp можна віднести: повний комплект інструментів, необмежена кількість одночасно відкритих зображень; повна підтримка альфа каналу; інструменти трансформації: обертання, масштабування, віддзеркалення, нахил; повна історія роботи із зображенням; можливість роботи з окремими кадрами.

Paint.NET – растровий графічний редактор, що підтримує роботу з шарами, є простим у використанні, містить достатню кількість ефектів для вдосконалення зображень. Paint.NET містить прості інструменти для малювання фігур. Українську мову можна встановити після інсталяції українзатора.

Synfig Studio – програмний засіб для створення двовимірної векторної анімації. Має інструменти подібні до Gimp. Робота цього програмного засобу базується на технології «ключових кадрів», яка полягає в тому, що створюються не всі кадри, а лише «ключові», між ними «проміжні кадри» створюються автоматично. Цей процес називається твінінгом [5].

Vectorian Giotto – програма для створення Flash анімації. Vectorian Giotto підтримує схеми і градієнтні заповнення, містить розширений редактор кольорової палітри, генератор вбудованих ефектів, є можливість вибору параметрів анімації, додавання звуків, фільтрів.

Blender – багатофункціональний 3D редактор, що створює моделі і анімацію. Є можливість переходу між режимами моделювання та анімації. Програма підтримує кісткову анімацію, нелінійну анімацію, морфінг, інверсну кінематику, різні прив'язки ключових кадрів, скриптову мову Python.

Учні Технічного ліцею активно приймають участь у районних та міських олімпіадах. У 2015-2016 н.р. призерами районного етапу стали 160 учнів, серед них 61 учень став призером з інформатики.

Кількість призерів районних етапів олімпіад учнів Технічного ліцею у 2015-2016 н.р. зросла (для прикладу у 2014-2016 н.р. – 29 учнів, 2013-2014 н.р. – 9 учнів), що є результатом системної роботи по підготовці обдарованих дітей. Щодо використання ПЗ, то при підготовці вчителі кафедри інформатики використовували лише вільно поширюване ПЗ. Також необхідно звернути увагу на той факт, що вперше на районній олімпіаді з КА

було використано Synfig Studio (з 23 учнів Технічного ліцею 6 використали цей ПЗ). Зазвичай анімації створювались за допомогою GIMP, проте покадрова анімація поступається технології ключових кадрів.

На міському етапі олімпіади з КА прийняли участь 143 учасники київських шкіл, що використовували наступне ПЗ (рис.1):

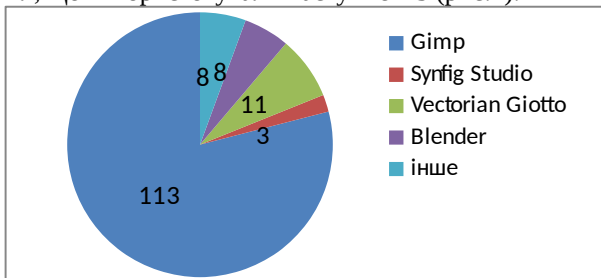


Рис.1. Використання ПЗ на олімпіаді з комп'ютерної анімації у 2015-2016 н.р.

Успіх на олімпіаді чекає того, хто ретельно готувався. Без системної роботи на уроці і після уроків перемога в олімпіаді неможлива. Згідно з новою програмою вивчення інформатики в 5-9 класах на вивчення теми створення та опрацювання графічних зображень виділяється 9 годин в п'ятому класі і 6 годин в дев'ятому, щодо анімації, то на роботу в цьому перспективному напрямі години на вивчення не відводяться.

Для обдарованих учнів у столичних навчальних закладах ведеться планомірна позаурочна діяльність, спрямована на задоволення індивідуальних освітніх інтересів, потреб і схильностей учнів. При підготовці до олімпіад вчителями кафедри інформатики Технічного ліцею було розпочато роботу над курсом по основах роботи з вільно поширюваним графічним ПЗ за допомогою LCMS MOODLE.

Висновки. Редактори векторної графіки **Inkscape**, растрової графіки **Gimp** та **Paint.NET**, двовимірної анімації **Synfig Studio** та тривимірної анімації **Blender** є вільно поширювальним програмним забезпеченням, яке використовується при підготовці обдарованих дітей, зокрема учнів Технічного ліцею м.Києва, до олімпіад з комп'ютерної графіки та анімації, що сприяє розвитку творчих здібностей і навичок самостійного наукового пізнання, самореалізації особистості.

Джерела:

1. Войцеховський М.О. Робота з обдарованою молоддю міста Києва //Комп'ютер у школі та сім'ї. – 2010. – № 4. – С. 27.
2. Глухова М.В., Герман А.Ю. Досвід роботи з обдарованими учнями в галузі ІКТ //Комп'ютер у школі та сім'ї. – 2009. – № 4. – С. 14.
3. Київські учнівські олімпіади з інформаційних технологій [Електронний ресурс] – Режим доступу: <http://www.kievoit.ippo.kubg.edu.ua>

Noise signal simulation with Octave

Lapich S.V., Derechennik S.S., Menshih T.J., Divinets A.A.

Brest State Technical University

Simulated model was designed to find Gaussian and Rayleigh noise probabilistic characteristics. The project is based on GNU Octave. The main fields of application for this project are the microelectronics and radio-engineering.

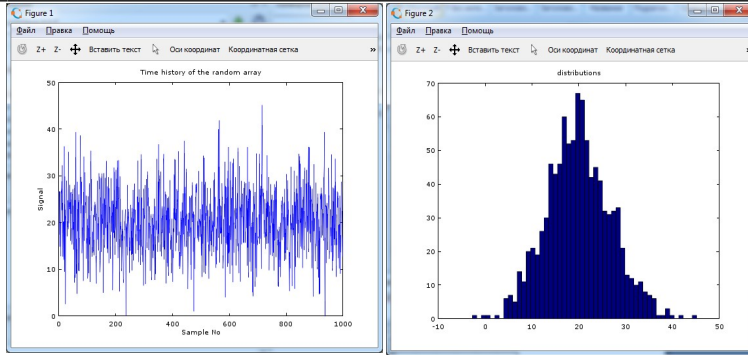
The noise is a random signal that exists at all frequencies. Currently noise signal used as a carrier instead of harmonic one are of the utmost interest for the information transfer. Specifically, the noise is most promising as a candidate for data carrier in terms of data protection on the first layer of OSI model [1].

Due to the rise of popularity noise signals simulation is becoming even more urgent, for example at development of multichannel noise modules with a solid-state diode generator used as a random signal source.

At present, mathematical apparatus for this simulation is developed rather in details. Most common mathematical models of random signals and noise are the telegraph signal, "white" noise, Gaussian noise, and Rayleigh noise. In real-life tasks of the radio technology, telecommunications and in other fields (including computer science) the model of Gaussian noise is typically used, which is created by summation of statistically independent "white" noises [2]. The "white" noise in its turn is a stationary random process $q(t)$ with autocorrelation function described by the Dirac delta function, and with frequency-independent spectral power density which has a constant value $Gq(w)=\sigma^2$ equal to $q(t)$ values dispersion.

"White" noise model fits into the AR model [3] which can be described by a following difference equation: $F_t = \sum_{i=0}^{i=\infty} (a \cdot x_{t-i}) + \sigma$, which involves AR model parameter a , a random value x_{t-i} , and dispersion σ .

In noise signals computer simulation cases the developer bumps into a considerable variety of software packages which are satisfactory for this purpose. However, search does not show specialized software for noise signals simulation, and so universal mathematical packages become an option. In theory, large data packets with functionality for all occasions should contain tools to simplify and automate this simulation. However, the experience gained by authors while dealing with Matlab have shown that its built-in generators allow to create standard model, while any deviation from standard parameters requires implementing noise source from scratch. Obviously the rejection of commercial mathematical package in favor of free software is a significant advantage in such case. On the one hand, it addresses issues of affordability of the software, and on the other hand, open source improves verifiability of the model (for example, when it is necessary to substantiate some unexpected results).



*Figure 1 – Gaussian random process of "white" noise and density histogram
(Number of samples - 1000, Give the average - 20, Give the variance - 50)*

These considerations were the main argument for choosing the programming environment for analysis of the experimental sample parameters of noise modules manufactured in Belarus, based on crystals of the ND201L silicon diode noise generators [2]. As a result the Gaussian noise model was implemented with help of the open source GNU Octave package [4].

Octave provides an interactive command line interface for solving linear and nonlinear mathematical problems and numerical experiments. In the simulation of Gaussian noise `randn()` function generating an array of random variables was used, as well as standard tools for noise visualization (`plot`) and histogram distribution density (`hist`). Gaussian random process of "white" noise and density histogram simulated with Octave are shown in Figure 1. Figure 2 shows the similarly simulated Rayleigh noise (built on the same apparatus but having differently defined mathematical function) and the corresponding histogram of the distribution density.

As an advantage of the used software we can note the high compactness of the code (typical implementations of our the models can be found at <https://github.com/svlapich/signal>) as well as a simple parallelization of computational load in Octave which is relevant to noise signal generation problems.

References:

1. Корчинский В.В. Модель шумового сигнала для передачи конфиденциальной информации / В.В Корчинский // Вестник НТУ "ХПИ", 2013. – №11. – С. 89–94.
2. Разумейчик В.С. Оценка вероятностных характеристик случайных сигналов микроэлектронного шумового модуля / В.С. Разумейчик, В.В. Буслюк, С.С. Дереченник, В.И. Поляков, С.В. Лапич // Вестник Брестского государственного технического университета, 2014. – № 5 : Физика, математика, информатика. – С. 4145.
3. Громаков Ю.А. Структура TDMA кадров и формирование сигналов в стандарте

GSM / Ю.А. Громаков // Электросвязь, 1993. – №10. – С. 9–12.

4. Алексеев Е.Р., Чеснокова О.В. GNU Octave для студентов и преподавателей. – Донецк.: ДонНТУ, Технопарк ДонТНУ УНИТЕХ, 2011. – 332с.

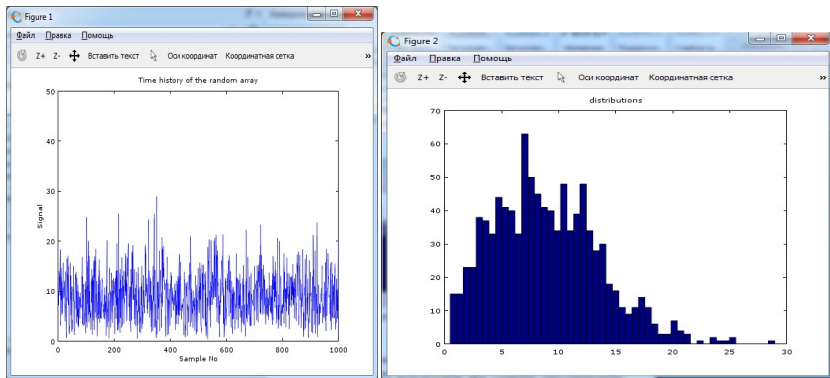


Figure 2 – Rayleigh noise and density histogram (Number of samples - 1000, Give the variance - 50)

Використання rust для реалізації розширень інтерпретованих мов І. Лаврів

Lohika.Ltd, lavriv92@gmail.com

Rust is a general-purpose, [multi-paradigm](#), [compiled programming language](#) sponsored by [Mozilla](#) Research. It is designed to be a "safe, [concurrent](#), practical language", supporting [pure-functional](#), [imperative-procedural](#), and [object-oriented](#) styles.

Rust - це системна мова програмування, яка розробляється та підтримується організацією Mozilla Research. Розробку мови почав у 2006 р. Грейдон Гоар, а в 2009 р. до розробки приєдналась Mozilla. Тривалий час Rust розроблявся, як експериментальна мова програмування і тільки в травні 2014 р. вийшла перша стабільна версія. Наразі стабільною є версія 1.8.0. Серед особливостей Rust можна виділити компіляцію з допомогою низькорівневої віртуальної машини, сфокусованість на безпеці та коректній роботі з пам'яттю, поєднання різних парадигм програмування.

Традиційна програма "Hello world" мовою Rust виглядає доволі просто і зрозуміло:

```
// hello_world.rsfn main() { println!("Hello world!!!");}
```

У цьому прикладі за допомогою ключового слова `fn` оголошується вхідна функція, яка повинна мати ім'я `main`, і в тілі цієї функції за допомогою вбудованого макросу `println!` на екран виводиться повідомлення "Hello world".

Важливим елементом мови програмування є оголошення змінних, особливістю змінних в Rust є те, що вони за замовчуванням не змінюються та

є статично типізованими.

```
let a = 1; // за замовчуванням змінна незмінювана
let mut a = 1; // Ключове слово mut вказує, що значення надалі
    повинне змінитись.
let a: i32 = 2; // змінні можуть отримувати типи, як автоматично, так
    і явно
```

Ще однією цікавою особливістю Rust є алгебраїчні типи даних. Літера в назві типу вказує тип даних, а число їх довжину в діапазоні від 8 до 64 біт.

Наприклад:

- i32 - ціле число довжиною 32 біти
- i64 - ціле число довжиною 64 біти
- f32 - число з плаваючою комою 32 біти

Мова дає змогу створювати структури даних за допомогою ключового слова `struct`

```
struct Point { x: i32, y: i32}let point = Point {x:1, y:1};println!(
    "{}, {}", point.x, point.y); // виведе (1, 1)
```

Також можна описувати методи для обробки даних структури за допомогою ключового слова `impl`

```
impl Point { fn format(&self) -> &str { format!("{x.3}, {y.3}"),
    self.x, self.y);}}
```

В Rust вбудована підтримка функціональної парадигми програмування.

- Оголошення функції.

```
fn sum(a: i32, b: i32) -> i32 { a + b}
```

- Підтримка лямбда-числення та функцій, як об'єктів першого класу.

```
let sum = |a: i32, b: i32| a + b;
```

- Підтримка функцій вищих порядків (Функція може бути як аргументом, так і результатом іншої функції.)

```
fn test(n: i32) -> fn(i32) -> i32 { let f = |a, b| a + b;
    return f;} let plus_one = |x| x + 1;test(plus_one);
```

Щодо абстракцій, то Rust дозволяє створювати трейти для даних та генерики для відділення опису та реалізації типів даних і функцій

```
struct Point<T> {x: T, y: T}let point = Point {x:1,y:1};let point1 =
    Point{x:1.0, y: 1.0};
```

До репозиторію Rust також входить потужна утиліта для управління кодом проекту, вона дає змогу компілювати вихідний текст, запускати тести, збирати документацію, керувати сторонніми залежностями тощо.

```
cargo new project_name --bin # створює структуру для нового проекту
cargo build # компілює код проекту cargo run # компіляція і запуск вихідного
файла
```

Найпоширенішою нішею, де Rust знайшов вже своє застосування є написання системних утиліт та розширень для проектів, написаних іншими мовами програмування (python, ruby, javascript), та реалізація функціоналу недоступного цим мовам (швидкодія, багатопоточність, низькорівневе управління пам'яттю). Для інтеграції з цими мовами можна використовувати наступні способи:

- Реалізація мікросервісної архітектури з використанням черг

повідомлень;

- Виклик застосунків на Rust, як консольних утиліт;
- Взаємодія з допомогою FFI (Foreign functional interface).

Ось приклад написання модуля за допомогою FFI;

- Опис типу скомпільованого файлу та залежностей проекту на Rust

```
# Cargo.toml[package]name = "test_addon"version = "0.0.1"authors =
["Ivan Lavriv <lavriv92gmail.com>"][lib]name =
"simple_addon"crate_type = ["dylib"][dependencies]# List of your
dependencies mongodb = "*"

• Реалізація модуля мовою Rust
// lib.rs #[no_mangle]fn public_function() {    // do something}

• Реалізація модуля обгортки (в цьому випадку мовою python).
# module.py import ctypesmodule =
ctypes.cdll.LoadLibrary('./path_to_compiled_addon')module.public_
method()
```

У разі реалізації розширення за допомогою FFI існує типова проблема синхронізації типів даних та семантичних особливостей між двома мовами. Для цього існують вже готові бібліотеки, які реалізують особливості інших мов з допомогою особливостей Rust (rust-cpython, rust-rmi, тощо). Ось приклад використання rust-cpython

```
use std::any::Any;extern crate cpython;use
cpython::ObjectProtocol;use cpython::{PyList, PyDict, PyObject,
Python};#[no_mangle]fn get_data_as_dict() -> PyDict
{}#[no_mangle]fn get_data_as_list(iterable: &[f64]) -> PyList
{let python = Python(); let result = iterable.map(|&elem: &T|
{ match elem { Some(elem) => process(elem), None =>
Python.None()}); return PyList(python, result);}
```

Як висновок можна сказати, що Rust є сучасною, швидкою та безпечною мовою програмування, яка дає змогу програмісту цікаво і якісно вирішувати системні задачі в програмуванні.

Використання вільного програмного та апаратного забезпечення для точної реєстрації часових міток подій.

Мартинюк-Лотоцький К.П., Сергеев О.В.

1. Міжнародний центр Астрономічних і медико-екологічних досліджень, національна академія наук України;
2. Львівський національний університет імені Івана Франка, Астрономічна обсерваторія, langurek@gmail.com

Precise timing is one of the keystones for modern astronomical observations. The local timing service of the telescope Zeiss-600 at International Center for Astronomical, Medical and Ecological Research was upgraded. Its hardware consists of the GPS receiver and micro-PC RaspberryPi with RS422-to-TTL convertor. The corresponding software (Raspbian OS based) provides the necessary operations: timing data storage on SD card with access via ssh or ftp, providing local ntp server (it is synchronized over the public Internet and/or by PPS from GPS receiver). The developed ntp server maintains time to within 0.004 ms jitter

(accuracy), that is 100 times better than without GPS.

Для забезпечення точності і достовірності астрономічних спостережень необхідно забезпечити реєстрацію часу з високою точністю і стабільністю. Раніше використовували службу часу, на основі прийому сигналів точного часу по радіо. Її недоліки: спонтанне завмирання радіосигналу, під час зміни погоди сильні завади, тощо. З появою Інтернету почали використовувати ntp-сервери часу. Перевага: відразу можна отримати час з похибкою менше 20 мс, що є для більшості задач достатньо. Недолік: якщо немає зв'язку з Інтернетом, то час теж не можна синхронізувати.

На даний час існує достатньо багато високоточних систем реєстрації моментів подій на основі GPS-технологій. В цій роботі обговорюється використання мікрокомп'ютера RaspberryPi для узгодження інтерфейсів і організації роботи GPS і служби часу в цілому, включно з записом моментів подій, перетворенням форматів файлів і дистанційним доступом до них. Причому, служба часу повинна забезпечувати неперервний режим реєстрації моментів часу в складних умовах, коли сигнал від супутників системи GNSS (GPS) може тимчасово пропадати.

Для забезпечення вищої точності і надійності було встановлено GPS-приймач з входом маркера подій, який дає похибку реєстрації однієї події менше 100 нс. GPS має вихідний PPS-сигнал, по інтерфейсу RS-422 у форматі TSIP можна отримати дані про дату, час, точність реєстрації координат/часу, кількість супутників та інше.

GPS під'єднано до мікрокомп'ютера RaspberryPi через перехідник RS-422 -> RS232(TTL), сигнал PPS заведено на один з входів GPIO. На системі Raspbian (Debian Linux) запущено локальний ntp-сервер який синхронізується по мережі з ntp-серверами інтернету, а також уточнює час по PPS сигналу з GPS. Для реєстрації моментів відкриття/закриття затвору світлоприймального пристрою (ПЗЗ-камери, ФЕП тощо) зроблено перехідник (з оптич-роз'язкою) з вхідного струмового сигналу на сигнал event GPS (RS-422). Дані з GPS у форматі Trimble TSIP розшифровуються Perl-скриптом і далі зберігаються локально у файли і sql-бази.

Інформація про статус служби часу, про файли з моментами часу спрацьовування затвору відображаються на web-сторінці сервера, який запущено на легкій платформі lighttpd. Також дані можна завантажити по rcp(ssh), ftp, які періодично синхронізуються з резервним сервером по rsync, і до якого є доступ по samba для win-машин.

Система створена і введена в роботу на телескопі Zeiss-600 (пік Терскол, МЦ АМЕД) для реєстрації фактичних моментів відкриття/закриття затвору астрономічного світлоприймального пристрою (CCD SBIG, Apogee).

Порівняльні характеристики роботи локального ntp-сервера приведено у таблиці. Значення зміщення (offset) і тремтіння (jitter) міток зовнішнього ntp-сервера (супутниковий інтернет) на декілька порядків більші ніж для міток на основі PPS від GPS.

Тип мітки часу	delay, ms	offset, ms	jitter, ms
ntp	0.617	21.203	0.486
PPS	0.000	-0.005	0.004

Джерела:

1. <http://www.raspberrypi.org>
2. <http://www.satsignal.eu/ntp/Raspberry-Pi-quickstart.html>

Месенджер для платформи Android на основі протоколу WebRTC

Муха Б.М., ШІнак З.Я.

Національний університет «Львівська політехніка», muhabohdan@hotmail.com

The new open source protocol WebRTC for browser-to-browser real-time communication is analyzed. WebRTC-based application named Q-municate for Android platform is presented. Q-municate supports voice calling, video chats and group conferences, peer-to-peer data and file sharing without the need of either internal or external plugins. Designed messenger enforces usage of the encryption for both the media and the signalling.

Кожного дня люди передають один одному великі потоки інформації, спілкуються між собою за допомогою телефонів, чатів, відеоконференцій, обмінюються даними, фотографіями, файлами тощо. Передача відео і аудіо-інформації у режимі реального часу стала важливим етапом у розвитку нових Веб-технологій. Масового поширення набули засоби передачі медіа-інформації, що дають змогу здійснювати аудіо та відео-дзвінки, а також надсилати текстові повідомлення і медіаконтент. Актуальною і важливою є задача організації захищеного з'єднання між двома й більше учасниками розмови та безпосереднього передавання інформації до адресата без використання проміжних ланок.

Класичні протоколи передачі даних, такі як SIP та RTP, не задовольняють цих вимог, оскільки вони використовують проміжні сервери для забезпечення постійного з'єднання між клієнтами, що призводить до зниження захищеності. Протокол режиму реального часу RTP використовує динамічно призначувані адреси портів, чим створює труднощі у процесі проходження міжмережевих екранів. Для обходу цієї проблеми зазвичай використовують STUN-сервер. Це призводить до різкого зниження швидкості передачі даних та істотного спотворення відеозображення. Також недоліком є те, що STUN- і SIP-сервери, які необхідні для встановлення і підтримки з'єднання між клієнтами, не є безпечними ланками. Специфікація

цих протоколів не змінювалася тривалий час [1], не вносились зміни в систему їхньої безпеки, що збільшує загрозу стороннього проникнення у сеанс зв'язку і викрадення конфіденційної інформації.

Вказаних недоліків можна позбутися, якщо для організації передавання поточкових даних у месенджері використати кросбраузерний та кросплатформний протокол з відкритим кодом WebRTC (Web Real-Time Communication). Цей протокол підтримує створення застосунків для реалізації міжбраузерних голосових дзвінків та відеочатів, а також обміну медіафайлами без необхідності під'єднання додаткових зовнішніх чи внутрішніх плагінів [2]. Безпека і шифрування не є додатковими функціями протоколу WebRTC, а забезпечуються вбудованими компонентами за замовчуванням. Крім того, WebRTC пропонує наскрізне шифрування між вузлами практично на будь-якому пристрої, створюючи безпечний зв'язок в режимі реального часу.

Для передачі даних WebRTC використовує архітектуру «точка- точка» та протокол датаграм безпеки транспортного рівня DTLS (Datagram Transport Layer Security). Цей протокол за замовчуванням вбудований у всі Веб-переглядачі, що підтримують технологію WebRTC (Chrome, Firefox, Opera), він набуває широкої популярності серед розробників мобільних додатків. У з'єднанні, зашифрованому за допомогою DTLS, унеможливується підслуховування чи підміна інформації.

Крім DTLS, технологія WebRTC використовує для шифрування аудіо та відеоданих безпечний протокол передачі даних SRTP (Secure Real-Time Protocol). Цей протокол не допускає прослуховування або перегляду IP-зв'язку (голосового і відео трафіку) недозволенними сторонами.

Двигун обробки аудіо протоколу WebRTC – це компонент, який забезпечує передачу аудіосигналу від аудіокарти до мережевого інтерфейсу і в зворотному напрямі. Він містить аудіокодеки iSAC і Opus, системи послаблення луни і зменшення шумів. Кодек iSAC є широкосмуговим аудіокодеком для VoIP і потокового аудіо, використовує частоту дискретизації 16/32 кГц з адаптацією швидкості передачі даних від 12 до 52 Кбіт/с. Opus підтримує постійну і змінну швидкість передачі даних від 6 до 510 Кбіт/с і частоти дискретизації від 8 до 48 кГц.

Двигун обробки відео є компонентом для управління відеосигналом, що містить кодек VP8 та систему адаптивного відеобуфера Jitter Buffer. Кодек автоматично підлаштовується до поточного стану з'єднання і системи корекції зображення. VP8 має підвищену стійкість до втрати пакетів, механізм фільтрації артефактів, який може застосовуватися диференційовано до різних частин кадру залежно від їхніх особливостей, можливість масштабувати декодоване зображення, а також профілі, оптимізовані для проведення відео-конференцій в реальному часі. Кодеки Opus і VP8 добре підходять для глобального Інтернету, де швидкість передачі даних під час передачі може падати до дуже низьких значень через низьку якість зв'язку,

тим самим забезпечуючи широкую сферу використання протоколу WebRTC [3].

З урахуванням описаних вище властивостей стеку протоколів WebRTC розроблено месенджер Q-municate як застосунок для платформи Android, який дає змогу здійснювати аудіо та відеодзвінки між двома й більше користувачами в режимі реального часу, передавати файли, зображення, спілкуватись у приватних та групових чатах, організовувати відео конференції. Додатковими корисними функціями програми є можливість вибору співбесідника з телефонної книги, авторизація через обліковий запис соцмережі Facebook (рис.1), модерація груп для спілкування та інші.

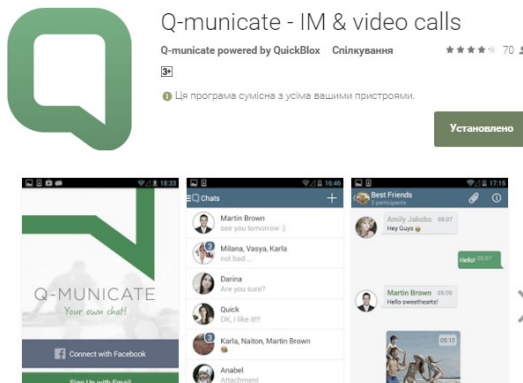


Рис.1 Програма Q-municate у магазині додатків Google Play.

Основною перевагою та відмінністю від аналогів застосунку Q-municate, орієнтованого на протокол WebRTC, є те, що між усіма учасниками спілкування створюється локальне з'єднання – підмережа, в якій відбувається захищена шифруванням передача повідомлень різного роду, що дає змогу безпечно розмовляти та обмінюватися довільним медіа контентом, не турбуючись про приватність даних.

Двигун програми орієнтований на роботу з API функціями BaaS-сервісу QuickBlock, що має розроблене серверне рішення для створення аудіо й відеоконференцій та забезпечує можливість під'єднання до 40 співбесідників в одному потоці.

Таким чином, застосунок Q-municate відповідає основним вимогам до сучасних засобів спілкування. Застосування протоколу WebRTC, забезпечує шифрування всіх даних у режимі реального часу і встановлює з'єднання за технологією «точка-точка», гарантуючи користувачам обмін аудіо та відеопотоками без хвилювання за конфіденційність та цілісність переданої інформації, що є надзвичайно важливим для сьогодення. Розроблений месенджер отримав схвальні відгуки користувачів сервісу Google Play.

Джерела

- 1) Гольдштейн, Б. С. Протокол SIP. Справочник. – СПб.: BHV-Санкт-Петербург, 2005. – 456 с.
- 2) WebRTC 1.0: Real-time Communication Between Browsers. W3C Working Draft 28 January 2016. [Електронний ресурс] – Режим доступу: <https://www.w3.org/TR/webrtc/#peer-to-peer-connections>
- 3) [How WebRTC Is Revolutionizing Telephony](http://blogs.trilogy-lte.com/post/77427158750/how-webrtc-is-revolutionizing-telephony). [Електронний ресурс] – Режим доступу: <http://blogs.trilogy-lte.com/post/77427158750/how-webrtc-is-revolutionizing-telephony>

WxPython – вільна бібліотека для створення повнофункціональних кросплатформних графічних інтерфейсів мовою Python *Миронюк Д.*

Львівський національний університет імені Івана Франка, myronyukdmytro@gmail.com

In this article describes free tool for build full functional cross-platform GUI – wxPython as a part of wxWidgets library. Showing some useful features of this library and some visual constructors for wxWidgets(wxPython).

Відібрано бібліотеку для мови програмування Python, яке дає змогу створювати повноцінні кросплатформні графічні інтерфейси на основі API операційної системи. Критерії, які були проаналізовані під час вибору засобів:

- ✓ Кросплатформність — засіб має бути портованим на різні платформи, оскільки сама мова Python дає змогу створювати сценарії у різних ОС;
- ✓ Повнота бібліотеки — бібліотека мусить мати повний набір компонент для створення повнофункціональних програм;
- ✓ Наявність графічних засобів створення графічних форм — бібліотека повинна мати графічну оболонку, де можна конструювати необхідні форми з повним заданням параметрів компонент. Це дасть змогу зекономити час розробки інтерфейсу;
- ✓ Просте введення готового інтерфейсу в програму - засіб мусить мати прості механізми під'єднання до програми з використанням мінімальної кількості додаткових бібліотек;
- ✓ Можливість повторного використання як шаблону для інших програм такого самого типу;

Як рішення було обрано бібліотеку wxPython, яка є складовою частиною wxWidgets. Бібліотека wxWidgets (раніше відома як wxWindows) — це багатоплатформна бібліотека віджетів. wxWidget дає змогу коду для побудови графічного користувацького інтерфейсу компілюватись і працювати на різних апаратних і програмних платформах з мінімальними змінами, або взагалі без

них. Надаються прив'язки (біндинги) для популярних мов програмування, включно з PHP, [Python](#), Perl і [Ruby](#). На відміну від інших інструментів wxWidgets забезпечує для програми по-справжньому рідний для цільової системи зовнішній вигляд і методи взаємодії, завдяки використанню системних API, а не імітації GUI.

Підтримує такі операційні системи, як Microsoft Windows, Mac OS, Linux/Unix (X11, Motif та GTK+), OpenVMS, OS/2, AmigaOS та мобільних платформ. Написана мовою C++. Проект бібліотеки ще розвивається, і в кожній новій версії вона поповнюється новими компонентами. Має досить велику документацію з описом кожного класу бібліотеки.

wxPython – це обгортка до бібліотеки кросплатформного графічного інтерфейсу користувача, основою якого є wxWidgets. Має великий набір компонент, що дає змогу використовувати його для розробки багато віконних графічних інтерфейсів на основі API системи через wxWidgets. Велика бібліотека компонентів дає змогу розробляти різноманітні повнофункціональні вікна з підтримкою механізму батьківського та дочірнього вікна та відмежування діалогових вікон.

Особливості бібліотеки

На відміну від TkInter, виконану на основі Tcl/Tk з використанням обмеженого числа віджетів, wxPython має повний набір компонент для конструювання інтерфейсів різного рівня складності. Інтерфейси цієї бібліотеки конструюються за допомогою класової моделі, що дає змогу швидко вносити зміни до інтерфейсу, а також підвищує ступінь повторного використання таких форм. Така побудова також дає змогу швидко та просто ввести готову форму в програму (достатньо під'єднати модуль до програми та створити об'єкт класу інтерфейсу).

Поряд з ручним режимом написання нових вікон існує можливість пришвидшити розробку шляхом застосування графічних конструкторів форм, які також є портованими для різних платформ. Найпопулярніші візуальні редактори форм бібліотеки wxWidgets(wxPython):

-wxFormBuilder – одна із найпопулярніших графічних оболонок для конструювання графічних інтерфейсів типу wxWidgets з можливістю генерування готового коду мовами php,Python,Lua, C++ та XRC. Має простий та зрозумілий інтерфейс з поділом компонент по категоріях. Також є можливість переходити до вихідного коду інтерфейсу та редагувати його. Останні зміни (SourceForge – 24.08.2015)

- wxGlade – графічна оболонка, написана мовою Python для конструювання графічних інтерфейсів типу WxPython/wxWidgets з можливістю генерування готового коду мовами Python, C++, Perl, Lisp та XRC. До переваг цього засобу можна віднести простоту інтерфейсу, який представлений трьома окремими вікнами з панеллю інструментів, вікном властивостей та нашою формою, можливість імпорту шаблону, написаного мовою XRC, а також можливість генерувати код різними мовами. Недоліком

оболонки є відсутність редактора коду інтерфейсу. Незважаючи на те, що останній випуск цього інструменту був ще у 2007 році, вона користується популярністю у програмістів.

- VisualWx – графічна оболонка для конструювання графічних інтерфейсів типу wxPython/wxWidgets, яка використовує інтерпретатори Python до версії 2.5. Має простий інтерфейс, схожий з wsFormBuilder. Наразі використовується невеликим колом розробників.

Отже, бібліотека wxWidgets є одною з найповніших бібліотек для конструювання кросплатформних графічних інтерфейсів на основі API операційної системи різними мовами, що сильно розширює сферу її використання. Її оболонка wxPython, особливо у поєднанні з графічним конструктором форм дає змогу швидко розробляти програми з графічним інтерфейсом різного ступеня складності.

Література та посилання:

1. Noel Rappin wxPython in Action / Noel Rappin, Robin Dunn; - Manning Publication, 2006. – 552 p.
2. Cody Precord wxPython 2.8 Application Development Cookbook / Cody Precord; - Packt Publishing, 2010. – 296 p.
3. wxWidgets Compared to Other Toolkits.
[https://wiki.wxwidgets.org/WxWidgets Compared To Other Toolkits#Why You Shouldn't Use wxWidgets](https://wiki.wxwidgets.org/WxWidgets_Compared_To_Other_Toolkits#Why_You_Shouldn't_Use_wxWidgets)
4. Сказ о wx.Python <https://habrahabr.ru/post/139560/>
wxPython-Phoenix Reference Documentation
<http://wxpython.org/Phoenix/docs/html/main.html>
5. wxFormBuilder. <https://sourceforge.net/projects/wxformbuilder/>
6. wxPython <http://www.wxpython.org/>

Досвід проектування складених мереж у академічній хмарі, розгорнутій на основі Apache CLOUDSTACK

Олексюк В.П.

*Тернопільський національний педагогічний університет імені Володимира Гнатюка,
oleksyuk@fizmat.tnpu.edu.ua*

This paper describes deploying of academic cloud. Apache CloudStack has been chosen as platform for cloud. Deployed cloud infrastructure contains such elements: zone, pod, cluster, 3 hosts, 3 primary storages and 1 secondary storage. This cloud provides: run virtual machines (VMs), connect VMs through advanced networks, access to VMs through web-interface and standard protocols, distribution of computing resources for VMs, create template and snapshot of VMs.

Протягом останніх років у галузі інформаційних технологій спостерігається тенденція до надання віддаленого доступу до обчислювальних ресурсів. Популярною моделлю використання обчислювальних ресурсів за вимогою (on-demand computing) є хмарні технології. Ця концепція змінює існуючі уявлення щодо організації доступу

та інтеграції додатків, тому виникає можливість управління більш великими інформаційними інфраструктурами, у яких можна створювати і використовувати як індивідуальні, так і колективні «хмари» [1]. Застосування хмарних технологій у ВНЗ можливе відповідно до моделі «академічної хмари». Серед основних вимог до академічної хмари виділяють:

- забезпечення супроводу кожної навчальної дисципліни повним спектром електронних навчальних ресурсів і послуг;
- забезпечення доступу до програмних продуктів, які використовуються у навчальному процесі;
- надання обчислювальних ресурсів на вимогу користувача та гнучкого їх розподілу;
- підтримку єдиного входу користувачів, вимірювання кількості наданих послуг у часових і ресурсних показниках, гнучкого розподілу ресурсів [2].

Універсальною моделлю розгортання академічних хмар є комбінована (гібридна), перспективним напрямом впровадження хмарних технологій у вищих навчальних закладах вважаємо розгортання корпоративних хмар.

У межах роботи спільної науково-дослідної лабораторії з питань застосування хмарних технологій в освіті ТНПУ імені Володимира Гнатюка й Інституту інформаційних технологій і засобів навчання НАПН України було спроектовано та розгорнуто корпоративну хмару, доступну через віртуальну приватну мережу (адреса <http://cloud.fizmat.tnpu.edu.ua>). На основі порівняльного аналізу джерел [3], [4] програмною основою корпоративної хмари було обрано відкриту, вільно поширювану платформу Apache CloudStack.

Для забезпечення функціонування корпоративної хмари нами використано 3 комп'ютери на основі процесорів Intel Core i5 із загальним обсягом оперативної пам'яті 76 Gb.

Як відомо, основними компонентами хмарної інфраструктури Cloudstack є [5]:

- зона (zone) – найбільший підрозділ, який відповідає датацентру;
- стійка (pod) – є аналогом серверної стійки, яка містить кластери та хости, що належать одній підмережі;
- кластер (cluster) – сукупність фізичних серверів, розміщених у одній стійці;
- хост (host) – сервер, на якому виконується гіпервізор – програма що забезпечує виконання віртуальних машин та розподіл обчислювальних ресурсів для них;
- первинні та вторинні сховища (primary and secondary storages) – зберігають розділи і диски віртуальних машин, які можуть бути доступними за різними протоколами.

Враховуючи що основним завданням проекрованої академічної хмари є забезпечення навчання студентів, ми створили лише одну зону у якій

розгорнули такі компоненти (рис. 1). З метою заощадження обчислювальних ресурсів ми поєднали у кожному фізичному комп'ютері функції гіпервізора та первинного сховища. У процесі створення зони можна обрати базовий або розширений режим мережі. У першому випадку всі комп'ютери зони належатимуть одній мережі. Розширений режим дає можливість організовувати складені мережні структури на основі загально прийнятих технологій VLAN, VPN та інших. Перед нами було поставлено завдання спроектувати хмарну інфраструктуру, у якій можна було б створювати довільну кількість підмереж, кожна з яких можна було б асоціювати з певною фізичною мережею гіпервізора. Причому додавання згаданих мереж не має вимагати зміни топології фізичних мереж. Отож, нами було створено зону, яка функціонує у розширеному режимі.



Рис. 1. Компоненти розгорнутої хмарної інфраструктури

Особливістю платформи Apache CloudStack є те, що вона дає змогу об'єднувати в одній інфраструктурі фізичні та віртуальні мережі, у яких передаються різні види трафіку: управляючий (між сервером управління та вузлами в кластерах), публічний (трафік, який передають віртуальні машини у процесі доступу до Інтернету) гостьовий (генерується між віртуальними комп'ютерами), а також трафік між сховищами. У нашому випадку кожен з вузлів, на яких виконується гіпервізор, приєднаний до двох фізичних мереж. Зону спроектовано так, що в одній з них передаються усі види трафіку (управляючий, гостьовий, публічний, трафік сховищ), а в іншій – лише гостьовий. Крім цього у кожній мережі гостьовий трафік розподіляється на 6 підмереж (по одній мережі для кожної академічної групи). Для маркування трафіку в цих мережах використовується технологія віртуальних локальних мереж (VLAN). У зв'язку з цим нами, на основі ОС FreeBSD, було було сконфігуровано маршрутизатор з підтримкою технології VLAN. Отож, у кожному із віртуальних машин можна додавати мережні адаптери, які працюватимуть в різних підмережах. Загалом схема розгорнутої хмарної інфраструктури має такий вигляд (рис. 2).

Для встановлення відповідності між фізичними адаптерами і трафіком у системі CloudStack передбачено їх маркування. Завдяки цьому 6 фізичних адаптерів, встановлених у вузлах cloud1, cloud2, cloud3 у системі CloudStack відображаються як два. Трафік цих адаптерів передається через комутатори

sw1 та sw2 до маршрутизатора gw. Оскільки основним завданням було розгортання окремих гостьових підмереж, то відповідний трафік також маркується окремими тегами. Для кожного з зазначених тегів створено шаблони мережного обслуговування (network offering), які дають можливість вказати сервіси, які будуть функціонувати у відповідній мережі. Такими сервісами є: сервер динамічного призначення IP-адрес, перетворювач мережних адрес та портів (NAT), міжмережевий екран, балансувач трафіку та інші. Їх функціонування у кожній гостьовій мережі забезпечуватиме системна віртуальна машина – віртуальний маршрутизатор.

З метою підвищення безпеки ми відмовилися від публікування сервісу cloud.fizmat.tnpu.edu.ua в мережі Інтернет. Проте завдяки віртуальній приватній мережі студенти мають повсюдний доступ до корпоративної хмари. Для цього на маршрутизаторі gw налаштовано VPN-сервер.

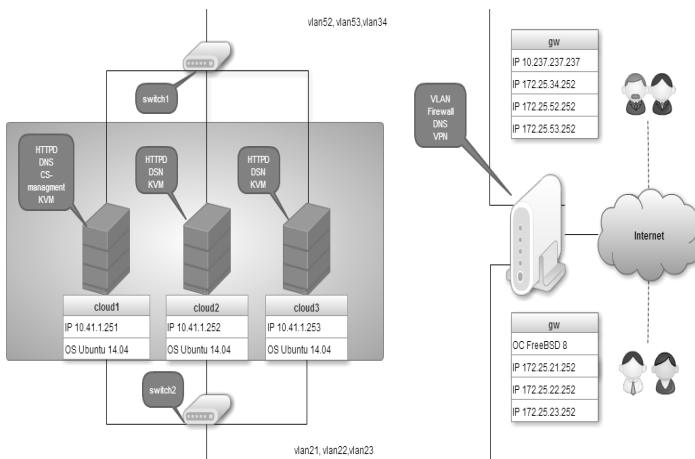


Рис. 2. Схема розгорнутої хмарної інфраструктури

Поряд з перевагами виділимо недоліки застосування платформи Apache CloudStack у навчальному процесі.

- нерациональний розподіл обчислювальних ресурсів, що передбачає їх резервування з розрахунку кількості та продуктивності віртуальних комп'ютерів;
- проблеми одночасного розгортання значної кількості віртуальних комп'ютерів;
- необхідність виконання певних дій шляхом редагування бази даних або використання API-функцій.

Незважаючи на принципи переваги хмарних технологій, вважаємо, що застосування системи Apache CloudStack у навчальному процесі вимагає супроводу та контролю з боку кваліфікованого фахівця.

Джерела

1. Биков В.Ю. Технології хмарних обчислень, ІКТ-аутсорсінг та нові функції ІКТ-підрозділів навчальних закладів і наукових установ / В.Ю. Биков // Інформаційні технології в освіті. – Випуск 10. – Херсон: ХДУ, 2011. – № 10. – С. 8-23..
2. Глазунова О.Г. Принципи формування «академічної хмари» сучасного університету на основі відкритих програмних платформ. // Інформаційні технології і засоби навчання. – 2014. – №5 (43). – Режим доступу: <http://journal.iitta.gov.ua/index.php/itlt/article/view/1096>
3. Munteanu V. Multi-cloud resource management: cloud service interfacing [Електронний ресурс] / V. Munteanu, C. Sandru, D. Petcu // Journal of Cloud Computing. – 2014. – Режим доступу до ресурсу: <http://journalofcloudcomputing.springeropen.com/articles/10.1186/2192-113X-3-3>
4. Олексюк В. П. Впровадження технологій хмарних обчислень як складових ІТ-інфраструктури ВНЗ. [Електронний ресурс] / В. П. Олексюк // Інформаційні технології і засоби навчання. – 2014. – №3. – Режим доступу: <http://journal.iitta.gov.ua/index.php/itlt/article/view/1042#.U7KuwPkrbPA>.
5. Apache CloudStack Documentation: open source cloud computing [Електронний ресурс]. — Режим доступу : http://cloudstack.apache.org/docs/en-US/Apache_CloudStack/4.2.0/html/Installation_Guide/cloud-infrastructure-concepts.html

**Використання системи GEOGEBRA у процесі навчання
математики в контексті впровадження ІКТ в освіту**
Рафальська М.В., Лященко Г.М.

*Національний педагогічний університет імені М.П. Драгоманова, спеціалізована
школа №260 м. Києва, m.v.rafalska@npu.edu.ua, luashchemko@gmail.com*

Publication is devoted to the problem of integration ICT in the secondary school. It presents the main ways of using GeoGebra in the teaching and learning Mathematics with an emphasis on the examples.

Необхідність впровадження ІКТ у навчальний процес вже ні в кого не викликає сумнівів. Разом з цим, актуальною проблемою є пошук ефективних шляхів впровадження новітніх засобів навчання, яке б було педагогічно доцільним і виваженим, сприяло досягненню цілей навчання. Спостереження за діяльністю вчителів математики показує, що наразі гостро стоїть питання належної їх підготовки до розробки навчальних ситуацій та організації різного роду діяльності учнів на уроках з використанням засобів ІКТ.

У цій публікації висвітлені питання застосування системи GeoGebra у процесі навчання математики, зокрема для організації дослідницької діяльності учнів.

GeoGebra є вільно поширювальним програмним засобом до складу якого внесено систему динамічної геометрії, систему комп'ютерної алгебри, електронні таблиці та ін. Інтерфейс програми реалізовано багатьма мовами, зокрема українською. Систему можна без труднощів встановити на

комп'ютері або використовувати онлайн-версію [1]. Програма легка для опанування учнями, у її середовищі можна розв'язувати різні класи математичних задач, що зумовило її широке поширення і використання у навчальних закладах багатьох країн світу.

Розглянемо шляхи використання GeoGebra у процесі навчання математики і дамо їм коротку характеристику, ілюструючи на прикладах.

1. Ознайомлення з основними теоремами математики та їх доведенням. Доведення відіграють значну роль в математиці, як у науці, та як у навчальному предметі в школі. На уроках математики з їх допомогою розкривається зміст дослідницької діяльності, демонструються правила ведення логічних міркувань та формулювання умовиводів. Тому важливо, щоб ознайомлення учнів з теоремами з курсу математики відбувалось не формально та сприяло розумінню учнями основних понять та методів доведення в математиці.

Так, у разі вивчення теореми Піфагора, можна запропонувати учням декілька динамічних моделей, розроблених у середовищі GeoGebra, для демонстрації різних способів її доведення (Рис. 1, 2).

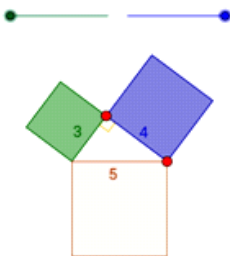


Рис.1

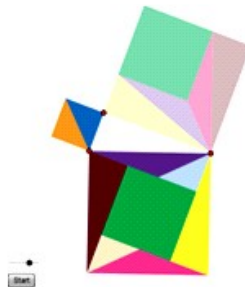


Рис. 2

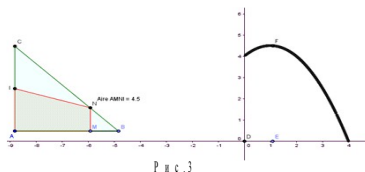
Використання наочності з елементами анімації активізує навчально-пізнавальну діяльність учнів, сприяє кращому запам'ятовуванню відомої теореми.

2. Побудова геометричних фігур та їх комбінацій. Використання засобів динамічної геометрії для побудови геометричних фігур на площині сприяє засвоєнню учнями їх основних властивостей та співвідношень між елементами, виявленню помилок у міркуваннях учнів та роботі над ними.

Так, завдання на побудову дотичної до кола у системі GeoGebra (без використання інструменту «Дотична») допоможе виявити учнів, які не розуміють або не пам'ятають властивостей дотичної до кола і будують її у зошитах «на око» (шляхом підставлення лінійки до кола). Переміщення кола на площині або зміна його радіуса у середовищі програми переконає учнів у хибності своїх міркувань.

3. Проведення досліджень. Використання засобів динамічної геометрії GeoGebra дає змогу учням експериментувати з геометричними фігурами,

висловлювати гіпотези щодо співвідношення довжин відрізків, величин кутів та площ фігур, для подальшого їх доведення або спростування аналітичними методами.



Приклад. Трикутник ABC – прямокутний ($\angle A = 90^\circ$), $AB=AC=4$, точка I – середина сторони AC, точка N – точка перетину MN і CB, $MN \perp AB$. Знайти положення точки M при якому SAMNI буде найбільшою.

Рухаючи точку M вздовж відрізка AB та фіксуючи значення SAMNI, учні висловлюють гіпотези щодо можливого положення точки M (Рис. 3). Для кращого розуміння як змінюється значення площі в залежності від положення точки M, доцільно використати графік функції, що відображає цю залежність (точка F переміщується у разі зміни положення точки M на AB). Легко бачити, що значення площі спочатку збільшується від 0 до 4,5, а потім зменшується до 4. Таким чином, функція досягає свого максимального значення 4,5 в точці $x=1$, де x – відстань від точки M до точки B. Перевіримо нашу гіпотезу.

Очевидно, що чотирикутник AMNI – трапеція з основами AI та MN і висотою AM. $AI=2$, довжини MN і AM змінюються в залежності від положення точки M. Задамо аналітично залежність SAMNI від довжини

$$S(x) = \frac{(4 + 2x)(4 - x)}{4} = \frac{16 + 4x - 2x^2}{4}$$

відрізка MB, $MB=x$: та дослідимо її на максимум. Маємо: $x_{\max}=1$, звідси $AM=3$, $MB=1$.

4. Розв'язування практичних задач. Для розкриття практичної значущості знань та умінь з математики учням доцільно пропонувати задачі з життя. Використання системи GeoGebra для моделювання деякого процесу чи явища дає змогу дослідити його для різних значень початкових даних (параметрів).

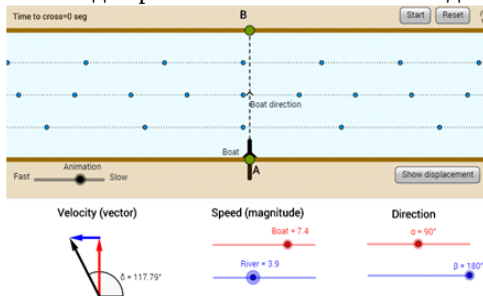


Рис. 4

Приклад. Скільки часу знадобиться човну для того, щоб перепливати річку, якщо власна швидкість човна v_1 , а швидкість течії v_2 ; яким буде кут відхилення?

У процесі розв'язування задачі, використовуючи розроблену динамічну модель (Рис. 4), учні відповідають на такі запитання:

1. Яка залежність між швидкістю човна (течії) та часом за який він долає відстань від А до В?
2. Як впливає швидкість човна (течії) на кут відхилення від обраного курсу?
3. Яким має бути напрямок руху човна, щоб з А потрапити в В?
4. Якими величинами є v_1 і v_2 ?
5. Яка векторна рівність має місце ?

Організації позакласної діяльності учнів. З метою розвитку творчого потенціалу учнів та формування в них інтересу до вивчення математики в позакласній роботі доцільно розглядати задачі на мозаїки та паркетти; побудову кривих, заданих параметрично та в полярній системі координат тощо. Використання GeoGebra при цьому дає змогу унаочнити розглядувані поняття, зробити процес навчання цікавим і різноманітним.

Джерела:

1. GeoGebra [online]. – Режим доступу: <https://geogebra.org/>

Можливості верстання математичних книжок за допомогою системи L_{yx} Репецький В.С.

Національний університет “Львівська політехніка”, кафедра інформаційних технологій видавничої справи, vldko96@gmail.com

In this publication were analyzed the features of typesetting mathematical books using L_{yx}. L_{yx} is a good combination of flexible markup language LaTeX and easy-to-use graphical interface.

Верстка математичних книг має свої особливості. Текст цих видань є ускладнений таблицями та формулами. Під час набору книг з таким умістом слід дотримуватись спеціальних правил, недотримання яких суттєво впливає на читабельність тексту.

Щодо набору формул, зазвичай використовують наступні вирішення: пропріетарне умовно-безкоштовне програмне забезпечення MathType, що є типовим редактором формул для нових версій офісних пакетів Microsoft та Macintosh, або ж вставка копій екранів з формулами (чи їх більш адекватніший експорт за допомогою стандартних засобів) з різноманітних систем комп'ютерної алгебри, на кшталт Wolfram Mathematica. Названі

методи мають наступні недоліки:

- проблеми у разі створення PDF-документів з формульним умістом;
- необхідність використання декількох програмних засобів;
- недосягнення “книжного” вигляду формул;
- платна основа програмного забезпечення;
- відсутність належної кросплатформності.

Потужною альтернативою цим засобам є використання мови розмітки даних LaTeX, яка є стандартним забезпеченням для високоякісного набору та верстки наукових видань. Типовими контраргументами цьому вирішенню є відсутність візуального редактора та необхідність безпосереднього володіння мовою розмітки. Але програмний продукт LyX робить неактуальними ці зауваження. Будучи кросплатформним текстовим процесором, що притримується парадигми WYSIWYM (англ. what you see is what you mean — бачите те, що мали на увазі), ця програма надає користувачеві численні переваги.

Верстка книг з математичним умістом у цьому редакторі є простою та очевидною. Серед основних засобів, які передбачені для цього середовищем, можна виділити наступні:

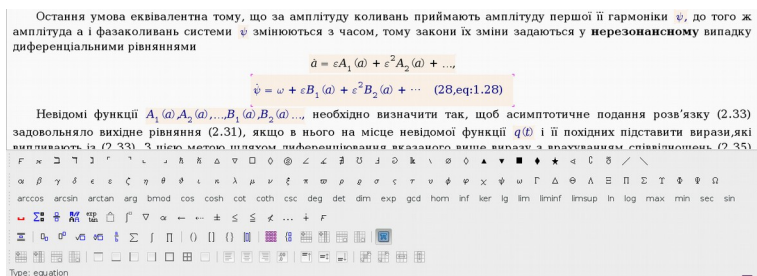


Рис. 1: Виведені панелі інструментів

1. Можливість виведення панелей з наборами типових математичних функцій, грецьких літер, дужок, матриць, спеціальних символів, реалізації вставки та редагування таблиць (Рис. 1) за допомогою пунктів меню Перегляд — Панелі інструментів;
2. Добре продумане меню Вставка, що дає змогу обирати об’єкти різноманітних категорій для вставки, зокрема користувач може обрати необхідне йому оточення для формул (Рис. 2);
3. Можливість швидкої вставки умісту з таблиць, створених у інших текстових редакторах, чи з Веб-сторінок (були випробувані вставка з таблиць реалізованих у Microsoft Word та з Веб-сторінок) за допомогою переміщення курсора в одну з клітинок таблиці та з застосуванням комбінації клавіш Ctrl+Shift+V;

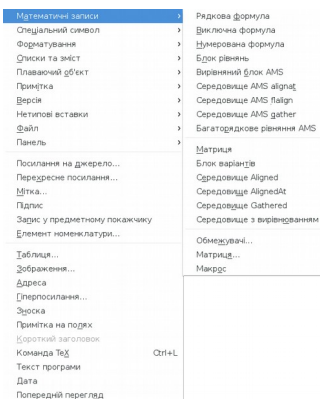


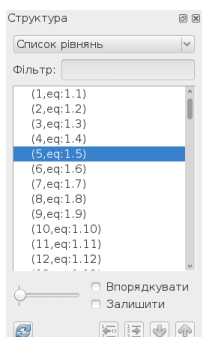
Рис. 2: Меню Вставка

4. Швидке додавання позначок та перехресних посилань до формул, які нумеруються автоматично (або ж відповідно до встановлених правил) за допомогою меню Вставка чи відповідних кнопок на додатковій панелі інструментів (Рис. 3).



Рис. 3: Вставка позначок та перехресних посилань

5. Забезпечення перегляду списку пронумерованих таблиць та формул з можливістю швидкого переходу до їх місцезнаходження у тексті (Рис. 4), за допомогою увімкнення панелі структури у меню Документ — Структура;



де p і q дійсні числа .
У частковому випадку, якщо $x = 1$, формула (Ref: e першого роду

$$B_x(p, q) = \{0\} \{x\} \{f\} t^{p-1} (1-b)^{q-1}$$

тобто повної Beta-функції.
Для всіх x з інтервалу $[0,1]$ функції B_x (Ref: eq:1.3), (Ref: eq:1.4) , є додатними і задовольняють ум

$$0 \leq B_x(p, q) \leq B_1(p, q)$$

$$B_x(p, q) = B_1(p, q) \cdot B_{1-x}$$

Розглянемо два випадки, а саме

$$p = \frac{1}{n+1}, q = \frac{1}{m+1}; \quad (5,$$

Рис. 4: Перегляд списку формул за допомогою навігатора

Загалом, описаний програмний продукт можна назвати надзвичайно перспективним для створення математичних текстів, статей, підручників та наукових монографій. Цьому сприяє: належність до вільного програмного забезпечення; наявність безкоштовних інсталяцій не тільки для операційної

системи Linux, але і інших, зокрема Microsoft Windows; часті оновлення; можливість перегляду та окрема опція безпосереднього вставлення TeX-коду; меню налаштувань клавіатурних скорочень. Також підтвердженням цього є незаперечна популярність TeX серед математиків. Тому графічна оболонка над TeX'ом обов'язково з плином часу припаде до смаку математикам.

Джерела

1. LyX – The Document Processor. <https://www.lyx.org/>
2. LyX Wiki. <https://wiki.lyx.org/>

Універсальний конвертер на базі ОС Андроїд

Рухолла О., Павлюк О.М.

Національний Університет «Львівська Політехніка», ruholkin@gmail.com

In these theses seeks to create an application that will facilitate learning for pupils and students. As a result, the author creates a universal converter for smartphones based on Android OS .

На сьогодні смартфонами користується більша половина користувачів мобільного зв'язку, особливо популярними вони є серед молоді через свої можливості та мобільність[1]. Програмне забезпечення для розробки додатків безкоштовне та доступне для широкого загалу, що сприяє різноманіттю програм на ринку.

Метою даної роботи є створення універсального конвертера для полегшення навчання студентів та учнів шкіл. Оскільки потреба у швидких обчисленнях була і буде завжди, то необхідно створити додаток, який швидко конвертуватиме різноманітні величини. Щоб додаток витримав конкуренцію на ринку, важливо подбати і про його дизайн.

Серед найпопулярніших мобільних додатків, які існують у вільному доступі для платформи Android, є такі:

- Крайній конвертер [2]. Спеціалізується на перетворенні математичних та фізичних величин (прискорення та кутове прискорення). З об'єктивної точки зору, така деталізація призведе до погіршення розуміння школярами навчальної інформації. Також є незручним дизайн та вивід результатів і занадто нагромаджене подання інформації.
- Unit Converter [3]. Загальний конвертер, який не містить всіх необхідних категорій для навчання. Він є незручним із застарілим дизайном та нагромадженим поданням інформації.
- Конвертер. [4] Містить велику кількість зайвих та незрозумілих для перетворення величин, таких як часовий пояс, яскравість або магніт. Він включає конвертер валют, який не може бути ефективним через застарілість поступлення інформації. Також він включає надмірну кількість рекламних оголошень.

- Easy currency converter [5]. Конвертер валют, що містить всі грошові одиниці світу з виведенням графіку росту та падіння валюти за останній місяць. Цей конвертер має незручний дизайн.

Оглянувши існуючі програми та зробивши аналіз, виникла необхідність у створенні ефективного та стильного конвертера для школярів та студентів. Всі подібні додатки вузькопрофільні, тобто оперують тільки з десятковими та двійковими числами і не містять необхідних величин, або всіх категорій. У розроблюваному конвертері необхідно врахувати всі недоліки існуючих програм. При цьому, він не повинен перешкоджати навчанню, виконуючи складні обрахунки для різноманітних задач. Конвертер повинен полегшувати елементарні обрахунки на рівні калькулятора, чи "розумного" годинника, тобто бути максимально простим без зайвого нагромодження інформації та реклами. Користувач повинен вибирати вхідну та вихідну величину, а не шукати необхідний результат серед десятків інших. Особливу увагу потрібно приділити створенню простого дизайну, щоб витримати конкуренцію на ринку та зацікавити широку аудиторію.

Спочатку було створено модель майбутньої системи для візуалізації подальшої роботи. У Use Case діаграмі (див рис.1.) представлено взаємодію користувача з програмою: робота з меню та величинами.

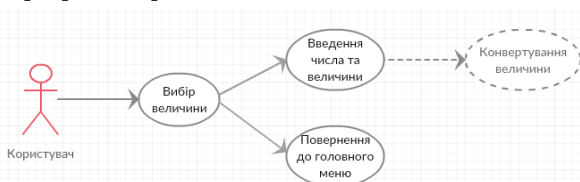


Рис.1. Use Case Діаграма роботи з меню та величинами

Програма реалізована для ОС Android. Це операційна система від компанії Google для різноманітних гаджетів. Її створили на основі ядра Linux, для роботи вона використовує віртуальну машину Dalvik. Для розробки програмного продукту було обрано середовище Android Studio, яке було створене та представлено у 2013 році як претендент на зміну середовища Eclipse. Воно призначене для виконання завдань, які виникають під час розробки додатків для ОС Android. В ньому існують засоби для покращення тестування готових додатків на сумісність з різноманітними версіями операційної системи, які працюють з екранами різної роздільної здатності. Середовище підтримує колекцію типових елементів для інтерфейсу, щоб пришвидшити створення додатків. Також можливо під час розробки побачити вигляд інтерфейсу на різних версіях ОС, різних розширеннях та орієнтаціях екрану. Редагувати інтерфейс можна просто та швидко через зручний візуальний редактор.

Для створення дизайну було використано Adobe Photoshop CC. Через сумісність програми з версіями ОС 2.2 Froyo і вище, є недоступні нові

вигляди інтерфейсу, такі як режим повного вікна та нові версії введення-виведення тексту. При завантаженні відкривається головне меню з вибором категорії. При виборі створюється нове вікно, де містяться поля для тексту та вибору величин для конвертації. При створенні дизайну обрано один колір фону і прямокутні кнопки для зручності використання на мобільних пристроях з малим розміром екрану, наприклад "розумного" годинника.

Для меню та кожної категорії існують власне вікно (Activity) та клас, назва залежить від призначення:

- Activity_menu містить перелік категорій та функцію переходу на нове вікно.
- Activity_chas, Activity_ploshcha, Activity_vaga, Activity_dovzhyna, Activity_shvydkist, Activity_temperatura, Activity_objem містять поля для введення та виведення величин та їх вибір у вигляді спадного списку.
- Activity_chysla містить поля для введення числа та основи, а також поле для виведення результату конвертування.

Класи та вікна побудовані ідентично. Під час обробки події натиснення кнопки «Конвертувати» програма перевіряє значення вибору у спадному списку (spinner) та проводить необхідні перетворення з введеним числом (editText), а результат виводить у відповідне поле (plainText). У разі обрання однакових значень величин програма не проводить обрахунків. При введенні нечислових значень програма повідомляє про помилку.

Програмний продукт тестувався на декількох пристроях з різними версіями ОС Android, таких як 2.2 Froyo, 4.4 KitKat, 5.1 Lollipop. Додаток працює однаково швидко на всіх пристроях та ОС і коректно перетворює величини. Під час внесення даних можна користуватися лише цифровою панеллю введення. У разі введення зайвих знаків виводиться повідомлення користувачу внизу екрану про неправильність вводу.

В розробленому конвертері є доступні перетворення наступних величин:

- Числа. Будь-які числа, які належать до будь-яких систем числення.
- Час. Години, хвилини, секунди.
- Довжина. Кілометри, метри, дециметри, сантиметри.
- Об'єм. м3, дм3, см3, мм3, літри, мілілітри.
- Вага. Тонна, центнер, кілограм, грам.
- Швидкість. Км/год, км/с, м/хв, м/с.
- Температура. Шкала Цельсія, Фаренгейта, Кельвіна.
- Площа. Ар, га, км2, м2.

Копії екранів виконання програми представлені на рис.2.

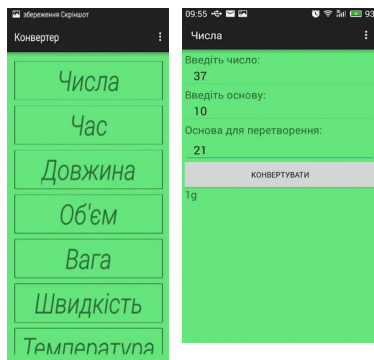


Рис. 2 а) меню програми б) перетворення числа

Висновки. Створено простий та ефективний мобільний додаток для смартфонів на базі ОС Android, який конвертує різноманітні величини. Даний конвертер охоплює максимально широке коло смартфонів, враховуючи старі версії операційної системи. Категорії для перетворення обрано з врахуванням того, щоб не зашкодити навчальному процесу. Окремим пунктом є перетворення чисел у різні системи числення, що є необхідним для студентів першого курсу технічного спрямування. Особливу увагу приділено розробці простого та зрозумілого інтерфейсу.

В майбутньому заплановано додати конвертер валют з прив'язкою до курсів купівлі та продажу усіх найбільших банків.

Джерела:

- [1] Вікіпедія, <https://uk.wikipedia.org/wiki/Android>
- [2] Google play, <https://play.google.com/store/apps/details?id=com.kuzmin.konverter>
- [3] Google play, <https://play.google.com/store/apps/details?id=kr.sira.unit>
- [4] Google play, <https://play.google.com/store/apps/details?id=com.androidapps.unitconverter>
- [5] Google play, <https://play.google.com/store/apps/details?id=com.easy.currency.extra.androary>

Вивчення природничих дисциплін з використанням PhET-модельовання

Дмитрів М.В., Твердохліб І.А.

Національний педагогічний університет імені М.П. Драгоманова, Dmytriw@ukr.net, IgTverd@ukr.net

У роботі визначено напрями використання проекту PhET при вивченні природничих дисциплін, наведено загальну характеристику проекту PhET та деякі аспекти його використання для підтримки вивчення природничих дисциплін.

В умовах становлення інформаційного суспільства, що супроводжується швидкими темпами науково-технічного прогресу, постійним вдосконаленням існуючих та створенням якісно нових технологій, широкої популярності набуло комп'ютерне моделювання як на виробництві, так і в системі освіти.

Моделювання розглядають як заміну вивчення явища в реальності вивченням аналогічного явища на моделі (уявлювана або матеріально реалізована система, яка, відображаючи об'єкт дослідження, здатна замінювати його так, що її вивчення дає нові відомості про об'єкт). Навчальна модель розглядається як система (матеріальна або уявлювана), що здатна відтворювати зовнішній вигляд чи внутрішню структуру об'єкта з метою більш глибокого його пізнання, коли безпосереднє вивчення цього об'єкта в умовах школи чи вищого навчального закладу становить значні труднощі або зовсім неможливе.

Із масовим впровадженням інформаційно-комунікаційних технологій в навчальний процес, все більшого розвитку дістало використання комп'ютерного моделювання як елементу уроку, особливо при вивченні природничих дисциплін. Комп'ютерне моделювання передбачає створення програмного середовища, що поєднує в собі на основі математичної моделі явища чи процесу засоби інтерактивної взаємодії з об'єктом дослідження і розвинуті засоби відображення даних [1]. Навчальні комп'ютерні моделі є віртуальними моделями, призначеними для формування в суб'єкта навчання відповідних пізнавальних умінь на основі представлення предмета навчання [2].

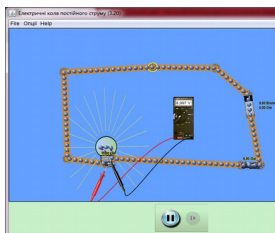
Одним із видів комп'ютерного моделювання є симуляція – процес розробки моделі реальної чи уявної системи і проведення експериментів з моделлю [4]. Симуляцію проводять з різною метою – тренування та навчання персоналу, тестування технології в граничних умовах, тестування безпеки, розваги (відеоігри, симуляція невагомості). Також симуляції використовуються науковцями для здійснення експериментів, які неможливі в реальності, коли експерименти над реальною системою неможливі, через її недосяжність, чи небезпеку або високу вартість таких експериментів [3].

Використання комп'ютерного моделювання дає змогу спостерігати й досліджувати явища й процеси в динаміці їх розгортання, здійснювати багаторазові випробування моделі, одержувати різноманітні кількісні показники в числовому або графічному вигляді, зокрема такі, що вимагають виконання складних математичних обчислень.

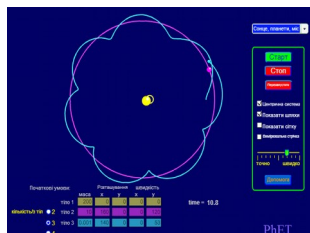
Комп'ютерне моделювання є унікальним інструментом пізнання при вивченні таких природничих дисциплін як фізика, хімія, біологія, геологія, математика тощо. Тому, вагоме місце в процесі вивчення природничих дисциплін повинне займати використання готових комп'ютерних моделей, віртуальних лабораторій, програмних засобів для створення та дослідження моделей.

Серед усього різноманіття програмного забезпечення для комп'ютерного

моделювання вагоме місце займає вільно поширюваний програмний засіб (проект) PhET (Physics Education Technology), що розроблений Університетом Колорадо і поширюється з ліцензією GNU/GPL. Метою створення цього проекту є інтерактивне моделювання фізичних явищ для використання моделей в процесі навчання.



а) складання електричного кола постійного струму



б) дослідження руху Сонця, Землі і Місяця «Моя сонячна система»

Рис. 1. Приклади фізичних моделей в PhET

Всі Phet-моделі знаходяться у вільному доступі на веб-сайті [5] Phet і прості у використанні. Вони написані з використанням Java та Flash, і можуть бути завантажені та відтворені за допомогою звичайного Веб-переглядача, навіть, якщо ці додатки не встановлені на комп'ютерах користувачів.

На сайті всі моделі впорядковані і можуть бути знайдені спеціальними інструментами пошуку за:

тематичними блоками (наприклад, «Взаємодія атомів», «Використання батарей та акумуляторів», «Спектр», «Молекулярна теорія» тощо);

типом занять (наприклад, для лабораторних робіт, демонстрації на лекціях, для виконання домашніх робіт тощо);

рівнями (для учнів початкових класів, учнів середніх і старших класів, завдання підвищеної складності тощо);

мовою (всього більше 80 мов) [5].

Таким чином, використання PhET значно розширює межі застосування різних методів та моделей для дослідження процесів у різних сферах людської діяльності. Phet-симуляції є більш ефективними для формування в учнів концептуального розуміння наукових понять, їх взаємозв'язків, законів і формул, проте існує багато навчальних і наукових цілей щодо практичних занять, які не можуть бути вирішені за допомогою комп'ютерного моделювання.



Рис. 2. Симуляція α -розпаду

Джерела:

1. Яремчук О.М. Педагогічні ідеї вивчення штучної радіоактивності (з використанням засобів комп'ютерного моделювання) / О.М. Яремчук // Наукові праці Миколаївського державного гуманітарного університету імені Петра Могили комплексу "Києво-Могилянська академія". Сер.: Педагогічні науки. – 2008. – Т. 97, Вип. 84. – С. 36 – 39.
2. Оспенников Н.А. Школьный физический эксперимент в условиях развития компьютерных технологий обучения / Н.А. Оспенников // Вестник ПГПУ. Серия «ИКТ в образовании». – 2006. – Вып. 2. – С. 47 – 76.
3. Bransford, J.D., Brown, A.L. And Cocking, R.R. How People Learn, Brain, Mind, Experience, and School. — Washington, D.C.: National Academy Press, 2000.
4. Roger D. Smith, Simulation Article. Encyclopedia of Computer Science, 4th Edition, July 2000. [Електронний ресурс]. Режим доступу: <http://www.modelbenders.com/encyclopedia/encyclopedia.html>
5. PHET Interactive Simulations [Electronic Resource]. – Mode of access : URL: <https://phet.colorado.edu/>

Використання вільного програмного забезпечення для вивчення технічних дисциплін майбутніми учителями інформатики

Ткачук Г.В.

Уманський державний педагогічний університет імені Павла Тичини,
galanet82@gmail.com

В статті охарактеризовано вільне програмне забезпечення для моделювання комп'ютерних мереж, яке може бути використане для технічної підготовки майбутніх учителів інформатики. Запропоноване імітаційно-моделююче середовище GNS3 дає змогу побудувати комплексні мережеві проекти, перевірити працездатність тієї чи іншої топології, визначити кількість мережевого обладнання для комп'ютерного класу, протестувати роботу віртуальної комп'ютерної мережі.

Технічна підготовка майбутніх учителів інформатики базується на отриманні компетентностей, що пов'язані з використанням різноманітного технічного обладнання. Забезпечити наявність потрібної кількості пристроїв та різного роду устаткувань практично неможливо, оскільки це потребує значних матеріальних затрат, які на жаль, сучасна система освіти надати не може. Вирішення цієї проблеми може бути завдяки використанню імітаційно-моделюючих програмних засобів, які дають змогу працювати з відповідними пристроями, налаштовувати їх, тестувати, організовувати їх роботу. І хоч студент не має змоги працювати з реальним пристроєм, проте він зможе набути відповідних компетентностей, працюючи у віртуальному середовищі з відповідним технічним оснащенням.

В умовах сучасної освіти набуває популярності дистанційна освіта, яка дає змогу студенту готуватись до занять дистанційно, в комфортних умовах, не витрачаючи зайвих ресурсів. Тому вважаємо за потрібне використовувати таке програмне забезпечення, яке студент може сам завантажити та встановити на своєму комп'ютері і виконати завдання, які надаються викладачем. В цьому випадку актуальності набуває вільно поширюване програмне забезпечення, оскільки воно завжди доступне в мережі і може бути встановлене на комп'ютер користувача.

Дисципліна «Основи комп'ютерних мереж та систем» в Уманському державному педагогічному університеті імені Павла Тичини вивчається студентами напряму підготовки 6.040302 Інформатика на першому курсі та передбачає набуття базових компетентностей щодо організації та створення комп'ютерних мереж та систем, вивчення сучасного апаратного і програмного забезпечення. Студенти вивчають різні топології комп'ютерних мереж, мережеві архітектури, роботу комутаторів, мостів, маршрутизаторів.

Коли мова йде про створення комп'ютерної мережі, потрібно знати заздалегідь як вона буде працювати, оскільки обладнання, яке потрібно для цього придбати є недешевим. Для того, щоб мати можливість моделювати взаємодію між різними елементами мережі, можна використовувати програми для її емуляції. Тому для набуття практичних навичок роботи з таким обладнанням пропонуємо використовувати графічний симулятор мережі – GNS3 (Graphical Network Simulator), що поширюється вільно та має ліцензію GNU.

GNS3 дає змогу в режимі реального часу змоделювати віртуальну мережу будь-якої топології без необхідності використання реального мережевого обладнання. Особливістю GNS3 є те, що він є кросплатформним симулятором і може бути використаний на декількох платформах, зокрема підтримуються Windows, Linux, Mac.

GNS3 має графічний інтерфейс, що дає змогу вибирати кожний з елементів, які є частиною мережі та встановлювати його параметри. Це значно полегшує створення віртуальних лабораторій, оскільки наперед можна визначити кількість мережевого обладнання, його працездатність в тих чи

інших умовах, конфігурацію пристроїв. Важливою перевагою GNS3 є простота і зручність при створенні проектів для виконання лабораторних робіт.

Залежно від апаратної платформи, на якій буде використовуватись GNS3, можна побудувати комплексні проекти, що складаються з маршрутизаторів Cisco, Cisco ASA, Juniper, а також серверів під управлінням мережних операційних систем.

Недоліком використання GNS3 є те, що він потребує значних ресурсів процесора та оперативної пам'яті. Мінімальними рекомендаціями для комп'ютера, на який потрібно встановити GNS3 версії 1.4.4 є наявність 2-ядерного процесора (серії AMD або Intel), 4 Гб оперативної пам'яті, 1 Гб вільного простору на жорсткому диску. Тобто, якщо проектувати 10 маршрутизаторів на студентському комп'ютері, то виникають певні затримки в роботі. Проте вирішенням цієї проблеми є використання механізму Idle PC, який може знизити використання процесора. Крім того, можна встановити старішу версію (наприклад, 0.8.3.1), тоді потреби зменшуються (зокрема, достатньо буде 2 Гб оперативної пам'яті). Звісно, змоделювати серйозний проект не вийде, проте для навчання цілком достатньо. Для демонстрації складних топологій, виконаних в середовищі GNS3, викладач може використовувати головний комп'ютер, який має відповідну обчислювальну потужність. Крім того, в багатьох студентів домашній комп'ютер може володіти достатньою потужністю для встановлення останньої версії програми.

Крім того, GNS3 – це практично віртуальний маршрутизатор і для його запуску потрібне програмне забезпечення, тому виникає проблема у пошуку образів IOS/IPS/PIX/ASA/JunOS, оскільки вони є частиною комерційних продуктів відповідних компаній. Частково цю проблему можна вирішити, скопіювавши образ з наявного в комп'ютерному класі обладнання.

В цілому варто зазначити, що графічний симулятор мережі GNS3, хоч і має певні недоліки у використанні, проте є ефективним засобом навчання майбутніх учителів інформатики із створення та налагодження комп'ютерної мережі. Звісно, навчатись найкраще на реальному обладнанні, але за його відсутності можна отримати відповідну компетентність, використовуючи імітаційно-модельюючі програмні комплекси.

Створення електронних освітніх ресурсів засобами вільного програмного забезпечення

Величко В.Є.

ВНЗ «Луганський національний університет імені Тараса Шевченка», e-mail: vladislav.velichko@gmail.com

We consider the requirements for creating electronic educational resources by means of free and open source software. The analysis of existing free and open source software for each type of electronic educational resources.

Концепція відкритої освіти, яку підтримує рух за вільне та відкрите програмне забезпечення передбачає наявність достатньої кількості навчальних матеріалів високої якості як з технічної точки зору, так і з методичної. З боку держави вимоги до навчальних матеріалів, які можуть бути використані в електронному навчанні, викладені в розробленому Положенні про електронні освітні ресурси (ЕОР) [1] та містять наступні умови. По-перше, це відповідність програмі навчальної дисципліни, для вивчення якого розроблявся ЕОР. Враховуючи часті зміни в навчальних програмах як до змісту, так і до кількісних характеристик необхідно мати можливість коригувати існуючі ЕОР, тобто мати легальні засоби їх редагування. По-друге, наявність відповідних методичних рекомендацій щодо використання ЕОР у професійній діяльності. Такі рекомендації матимуть практичну значущість коли в їх розробці приймають участь фахівці-практики, а не тільки розробники та фахівці-теоретики. По-третє, це дотримання чинних санітарних норм, ергономічних та програмно-технічних вимог до ЕОР. Відповідність ЕОР таким вимогам зазвичай визначається тільки під час її практичного використання з поступовим виправленням помилок та недоліків, а отже, знову приходимо до розуміння того, що необхідно мати доступні засоби створення ЕОР. Остання четверта вимога говорить про дотримання законодавства України щодо захисту авторських прав. Причому ці права мають дотримуватись не тільки стосовно створеного ЕОР, а й до засобів створення ЕОР.

Ліцензійна чистота програмного забезпечення в освіті й до нині не має державного вирішення незважаючи на прийняті законодавчі акти та регулятивні документи. За даними дослідження BSA Global Software Survey виконаного за підтримки IDC в 2013 році в Україні 83% програмного забезпечення, яке використовується, є неліцензійним, при цьому в Європейському Союзі неліцензійного програмного забезпечення тільки 31%, в США – 18% [2]. Для подолання ситуації, що склалася, необхідно, в першу чергу, розпочати активне використання вільного програмного забезпечення в освітній діяльності як під час підготовки фахівців з навчання, так і під час надання базової освіти. І першим кроком таких змін може слугувати

використання вільного та відкритого програмного забезпечення під час підготовки ЕОР.

Під електронними освітніми ресурсами розуміють навчальні, наукові, інформаційні, довідкові матеріали та засоби розроблені в електронній формі та представлені на носіях будь-якого типу або розміщені у комп'ютерних мережах, які відтворюються за допомогою електронних цифрових технічних засобів і необхідні для ефективної організації навчально-виховного процесу в частині, що стосується його наповнення якісними навчально-методичними матеріалами [1]. Розглянемо види ЕОР та існуючі засоби їх створення серед вільного та відкритого програмного забезпечення.

Електронний документ, електронне видання чи електронний аналог друкованого видання створюється текстовими редакторами, текстовими процесорами та системами комп'ютерної верстки яких дуже багато. Створені файли можуть розповсюджуватись як у власних форматах, так і у відкритих поширених форматах, що не суперечить політиці вільного та відкритого програмного забезпечення.

Електронні дидактичні демонстраційні матеріали, що призначені для супроводу навчально-виховного процесу представляють собою різноманітні презентації, схеми, відео – аудіозаписи тощо. Для створення цього виду ЕОР існує різноманіття графічних програм та програм відображення графічного матеріалу, програм створення електронних презентацій, програм запису, обробки та генерації відео – й аудіозаписів.

До більш складного типу ЕОР відноситься інформаційна система як організаційно впорядкована сукупність документів та інформаційних технологій, що реалізують інформаційні процеси та призначені для зберігання, обробки, пошуку, розповсюдження, передачі та надання інформації. Враховуючи багатогранність та складність інформаційних систем, їх реалізація може бути найрізноманітнішою від веб-сервера в поєднанні з сервером баз даних та мовою виконання сценаріїв реагування на запити до бібліотечних систем, локальних баз даних тощо. Такими ж методами створюються депозитарії електронних ресурсів.

До ЕОР відносяться також системи комп'ютерного тестування, які є невід'ємною частиною сучасних освітніх технологій та які поділяються на локальні та глобальні. Локальні системи тестування дають змогу, використовуючи різноманіття видів тестів, доволі об'єктивно оцінювати рівень знань, умінь та навичок. Вільне програмне забезпечення надає розробникам та користувачам широкий набір можливостей зі створення, проведення та аналізу отриманих результатів (KEduca, iTest, Hot Potatoes, Шёлковий тест та ін.).

Вільне та відкрите програмне забезпечення дає змогу створювати наступний вид ЕОР – електронні словники. Можливості програм створення електронних словників досить різноманітні та полягають не тільки у створенні упорядкованого переліку мовних одиниць доповнених

відповідними довідковими даними, а й під'єднання додаткових словників і енциклопедій, підтримка мультимедійних даних і переклад слів, інтеграція в існуюче прикладне програмне забезпечення (StarDict, GoldenDict, Fluentizer та ін.).

Електронні довідники та споріднені до них електронні бібліотеки цифрових об'єктів, як різновиди інформаційних систем, можуть бути створені як відповідними методами, так і спеціалізованим програмним забезпеченням створення колекцій документів (Greenstone Librarian Interface, RSSOwl, KnowledgeTree та ін.).

Електронні навчальні посібники, підручники та методичні матеріали є навчальними виданнями різного спрямування, але спорідненої технології створення, яка має різноманіття від текстового процесора до спеціалізованої оболонки з інтерактивними компонентами (Debug Mode Wink, Course Lab, Ren'Py та ін.).

Наступним видом ЕОР є дистанційні курси для яких співтовариство розробників та користувачів вільного та відкритого програмного забезпечення розробило системи, які стали стандартом де-факто в цьому напрямку.

Останнім видом ЕОР є електронний практикум, тобто інтерактивна інформаційна система демонстрації моделей природних та штучних об'єктів, процесів та їх властивостей. Моделювання може бути представлене вільним та відкритим програмним забезпеченням як розширення систем комп'ютерної математики та як окремі програмні продукти.

Аналіз можливостей вільного та відкритого програмного забезпечення щодо створення ЕОР будь-якого виду дає змогу зробити наступні висновки. Вільне та відкрите програмне забезпечення використовуючи різноманітні технології дає змогу створювати електронні освітні ресурси будь-якої складності. Широке застосування вільного та відкритого програмного забезпечення дасть змогу не тільки збільшити його популярність через використання результатів його роботи, а й заохотити суб'єкти освітньої діяльності до більш широкого використання програмного продукту.

Література

1. Про затвердження Положення про електронні освітні ресурси [Електронний ресурс]. – Режим доступу : <http://zakon3.rada.gov.ua/laws/show/z1695-12>.
2. The Compliance Gap: BSA GLOBAL SOFTWARE SURVEY [Електронний ресурс]. – Режим доступу : http://globalstudy.bsa.org/2013/downloads/studies/2013GlobalSurvey_Study_en.pdf.

Впровадження вільного програмного забезпечення у систему підвищення кваліфікації науково-педагогічних працівників як системна проблема

Воронкін О. С.

Державний заклад “Луганський національний університет ім. Т. Шевченка”, м. Старобільськ, alex.voronkin@gmail.com

У матеріалах доповіді розглядається проблема низької готовності вишів до використання вільного програмного забезпечення. Одним з напрямків вирішення даної проблеми є здобуття нової компетенції у сфері вільного програмного забезпечення та вдосконалення наявних загальнокультурних і професійних компетенцій, необхідних науково-педагогічним працівникам для професійної діяльності та підвищення професійного рівня в рамках наявної кваліфікації.

На сьогоднішній день навчальний процес у більшості вишів України супроводжується домінуванням комерційних програмних засобів (пропріетарного програмного забезпечення) [1]. У роботі [2] дослідник О. Крутієнко виокремлює ряд причин широкого використання пропріетарного програмного забезпечення, як-от: а) стереотипи, пов’язані з вільним і безкоштовним програмним забезпеченням: ненадійність, запутаність, складність у засвоєнні та роботі; б) більшість користувачів вчать працювати з програмами, а не з об’єктами, внаслідок чого інтерфейс іншої програми їх відлякує; в) легкість доступу до «піратського» програмного забезпечення і майже відсутність контролю за його обігом та використанням. Одним з напрямків вирішення цієї проблеми є розроблення й впровадження відповідних навчальних програм у закладах післядипломної освіти.

Відповідно до статті 47 чинного Закону України «Про вищу освіту» [3] післядипломна освіта – це спеціалізоване вдосконалення освіти та професійної підготовки особи шляхом поглиблення, розширення та оновлення її знань, умінь і навичок на основі здобутої раніше вищої освіти (спеціальності) або професійно-технічної освіти (професії) та практичного досвіду. Післядипломна освіта містить: а) спеціалізацію – профільна спеціалізована підготовка з метою набуття особою здатності виконувати окремі завдання та обов’язки, що мають особливості в межах спеціальності; б) перепідготовку – професійне навчання, спрямоване на оволодіння іншою професією працівниками, які здобули первинну професійну підготовку; в) підвищення кваліфікації – підвищення рівня готовності особи до виконання її професійних завдань та обов’язків або набуття особою здатності виконувати додаткові завдання та обов’язки шляхом набуття нових знань і вмінь у межах професійної діяльності або галузі знань; г) стажування – набуття особою досвіду виконання завдань та обов’язків певної професійної

діяльності або галузі знань.

Викладач, який успішно пройшов навчання за програмою післядипломної освіти, отримує відповідний документ, зразок якого затверджується засновником (засновниками) навчального закладу або уповноваженим ним (ними) органом.

Необхідність інноваційного розвитку програм перепідготовки (підвищення кваліфікації) науково-педагогічних працівників пояснюється об'єктивними прискореним процесом старіння знань, швидким розвитком ІКТ. У той же час існуючі програми найчастіше орієнтовані на використання пропріетарних продуктів і технологій, що вказує на необхідність внесення системних змін у систему підвищення кваліфікації [4]. Висловлюються думки, що освіта у сфері ІКТ у найближчій перспективі має стати платформно незалежною [5]. Таким чином, проблемою сьогодення є здобуття нової компетенції у сфері вільного програмного забезпечення та вдосконалення наявних загальнокультурних і професійних компетенцій, необхідних для професійної діяльності та підвищення професійного рівня в рамках наявної кваліфікації.

Для вирішення вищезазначених проблем доцільно запропонувати ряд заходів, серед яких: а) розроблення й впровадження програм підвищення кваліфікації науково-педагогічних працівників щодо застосування вільного програмного забезпечення в освіті та науковій діяльності; б) популяризація вільного програмного забезпечення; в) підтримка та актуалізація у наукових і експертних колах уваги до питання вільного програмного забезпечення як унікальної можливості більш інтенсивно долучитися до сучасних світових процесів [6]; г) підтримка закладів, в яких вже використовується вільне програмне забезпечення; ґ) стимулювання науково-педагогічних працівників, які використовують у власній діяльності вільне програмне забезпечення; д) визначення вишів і закладів післядипломної освіти, в яких будуть реалізовуватись пілотні проекти із впровадження вільного програмного забезпечення.

Розгляд піднятої проблеми підтверджує потребу у розробці відповідних програм підвищення кваліфікації (у тому числі дистанційних курсів), які на нашу думку мають охоплювати змістовні, психолого-педагогічні, організаційні, методичні, методологічні, правові та етичні аспекти використання програмного забезпечення.

Джерела

1. Коломієць А. М. Основні напрями використання вільного програмного забезпечення в закладах освіти зарубіжжя [Електронний ресурс] / А. М. Коломієць. – Режим доступу : http://zerna.at.ua/publ/osvita/osnovni_naprjami_vikoristannja_vilnogo_programnogo_zabezpechennja_v_zakladakh_osviti_zarubizhzhja/1-1-0-49.
2. Крутієнко О. М. Використання вільного і безкоштовного програмного забезпечення в навчальному процесі ЗНЗ / О. М. Крутієнко // Комп'ютер у школі

- та сім'ї. – 2014. – № 5. – С. 20–21.
3. Закон України про вищу освіту [Електронний ресурс]. – Режим доступу : <http://zakon2.rada.gov.ua/laws/show/1556-18>.
 4. Дочкин С. А. Использование пакетов свободного программного обеспечения в дополнительном профессиональном образовании / С. А. Дочкин // Научное обеспечение системы повышения квалификации кадров. – 2011. – Вып. 2 (7). – С. 46–52.
 5. Рудик О. Перспективи платформонезалежної та Linux-орієнтованої інформатики в Україні [Електронний ресурс] / О. Рудик. – Режим доступу : <http://www.kievoit.ippo.kubg.edu.ua/kievoit/c130829/4-Rudyk.html>.
 6. Карпенко М. Перспективи та можливості впровадження вільного програмного забезпечення в навчальних закладах та державних установах України [Електронний ресурс] / М. Карпенко, М. Кияк. – Режим доступу : <http://old.niss.gov.ua/MONITOR/june2009/15.htm>.

Розробка Android-додатків на Android Studio під Ubuntu на Kotlin ***Яхненко О.В***

*Полтавський національний педагогічний університет імені В.Г. Короленка,
yahnencko.sasha@gmail.com*

В доповіді розглядаються рекомендації до впровадження вільного програмного забезпечення в освіті з метою спрямування користувачів, які стали перед вибором засобів розробки. З поетапним аналізом найбільш доцільних та доступних засобів для реалізації.

На сьогоднішній день враховуючи темпи впровадження найрізноманітніших пристроїв в наше життя, одним з найактуальніших методів впровадження ВПЗ є дана сфера. Такий висновок зроблено виходячи з тенденції розвитку технологій, враховуючи частку яка припадає саме на мобільні пристрої [2]. Про це говорить статистика представлена компанією Google, де вказано що кількість інформаційних запитів зроблених з мобільних пристроїв, значно більша ніж з ПЕОМ (статистика на основі 10 країн включно з США та Японією). Таким чином ми стикаємося з вибором між iOS, Android чи Windows Phone. В першу чергу слід зазначити, що для розробки додатків на тій чи іншій платформі бажано мати пристрій де можна буде провести процес відладки. iPhone завжди був одним з найдорожчих смартфонів на ринку. iPhone 5s коштує 250\$, не кажучи вже про 6S+ з цінником в 1100\$. З Windows Mobile виникають проблеми як в плані наявності ліцензії для розробки додатків на цій платформі, так і подальше просування свого товару на ринок. Масштабами і гнучкості у Android немає конкурентів. Мобільні пристрої на Android робить чи не кожен, хто пов'язаний з ринком смартфонів і тут завжди можна вибрати варіант від «дуже дешево» до «я не маю стільки грошей». А за величезну кількість безкоштовних додатків в PlayMarket Google отримує +5 до карми.

На Android все дуже просто: додаток можна завантажити безпосередньо

з онлайн-магазину, встановити з комп'ютера або скористатися одним із сторонніх магазинів яких існує велика кількість. А ось Apple і Windows навпаки відкрито виступають проти сторонніх магазинів. Тому якщо ви хочете мати простір для маневру, то чаша терезів схиляється в бік Android.

Недовге життя батареї - це головний біль і користувачів, і виробників. Вважається, що у iOS система енергоспоживання більш збалансована, однак Windows і Android виграють за рахунок більш містких акумуляторів, які зараз ставить кожен другий виробник.

Для кастомізації оболонки можна змінити вбудоване програмне забезпечення, встановити лаунчер, змінити екран блокування, тасувати віджети і змінювати розміри ярликів. За статистикою 85% всіх смартфонів, проданих у другому кварталі 2015 року, використовують ОС Android. Щодня в світі активується більше 1,3 млн пристроїв на базі ОС Android, і слід зазначити, що це не лише смартфони, а найрізноманітніші пристрої на кшталт цифрових годинників, білбордів чи навіть “розумного будинку”. Визначившись із платформою постає питання з вибором середовища програмування. В цьому питанні це скоріше вимога часу. Чому ж саме Android Studio:

1. Gradle. Надзвичайно потужний інструмент для збору проекту. Найбанальніший приклад використання - це під'єднання бібліотеки прописуванням одного рядка в файлі build.gradle на рівні додатку. До переваг системи слід віднести швидкодію програми яка зростає з кожною новою версією. Постійне оновлення бібліотек та зміна окремих модулів забирає багато часу на пошук помилки в одному з файлів, гнучкість gradle-змінних дасть змогу уникнути такої ситуації. У розроблюваних нами додатках ми часто використовуємо сторонні бібліотеки, що містять різні внутрішні ресурси, які в наших додатках абсолютно не потрібні. Завдяки Android Gradle Plugin ми можемо встановити мови і розширення, які використовуються в додатку, інші будуть вилучені, що дасть змогу зменшити розміри файлів[1].

2. Вбудований SDK. Існує багато різних IDE з уже вбудованим Android SDK, але додані вони туди не так тісно як в студії. Наприклад вам потрібно запустити старий проект, з низьким API level, який за непотрібністю ви просто не завантажили, половина класів заливаються фарбою, кількість помилок в проекті невблаганно росте і ви розумієте, що чогось тут не вистачає, але чого? Якої версії API немає? - потрібно розбиратися. Подібної ситуації у разі використанні студії не виникає. У разі використання будь-якого елементу, якого у вас немає, вона сама визначити де і чого не вистачає і викине віконце з повідомленням. Більш того, всі необхідні елементи можуть бути встановлено за лічені секунди.

3. Зручний інтерфейс. З першого погляду він не має ніякої відмінності від аналогічного в Eclipse, але понатискавши незнайомі кнопки, змінюючи відображення екрану і перегляді доданих нами елементів інтерфейсу, можна відразу зрозуміти, наскільки все стало краще. В одне натискання можна

переглянути відображення нашого екрану на будь-якому пристрої, аж до телевізорів і годинників. Самі елементи інтерфейсу відображаються саме так, як вони будуть виглядати у конкретній версії ОС, на відміну від інших IDE, де віджети відображаються на всіх версіях у вигляді однієї стандартної картини.

4. Зручний дизайн. В решті-решт це стара добра IDEA, яку знають як свої п'ять пальців добра частина Java розробників у всьому світі [4]. Дуже велику частину простору займають вікна повідомлень, властивостей елементів і самого проекту, але для кожного вікна передбачена функція згортання в одну з частин інтерфейсу. Більш немає потреби, як це було в Eclipse, закривати вікна за непотрібністю, всі ваші вікна будуть знаходитися в одному місці на панелі задач.

Наступним кроком буде вибір операційної системи на якій власне кажучи і буде відбуватись весь процес. З одного боку це швидше питання вподобань смаків чи навіть просто звички, те з чим найчастіше доводилось працювати користувачеві. Але якщо більш детально заглибитись в це питання, то можна побачити очевидні переваги Ubuntu.

Почнемо з того, що практично всі написані для Linux програми абсолютно безкоштовні. Так історично склалося, що програми під Linux поширюються з відкритим вихідним кодом. Це дає змогу запускати програму в будь-якій системі. Важко вимагати гроші за те, що може трохи підправити і поширювати безкоштовно будь-який студент-програміст.

У світі Linux майже не можливо підхопити якийсь вірус. З великою точністю можна вважати, що їх взагалі немає.

В порівнянні з Mac OS Ubuntu виграє за рахунок того, що працює на будь-якому комп'ютері та з будь-якою конфігурацією обладнання. До плюсів можна віднести прості автоматичні оновлення, можливість вибору, і інтеграцію програмного забезпечення незалежно від його вихідного коду, і напевно найголовнішим аргументом є ціна, де Ubuntu немає рівних за доступністю. Стосовно Microsoft Windows можна сказати, що Linux працює швидше. По-перше, раз немає вірусів, отже пересічному користувачеві не доводиться встановлювати антивіруси і міжмережеві екрани. І ті, і інші існують в світі Linux, але використовуються переважно на серверах. По-друге, оскільки програми поширюються у вигляді вихідного коду, то під час компіляції можна провести оптимізацію під конкретну систему. Згодом операційна система не буде гальмувати. Програми не вимагають перезавантаження системи після встановлення, оновлення або видалення. Оскільки за ПЗ платити не треба, то розробники не женуться за випуском нових релізів. А отже ніхто не женеться за прикрашанням і не потрібним функціоналом, займаючись натомість оптимізацією програм. Як наслідок, Linux добре працює на старому залізі, а також смартфонах і нетбуках; роздолья для програміста - всі протоколи і бібліотеки прекрасно задокументовані, можна заглянути в код будь-якої програми і дізнатися, як

вона працює. Емуляція віртуальних машин не працює на процесорах від AMD в операційній системі Microsoft Windows, тому що Microsoft любить інтереси Intel. На офіційному сайті Android Developers сказано, що скористатися емулятором Android на AMD можна тільки в Unix-сумісній системі. ARM-образи працюють занадто повільно для нормального їх використання. [5]

Переходячи до останнього і найбільш складного етапу, слід зазначити що в сучасному світі є досить велика кількість мов програмування, що спочатку ставить перед складною альтернативою - яку ж мову обрати? Звісно можна скористатись різними рейтингами, статистикою, або не задумуватись над вибором та стати на бік більшості. Але слід дивитись більш прагматично та звернути увагу на нові мови, які розвиваються, що в свою чергу свідчить про те що вони враховують всі недоліки та переваги своїх попередників. Такий чином вибір зупиняється на такій мові як Kotlin.

Дві головні особливості Kotlin, це її простота і повна сумісність з Java. Мова Kotlin створювалась компанією, яка робить дуже багато продуктів мовою Java і яка добре розбирається в сучасних інструментах розробки.

Сумісність з Java. Платформа Java - це перш за все екосистема: крім «офіційних» продуктів компанії Oracle, в неї входить безліч проектів з відкритим кодом: бібліотек і програмних каркасів різного профілю, на базі яких будується величезна кількість додатків. Тому для мови, яка компілює для цієї платформи, дуже важлива сумісність з наявним кодом, який написаний мовою Java. При цьому необхідно, щоб наявні проекти могли переходити на нову мову поступово, тобто не тільки код мовою Kotlin повинен легко викликати код мовою Java, але і навпаки.

Статичні гарантії коректності. Під час компіляції коду з статично типізованою мовою відбувається безліч перевірок, покликаних гарантувати, що ті чи інші помилки не відбудуться під час виконання. Наприклад, компілятор Java гарантує, що об'єкти, на яких викликаються ті або інші методи, «вміють» їх виконувати, тобто що у відповідних класах ці методи реалізовані. На жаль, крім цієї дуже важливої властивості, Java майже нічого не гарантує. Це означає, що успішно скомпільовані програми завершуються з помилками часу виконання (викликають виняткові ситуації). Яскравим прикладом є розіменування нульового посилання, при якому під час виконання викликається виняток типу `NullPointerException`. Важливою вимогою до нової мови є посилення статичних гарантій. Це дасть змогу виявляти більше помилок на етапі компіляції і, таким чином, скорочувати витрати на тестування.

Швидкість компіляції. Статичні перевірки спрощують програмування, але уповільнюють компіляцію, і тут необхідно домогтися певного балансу. Досвід створення мов з потужною системою типів (найбільш яскравим прикладом є Scala) показує, що такий баланс знайти непросто: компіляція часто стає неприйнятно довгою. Взагалі, така характеристика мови, як час

компіляції проекту, може здатися другорядною, однак в умовах промислової розробки, коли обсяги компільованого коду дуже великі, виявляється, що цей фактор дуже важливий - адже поки код компілюється, програміст часто не може продовжувати роботу. Зокрема, швидка компіляція є одним з важливих переваг Java в порівнянні з C++ і Kotlin повинен цю перевагу зберегти.

Лаконічність. Відомо, що програмісти часто витрачають більше часу на читання коду, ніж на його написання, тому важливо, щоб конструкції, доступні в мові програмування, давали змогу писати програми коротко і зрозуміло. Java вважається багатослівною, і завдання Kotlin - поліпшити ситуацію в цьому сенсі. На жаль, суворі методи оцінювання мов з точки зору їх лаконічності розвинені досить слабо, але є непрямі критерії; один з них - можливість створення бібліотек, робота з якими близька до використання предметно-орієнтованих мов (Domain-Specific Language, DSL). Для створення таких бібліотек необхідна певна гнучкість синтаксису в поєднанні з конструкціями вищих порядків; найбільш поширені функції вищих порядків, тобто функції, які беруть інші функції в якості параметрів.

Доступність для вивчення. Складні статичні перевірки, гнучкий синтаксис і конструкції вищих порядків ускладнюють мову і утруднюють її вивчення, тому необхідно до певної міри обмежувати набір підтримуваних можливостей, щоб мова була доступна для вивчення. Під час розробки Kotlin враховувався досвід Scala і інших сучасних мов, і занадто складні концепції в мову не вносились.

Інструментальна підтримка. Сучасні програмісти активно використовують різні автоматизовані інструменти, центральне місце серед яких займають інтегровані середовища розробки (Integrated Development Environment, IDE). Десятирічний досвід, накопичений в компанії JetBrains, показує, що певні властивості мови можуть істотно ускладнювати інструментальну підтримку. Під час розробки Kotlin враховано цей факт і створено IDE одночасно з компілятором.[3]

Отже, актуальність цього дослідження визначається потребою спрямування користувача в сфері розвитку вільного програмного забезпечення, враховуючи альтернативність різних методів розробки. В доповіді наведено причини вибору кожного сегменту окремо, їх взаємодію, та продуктивність роботи системи в цілому. Таким чином, кожен, хто усвідомлює перспективність роботи в цій сфері, може стати Android-розробником. В сучасному світі для цього не потрібні дорогі ліцензії чи спеціальне обладнання.

Джерела

1. Gradle: 5 полезностей для разработчика [Електронний ресурс] : Блог компании REDMADROBOT / Хабрахабр. — Режим доступу до блогу: <https://habrahabr.ru/company/redmadrobot/blog/271269/>
2. Bosomworth D. Mobile Marketing Statistics compilation [Електронний ресурс] : Smart insights. — Режим доступу до сайту: <http://www.smartinsights.com/mobile->

marketing/mobile-marketing-analytics/.

3. Leiva A. Kotlin for Android Developers [Електронний ресурс] :Leanpub: Publish Early, Publish Often. — Режим доступу до сайту: <https://leanpub.com/kotlin-for-android-developers>

4. ТЮБЕ Index for March 2016 [Електронний ресурс] : Tiobe - The Software Quality Company. — Режим доступу до сайту: http://www.tiobe.com/tiobe_index?page=index

5. Using the Emulator [Електронний ресурс] :Android Developers. — <http://developer.android.com/intl/ru/tools/devices/emulator.html#acceleration>

Розробка програмного забезпечення для оцінки та зміни параметрів якості растрових зображень ***Яворівський Б.***

Національний університет «Львівська політехніка», pnelconn@gmail.com

The developed software based on open systems and software is available for operating systems Linux and Windows. It is planned to develop for Android. The application uses assesses the quality bitmaps quantitative indicators. Quality of image is formed on the basis of the calculation and adjustment of brightness, contrast, contrast, tone, contrast, saturation, brightness and tone. Experimental results are shown as histograms. A convolution method to change the color characteristics of the pixels is presented.

Зорова система людини - найбільш надійний і досконалий вимірювальний інструмент для оцінки якості цифрового зображення. Проте суб'єктивна оцінка растрових зображень є досить складною і повільною, для здійснення вимагає залучення досвідчених експертів. Тому розробка програмного забезпечення для оцінки якості та редагування растрових зображень є актуальним завданням. Даний ужиток може бути використане у різних сферах життєдіяльності людини, де необхідно покращувати якість цифрових зображень.

Можливі два підходи до оцінки якості зображень: кількісна оцінка, яка базується на основі використання математичних і суб'єктивна оцінка на основі експертних оцінок [1].

Суб'єктивні та кількісні оцінки якості зображень можуть бути абсолютними або порівняльними. Абсолютна міра якості використовується для оцінки одного зображення, тобто зображенню присвоюється відповідна категорія в рейтинговій шкалі. Порівняльні заходи використовуються для ранжування набору зображень в якісній шкалі від «найкраще» до «найгірше» або взаємного порівняння двох зображень, наприклад, вихідного і відфільтрованого (чи отриманого у різні дні, різними камерами і т.д.).

Міру різкості зображення можна визначити шляхом знаходження кута нахилу профілю яскравості зображення на кордоні перепаду. Для оцінки контрасту зображення здійснюють порівняння пікселів, виходячи з окремих комбінацій елементів зображення. При цьому всі елементи вважаються

рівнозначними. Застосовуючи правило підсумовування контрастів, обчислюють набір величин, які визначають сприйняття кожної пари елементів зображення. Проводячи усереднення матриці локальних контрастів, отримують сумарний контраст [2].



Рис. 1. Зображення до зміни параметрів якості Рис. 2. Зображення після зміни параметрів якості ужитком

Контрастність яскравості - це різниця між фізичною або видимою яскравістю окремих ділянок зображення. Обчислення фізичної або видимої яскравості можна розглядати як конвертацію кольорового зображення в ахроматичні кольори. Тому контрастність яскравості- це порівняння двох ділянок зображення , приведених до ахроматичних кольорів.

Якщо проаналізувати RGB-гістограми , то можна зробити висновок, що у контрастного зображення кількість темних і світлих пікселів має бути приблизно однакова, різниця в їх яскравості - значна, а основне місце зосередження пікселів - біля кордонів діапазону.

Показник тонової контрастності належить до складніших показників якості. Конвертовані у відтінки сірого кольору можуть мати однакову яскравість, але візуально чітко розрізнятися. Тонові насиченість- це відмінність кольору від ахроматичного при його однаковій яскравості. У RGB-кубі тонову насиченість пікселя можна виразити як відстань до діагоналі ахроматичних кольорів . Для всього зображення оцінка тонової насиченості може бути виражена як середнє значення тонової насиченості для всіх пікселів. Яскравість зображення можна виразити як середню яскравість усіх пікселів.

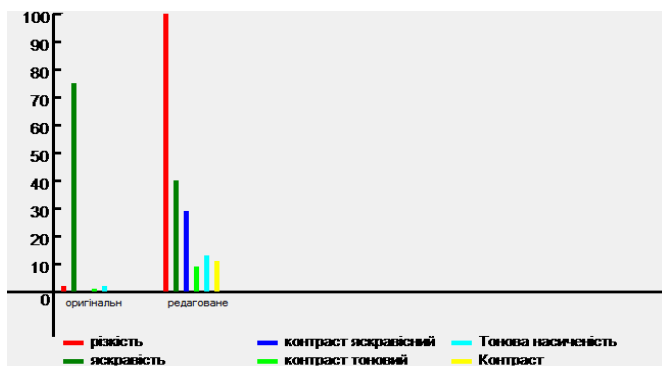


Рис. 3. Гістограми тестів зображень 1 та 2.

Усі описані критерії запрограмовано у створеному програмному забезпеченні для оцінки якості зображень. Ужиток розроблено за допомогою Qt Creator мовою C++. Опишемо роботу ужитку на прикладі зображення, зображеного на рис.1. Рисунок 1 і 2 представляють оригінальне і редаговане зображення відповідно. Тестування зображення було проведено до і після застосування інструментів зміни параметрів якості реалізованих у програмі. Після застосування ужитку показники різкості, яскравості і контрасту були покращені, що підтверджується гістограмою (див. рис. 3). На даній гістограмі відображаються ті параметри оцінки зображення, які можна визначити у створеній програмі.

Оскільки чутливість людського зору до різних частин спектра неоднакова (максимальна в жовто-зеленій, менша в червоній, ще менша у синій) [3], то яскравість кольорового пікселя буде сприйматися суб'єктивно в залежності від його тональних характеристик.

Висновок

Розроблене програмне забезпечення для оцінки якості зображення, яке ґрунтується на кількісних та якісних показниках. Показники якості зображення сформовано на основі обчислення та корекції яскравості, контрасту, контрасту тонового, контрасту яскравості та тонової насиченості. Приведені результати експериментів у вигляді гістограм та таблиць.

Джерела

- [1] Jain A.K. Fundamentals of Digital Image Processing. – Prentice-Hall, Inc., USA, 1989.
- [2] Wang X., Tian B., Liang C., Shi D. Blind Image Quality Assessment for Measuring Image Blur // Congress on Image and Signal 2008 Congress on Image and Signal Processing, 2008.
- [3] Лотошинська Н.Д., Івахів О.В. Теорія кольору та кольороутворення // Навч. посібник. Львів: Видавництво Львівської політехніки, 2014. 204 с.

Вольная альтэрнатыва Dropbox і сэрвісам GOOGLE

Захарэвіч А.Л.

EPAM Systems, Мінск, Беларусь, andrej@zahar.ws

The experience of an attempt to get some open source and provider independent solutions is presented for some tasks user usually outsources to other companies and corporations like Dropbox Inc or Google Inc.

Аўтар вырашаў задачу па замене вольнымі аналагамі агульнаўжытных прапрыетарных сэрвісаў, якія рэалізуюць наступны функцыянал:

- Файлаабменнік – магчымасць падзяліцца спасылкай на файл з механізмам абмежавання доступа, стварыць агульны набор рэсурсаў для некалькіх карыстальнікаў, абнаўленне файлаў з адной крыніцы на ўсе месцы, дзе ляжыць яго копія, кліент пад Linux для сінхранізацыі ўсяго акаўнта ці асобных яго папак;
- Кантакты і календар – цэнтралізаванае сховішча для календара і кантактаў з магчымасцю сінхранізацыі на прылады Android, зручны (хаця б мінімальна) інтэрфейс для камп'ютэра з магчымасцю аб'ядноўваць кантакты ў групы, бачныя і даступныя для любых аперацый як з камп'ютэра, так і з тэлефона, магчымасць імпартаваць кантакты і календар ў Thunderbird, экспарту ў файл і пераносу на іншы сервер;
- RSS – магчымасць чытаць і дадаваць як з камп'ютэра, так і з Android-прылад, дадаваць з камп'ютэра і андроід-прылад, сінхранізаваць стан (прачытаны, пазначаны як цікавы) для розных крыніц, магчымасць экспарту ў файл і пераносу на іншы сервер, доступ праз вэб-інтэрфейс;
- Спасылкі – магчымасць чытаць і дадаваць як з камп'ютэра, так і з Android-прылад, групіроўка па групах ці тэгах, даступная для ўсіх кліентаў, магчымасць экспарту ў файл і пераносу на іншы сервер.

Ніжэй разглядаецца атрыманы аўтарам вопыт дасягнення гэтых мэтаў на базе прыватнага воблака ownCloud.

Асаблівасці наладкі

Самы прасты варыянт атрымання інстансу ownCloud – гэта паставіць яго, карыстаючыся парадкамі з афіцыйнага кіраўніцтва да бягучай стабільнай версіі [1]. Альбо можна узяць гатовы пакет да большасці папулярных дыстрыбутываў, ці нават прапісаць адпаведную крыніцу для інсталяцыі пакетаў (гэта дазваляе даволі зручна абнаўляцца). Калі плануецца карыстацца вэб-інтэрфейсам каб дадаваць файлы ў воблака, то варта яшчэ выканаць дадатковыя настройкі для гэтага [2].

Пры настройцы базы дадзеных [1, 3] трэба ўлічваць, што на серверы з не

надта вялікім аб'ёмам памяці кэш можа не паскараць працу, а замаруджваць яе, насуперак парадам з афіцыйнага кіраўніцтва [4]. А неправильная настройка РНР-кэшавання можа зрабіць вашае воблака бясконца бяспечным (то бок, недаступным нікому).

У выпадку з RSS калі працэс не здолее атрымаць абнаўленні за пэўны час і перастане працаваць не знішчыўшы lock-файл, то абнаўленне будзе немагчымае без таго, каб гэты файл знішчыць. Гэта магчыма зрабіць рукамі, альбо натравіць асобны скрыпт, які будзе па раскладзе маніторыць стан працэсу.

Што сапраўды варта зрабіць, гэта настройка бяспекі. Зноўку ж, варта звярнуцца да адпаведнага раздзелу афіцыйнага кіраўніцтва [5] – адключыць доступ не па HTTPS і выканаць іншыя парады. З дапамогай гэтых мер аўтар змог падняць сайт па рэйтынгу SSL Server Test [6] ад T(C) да T(A).

Перанос дадзеных ва ўласнае воблака

Прасцей за ўсё выконваецца перанос файлаў: яго можна пачаць адразу ж, выкарыстоўваючы вэб-інтэрфейс, дэсктопны [7] ці мабільны [8] кліент. Дэсктопныя кліенты з нядаўніх часоў дазваляюць сінхранізацыю з некалькімі акаўнтамі.

Перанос кантактаў прыйшлося выканаць шляхам экспарту з Google Contacts і паўторным імпартам ужо ў ownCloud. Калі ў кантактах ёсць кірыліца і іншы юнікод па-за межамі ASCII, ёсць сэнс выбіраць фармат vCard, інакш будуць праблемы з кадзіроўкаю. Акрамя таго, выкарыстанне знакаў пунктуацыі у імёнах груп кантактаў недапушчальнае – у выніку пасля сінхранізацыі імёны будуць ламацца (аўтару давалося міграваць групу “Сям’я” у групу “Сваякі”).

Вывявы для кантактаў пры гэтым падыходзе будуць страчаны, але ўсе групы захавваюцца (кантакты фактычна не належаць да нейкай групы, яны проста маюць адпаведны тэг).

Пасля пераноса лепш за ўсё настроіць двухбаковую сінхранізацыю з Android-смартфонам ці планшэтам. Пачынаючы з версіі 4.0 Андройд дазваляе дадаваць правайдэры кантактаў і календара. Сярод розных праграм DAVdroid – CalDAV/CardDAV Sync [9] паказаў сябе лепш за ўсё: сінхранізацыя працуе хутка і надзейна, і ён дазваляе сінхранізаваць адразу кантакты і календар.

Сітуацыя з імпартам календара крыху больш складаная: аўтар не знайшоў спосабу штатным чынам экспартаваць яго з гуглаўскага календара і скарыстаўся метадам з блога Map and Keyboard [10]:

1. Выдаліць непатрэбныя падзеі;
2. Настроіць двухбаковую сінхранізацыю календара паміж Android-прыладай і ownCloud;
3. Скрыстацца праграмай iCal Import/Export CalDAV [11] і перанесці сінхранізаваць календар паміж аб'ектамі.

Таксама даступная і сінхранізацыя з дэсктопам [12, 13].

Абрана́я RSS-чы́талка [14] дазваляе зручна чытаць і сінхранізаваць. Нажаль, у сэнсе кіравання падпіскамі па-за межамі дадаць новую крыніцу лепш карыстацца веб-інтэрфейсам. Але, з іншага боку, карыстацца веб-інтэрфейсам для чытання не так зручна.

Карыстацца спасылкамі праз веб-інтэрфейс досыць зручна, і ёсць неблагі кліент Андройд [15] (які, нажаль, не дазваляе дадаваць групы для спасылак і рэдагаваць спасылкі, але ва ўсім астатнім досыць зручны і невялікі).

Літаратура і спасылкі

1. ownCloud 9.0 Server Administration Manual/ Installation.
https://doc.owncloud.org/server/8.2/admin_manual/installation/index.html
2. ownCloud Server Administration Manual: Uploading big files.
https://doc.owncloud.org/server/8.2/admin_manual/configuration_files/big_file_upload_configuration.html
3. ownCloud Server Administration Manual: Database Configuration.
https://doc.owncloud.org/server/8.2/admin_manual/configuration_database/linux_database_configuration.html
4. ownCloud Server Administration Manual: Server Tuning & Performance Tips.
https://doc.owncloud.org/server/8.2/admin_manual/configuration_server/performance_tuning.html
5. ownCloud Server Administration Manual: Hardening and Security Guidance.
https://doc.owncloud.org/server/8.2/admin_manual/configuration_server/harden_server.html?highlight=security
6. Qualys SSL Labs SSL Server Test. <https://www.ssllabs.com/ssltest/>
7. ownCloud Desktop Clients. <https://owncloud.com/products/desktop-clients/>
8. ownCloud client for Android. <https://play.google.com/store/apps/details?id=com.owncloud.android>
9. DAVdroid – CalDAV/CardDAV Sync. <https://play.google.com/store/apps/details?id=at.bitfire.davdroid>
10. Man and Keyboard: Google to Owncloud, Contacts and Calendar.
<http://manandkeyboard.tk/2015/02/26/google-to-owncloud-contacts-and-calendar/>
11. iCal Import/Export CalDAV. <https://play.google.com/store/apps/details?id=tk.drlue.icalimportexport>
12. Moving your Contacts and Calendar Away from Google.
<http://flailingmonkey.com/moving-contacts-calendar-google/>
13. Birthday Calendar with ownCloud via CalDAV/ <http://blog.mehl.mx/2014/birthday-calendar-with-owncloud-via-caldav/>
14. ownCloud News Reader. <https://play.google.com/store/apps/details?id=de.luhmer.owncloudnewsreader>
15. ownCloud Bookmarks. <https://play.google.com/store/apps/details?id=cz.nethar.owncloudbookmarks>

Програмне забезпечення ІТ-компанії та фахова підготовка студентів напрямку “Комп’ютерні науки” на факультетах електроніки та прикладної математики ЛНУ ім. Івана Франка
Злобін Г.Г., Скоропад О., Рикалюк Р.

ЛНУ ім. Івана Франка, Львів, вул. Тарнавського 107, zlobingg@gmail.com; EPAM-Львів, sko@ukr.net; ЛНУ ім. Івана Франка, Львів, вул. Університетська 1, r.rykaluk@gmail.com

Software development company is a factory for the production of IT products. Like any factory, this company has complex structure and big set of processes with a lot of the necessary tools. All these tools are special applications for software development. This article describes a simple software landscape of modern IT company and explains major functionality of components. Particular attention is paid to free software.

Сучасна ІТ компанія – це фабрика з випуску програмних продуктів. Як і кожна фабрика, така компанія має свою структуру та налагоджений складний виробничий процес з великою кількістю необхідних інструментів. В ролі інструментів виступають програмні продукти призначені для розробки програмного забезпечення на всіх його етапах, а оскільки таких етапів є багато, то і перелік продуктів є дуже широким.

Спробуємо класифікувати програмне забезпечення для розробки:

1. Інструменти програмної архітектури, документування та аналітики призначені для візуалізації ідеї програмного проекту та візуалізації блок-схеми та основних компонентів майбутнього ПЗ;
2. Інструменти розробника (редактори, бази знань та інструменти відслідковування помилок /bug-track tool/);
3. Засоби контролю коду (code review);
4. Система контролю версій;
5. Центральний репозитарій програмного коду;
6. Програмні засоби безперервної інтеграції;
7. Програмні засоби автоматичного та ручного тестування;
8. Засоби доставки продукту до користувача, а також засоби комунікації з клієнтом.

Окремої уваги вимагає системне програмне забезпечення для підтримки роботи цієї інфраструктури, яке проте не входить в цей огляд:

1. Засоби віртуалізації;
2. Програмне забезпечення автоматизованого конфігурування;
3. Засоби розгортання програмних середовищ (orchestration tool) та інструменти управління контейнерами;
4. Засоби моніторингу;
5. Програмне забезпечення для резервного копіювання.

Це основний, але далеко не повний перелік типів продуктів, який розширюється настільки швидко, наскільки відбувається все вужча спеціалізація в індустрії програмного забезпечення.

Варто зупинитись на природі перерахованих продуктів. Якщо раніше і розробники програмного забезпечення, і його користувачі віддавали перевагу дорогим брендовим продуктам відомих компаній за принципом дорожче – краще, то зараз ситуація на IT-ринку кардинально змінилась: перевага надається безоплатно поширюваному, умовно-платному, а найбільше – відкритому програмному забезпеченню. Така переорієнтація ринку відбулась з наступних причин:

1. Код відкритого ПЗ вільний для модифікації чим забезпечується максимальна гнучкість проєктів;
2. Відкрите програмне забезпечення розвивається, по суті під наглядом спільноти, тому суттєво знижуються ризики пов'язані з неякісним кодом;
3. Повністю ліквідуються загрози, пов'язані з припиненням розробником супроводу програмного продукту, оскільки супровід відбувається консолідованими зусиллями самих користувачів;
4. Вартість проєктів з використанням відкритого ПЗ на порядок нижча за вартість таких же проєктів з використанням комерційного ПЗ.

Всі ці фактори приводять до того, що не лише користувачі, а й дедалі більше розробників починають переходити на відкрите ПЗ. Великі компанії з розробки ПЗ ефективно використовують практику тестування нових версій продуктів в якості відкритого ПЗ, а лише після цього ліцензують код найбільш стабільних версій. Яскравим прикладом такого методу є тандем CentOS – RedHat.

Проведемо короткий огляд основних спеціальних програмних продуктів, що широко використовуються компаніями з розробки ПЗ в промисловій експлуатації:

1. Засоби контролю коду – це програмне забезпечення, яке призначене для первинної перевірки програмного коду. Інша важлива функція цього ПЗ полягає в передачі знань від однієї людини, що пише код до інших спеціалістів, які в разі потреби зможуть продовжити написання програми. Вказане ПЗ часто вбудовується в системи контролю версій. Популярні пакети: `gerrit`, `Barkeeper`, `Phabricator`;

2. Системи контролю версій та репозитарій – це централізоване сховище коду з організованою структуризацією версій ПЗ. Є головною базою всіх напрацьованих продуктів, дає можливість повернення в будь-який момент до будь-якої зафіксованої версії ПЗ, а також дає змогу створити незалежну гілку написання програмного продукту з довільної версії поточного продукту. Найвідоміші продукти: `svn`, `git`, `nexus`;

3. Системи безперервної інтеграції (CI - continuous integration) з'явилися відносно недавно у зв'язку з переходом на нові технології розробки ПЗ – Agile та Scrum. Це ПЗ дає змогу повністю автоматизувати найбільш рутинні операції розробника, якими є : завантаження залежностей та бібліотек, компіляція, блок-тестування, публікація результатів. Системи CI практично завжди працюють в тандемі з системою контролю версій та системами тестування, що дає змогу, наприклад, здійснювати автоматичне повернення до попередньої версії проєктованого ПЗ за наявності помилок в його коді. Приклади пакетів: jenkins, metacity (комерційне ПЗ), RHPCI;

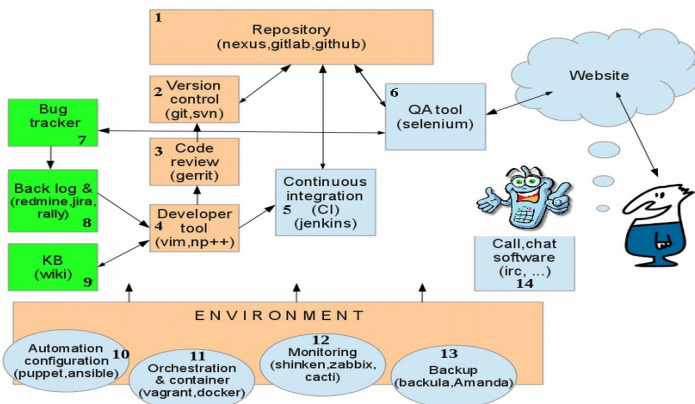


Рис. 1 1 – репозиторії (nexus, github, gitlab тощо), 2 – система контролю версій (git, svn тощо), 3 – система аналізу коду (gerrit), 4 – спеціалізовані текстові редактори із підсвіткою синтаксису (vim, np++ тощо), 5 – система безперервної інтеграції (jenkins, hudson, metacity, RHPCI тощо), 6 – засоби тестування (Selenium IDE, Sikuli, WebDriver, PHPUnit, Амос тощо), 7 – система відслідковування помилок (BUGS, Bugzilla, eTraxis, GNATS тощо), 8 – система управління проектами і задачами (readmine, jira, rally, GitLab тощо), 9 – вікі-система для створення єдиної бази знань організації (Wiki Media, Confluence), 10 – система автоматичного конфігурування операційних систем і програм (puppet, ansible тощо), 11 – система управління віртуальними машинами (docker, vagrant тощо), 12 – система моніторингу служб і станів (Nagios, monit, shinken, zabbix, cacti тощо), 13 – системи управління резервним копіюванням (backula, amanda, ar+gzip, pgdump, mysqldump, windows backup тощо), 14 – засоби Інтернет-спілкування розробників програм (irc, Jabber/GoogleTalk, HipChat, Skype тощо)

4. Засоби тестування – широка лінійка програмних продуктів, що працює в галузі забезпечення надійності програмного коду (QA – quality assurance).

Вказане ПЗ призначене для контролю якості робочих версій коду і дає змогу відсіяти у разі хорошої організації процесу до 90% помилок в продуктах. Галузь QA інтенсивно розвивається, що призводить до появи великої кількості нового ПЗ для тестування. На сьогодні одних лише типів тестів нараховується близько десятка і для кожного з них існує своя лінійка тестового ПЗ. Найвідоміші пакети: Selenium IDE, Sikuli, WebDriver, PHPUnit, Атос.

5. Інструменти для документування коду, що полегшують подальшу підтримку та передачу ПЗ іншим командам. Ці інструменти це не звичайні коментарі поміж рядками, а структуровані коментарі, що пізніше слугують для генерування Web- та Pdf-документів із зручними для читання блок-схемами та списками класів та методів (DoxyGen, Swagger, Dexus);

6. Також окрім самих інструментів важливими є правила та методи роботи в команді. До методів можна віднести правила комунікації з використанням електронної пошти, Skype тощо;

7. Методології ведення проєктів – Agile, Scrum, WaterFlow та їх модифікації.

Як вже було зазначено, одною з найбільших переваг відкритого програмного забезпечення є широка доступність документації та навчальних матеріалів, що дає змогу ефективно та максимально швидко готувати персонал для роботи з ним. Процес підготовки проходить у вигляді так званих техконференцій у максимально вільному форматі, де встановлюється лише загальна тема та перелік властивостей програмного забезпечення, що обов'язкові до розгляду. Такий вид навчання нагадує обмін знаннями в професійних спільнотах і дуже добре зарекомендував себе.

На рис. 1 зображена блок-схема базового ПЗ, що широко використовується розробниками програмного забезпечення.

Для використання базового ПЗ, зображеного на рис. 1, у професійній діяльності бажано забезпечити його вивчення під час навчання у ВЗО або на підготовчих курсах в ІТ-компанії. Результати вступних співбесід на посаду системного інженера (Junior System Engineer) зазвичай показують високий рівень фундаментальної (теоретичної) підготовки випускників Львівського національного університету імені Івана Франка і водночас дуже слабкі знання промислових методів та інструментів розробки ПЗ. Спробуємо проаналізувати причини такого стану на прикладі двох факультетів Львівського національного університету імені Івана Франка - факультету електроніки та факультету прикладної математики та інформатики. З 2013 р. У навчальних лабораторіях електроніки факультету встановлено лише ліцензійне програмне забезпечення - у чотирьох лабораторіях ОС Linux та прикладне програмне забезпечення для неї, у трьох ОС Linux та ОС Microsoft Windows 7 за програмою Dream Spark. Причиною такого становища став провал платної факультетської підписки на Dream Spark через помилки

в апікаційних формах, зроблених представником факультету електроніки, та пізніше оформлення безоплатної кафедральної підписки на Dream Spark на кафедрі радіофізики та комп'ютерних технологій. У навчальних лабораторіях факультету прикладної математики та інформатики встановлено ОС Microsoft Windows 7 за програмою Dream Spark та програмне забезпечення для неї. В деяких лабораторіях встановлено також ОС Linux для забезпечення навчальних курсів, які торкаються безпосередньо цієї операційної системи.

На факультеті електроніки ЛНУ ім. Івана Франка читаються лекційні курси, які пов'язані з розробкою та використанням програмного забезпечення [1]:

- I. Алгоритмізація та програмування, перший семестр;
- II. Комп'ютерна графіка, перший семестр;
- III. Об'єктно-орієнтоване програмування, другий семестр;
- IV. Кросплатформне програмування, третій семестр;
- V. Технології комп'ютерного проектування, четвертий семестр;
- VI. Організація баз даних і знань, четвертий семестр;
- VII. Операційні системи, четвертий семестр;
- VIII. Web-технології та Web-дизайн;
- IX. Технології створення програмних продуктів, шостий семестр;
- X. Проектування інформаційних систем, шостий семестр;
- XI. Управління IT-проектами, восьмий семестр.

Незважаючи на те, що в шести комп'ютерних лабораторіях факультету електроніки встановлені лише ліцензійні ОС (в чотирьох лабораторіях лише Linux, в трьох лабораторіях Linux і Microsoft Windows 7) та ліцензійно чисте ПЗ, рівень використання базового ПЗ є поки що недостатнім. Наразі лише в наступних курсах спостерігається використання базового ПЗ

Таблиця 1.

Лекційні курси, семестр/Інструменти	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Операційні системи, 4	-	-	-	-	-	-	-	-	-	+	-	+	+	-
Технології створення програмних продуктів, 6	+	+	-	-	+	-	-	+	-	-	-	-	-	-
Управління IT-проектами, 8	-	-	-	-	-	-	-	*	-	-	-	-	-	-

Умовні позначення: + використовується у лекційному курсі, - не використовується у лекційному курсі, * лектор використовує Open Proejct тому, що Open Proejct подібний до Microsoft Proejct.

На факультеті прикладної математики та інформатики ЛНУ ім. Івана Франка читаються лекційні курси, які пов'язані з розробкою та використанням програмного забезпечення [2]:

- I. Програмування, перший-третій семестр;
- II. Організація та обробка електронної інформації, другий семестр;
- III. Програмне забезпечення, четвертий семестр;
- IV. Бази даних та інформаційні системи, п'ятий семестр;
- V. Паралельні та розподілені обчислення, п'ятий семестр;
- VI. Платформи корпоративних інформаційних систем, п'ятий семестр;
- VII. Програмування та підтримка веб-застосувань, п'ятий семестр;
- VIII. Бази даних та інформаційні системи, шостий семестр;
- IX. Обробка зображень та мультимедіа, шостий семестр;
- X. Проектування програмних систем, шостий семестр;
- XI. Програмування мовою Java, шостий семестр;
- XII. Операційні системи та системне програмування, сьомий семестр;
- XIII. Теорія програмування, сьомий семестр;
- XIV. Нечітке моделювання в середовищі Matlab, сьомий семестр;
- XV. Логічне та функціональне програмування, сьомий семестр;
- XVI. Операційні системи та системне програмування, восьмий семестр;
- XVII. Проектування систем штучного інтелекту, восьмий семестр;
- XVIII. Розподілені інформаційно-аналітичні системи, восьмий семестр;
- XIX. Об'єктно-орієнтовна система Smalltalk, восьмий семестр;
- XX. Програмування під UNIX-подібними системами, восьмий семестр.

Домінування ОС Microsoft Windows в навчальних лабораторіях факультету прикладної математики та інформатики неявно призводить до домінування закритого прикладного програмного забезпечення часом навіть із порушенням ліцензійності цього програмного забезпечення. Рівень використання базового ПЗ як і на факультеті електроніки є недостатнім, що призводить до того, що студенти обох факультетів не отримують належних знань з промислових методів та інструментів розробки ПЗ. Це є основною причиною організації ІТ-компаніями як у себе, так і у вищих закладах освіти (курси SoftServe у Львівському національному університеті імені Івана Франка, курси Ерам у Львівській політехніці тощо). На жаль, таке становище призводить до втрати інтересу у частини студентів до вивчення дисциплін з навчального плану вищого закладу освіти та переоцінки важливості технологічних знань, які викладаються на підготовчих курсах.

На думку авторів доповіді потрібно:

- I. усунути розрив між навчанням та виробничою практикою в ІТ-компаніях шляхом максимально можливого впровадження базового ПЗ в навчальні курси, які викладаються у вищих закладах освіти;

II. зміцнити зв'язки вищих закладів освіти з виробництвом шляхом стажування викладачів напряму “Комп’ютерні науки” у провідних ІТ-компаніях на безоплатній основі;

III. відмовитись від стереотипу, який панує в головах багатьох викладачів вищих закладів освіти, про те, що пропрієтарне програмне забезпечення є найкращим та найнадійнішим.

Джерела

1. http://electronics.lnu.edu.ua/base_e/its/index.php?p=2/
2. <http://ami.lnu.edu.ua/academics/bachelor/curriculum-informatics>

ISBN 978-966-2598-63-6

Збірник наукових праць

**МАТЕРІАЛИ Шостої
НАУКОВО-ПРАКТИЧНОЇ
КОНФЕРЕНЦІЇ
FOSS LVIV 2016**

Під редакцією:

Злобін Г.Г., Апунович С.Є., Апунович С.В.

Підписано до друку 07.04.2016

Формат 60x84/16. Папір офсетний. Цифровий друк.

Гарнітура Liberation.

Фіз. друк. арк. 7,87 Умов. друк. арк. 7,3

Тираж 100.

Видавець:

Приватний підприємець Сорока Тарас Богданович

79026, м. Львів, вул. Володимира Великого, 2

Свідомство державного реєстру: серія ЛВ №17

soroka@soroka.lviv.ua