

## **Література**

1. Сбитнев Ю.И. Кластеры / Ю.И. Сбитнев. – Екатеринбург, 2009. – 119 с.
2. Высокопроизводительные вычисления на кластерах: Учебн. пособие/ Под ред. А.В. Старченко. – Томск: Изд-во Том. ун-та, 2008. – 198 с.
3. Воеводин В. В. Параллельные вычисления / В. В. Воеводин - СПб.: БХВ - Петербург, 2002.- 608 с.
4. Самарский А.А. Введение в численные методы /А.А.Самарский. – Спб.:Лань, 2009. – 288 с.
5. Ващенко Г.В. Параллельная реализация явных методов Рунге-Кутты / Г.В. Ващенко, Е.А. Новиков. - Вестник Красноярского государственного аграрного университета. – 2010.- № 2. – С.14-18.
6. Швачич Г.Г. О концепции неограниченного параллелизма в задачах теплопроводности / Г.Г. Швачич, А.А. Шмукин // Восточно-Европейский журнал передовых технологий. - 4 (10). - 2004. – с. 132-135.

### **Новий підхід до створення INITRAMFS**

*Сітало В.С., Ясько М.М.*

*Дніпропетровський національний університет імені Олеся Гончара, факультет прикладної математики, [root@stvad.org](mailto:root@stvad.org)*

Considered the principles of operation, uses and benefits of live systems. Proposed a new approach to the construction of the initial file system and described the benefits of this approach. Described the steps necessary to implement the approach. Approach demonstrated on the example of Puppy Linux.

Одним із засобів ширшого розповсюдження вільного програмного забезпечення, а конкретніше для ширшого ознайомлення користувачів з операційними системами GNU/Linux, є використання LiveCD. LiveCD – це операційна система, яка завантажується зі змінного носія (CD, DVD, USB-накопичувача і т.п.) та не вимагає для свого функціонування встановлення на жорсткий диск. Цим же поняттям позначають і носії з такими ОС (в залежності від носія розрізняють LiveCD, LiveDVD і LiveUSB).

LiveCD дозволяють швидко почати роботу з комп'ютером, уникаючи тривалого процесу установки ОС в сталу пам'ять (вінчестер). Запуск LiveCD звичайно триває декілька хвилин, в той час як перший запуск (встановлення) традиційних ОС часто вимагає від однієї до декількох годин.

Першим LiveCD, який базувався на Linux, був Yggdrasil Linux, вперше виданий у 1992р. На більш пізньому етапі розвитку з'явилась можливість

виконувати завантаження ОС також і з інших змінних носіїв. В останні десять років були створені сотні таких дистрибутивів, здебільшого на базі операційних систем Linux та BSD.

Найчастіше LiveCD використовуються для ознайомлення з можливостями операційних систем. Інше застосування LiveCD - відновлення даних та / або працездатності основної операційної системи після збою. LiveCD можна застосовувати на комп'ютерах, які не мають жорсткого диска і на комп'ютерах, що мають жорсткий диск дуже невеликого обсягу.

При створенні таких систем зазвичай використовується початкова файлова система в оперативній пам'яті (initrd або initramfs). Initrd пов'язана з ядром і завантажується як його частина в процесі завантаження системи. Ядро монтує образ initrd, в якому знаходяться необхідні модулі для монтування кореневої ФС і подальшого переходу в цей корінь як основний. В initrd міститься мінімальний набір директорій і виконуваних файлів для завантаження модулів. У випадку настільної системи або сервера, initrd — це тимчасова файлова система. Час її життя досить малий і слугує вона переважно лише проміжною ланкою до кореневої ФС.

В initramfs зазвичай розміщують модулі ядра, критичні для завантаження системи (драйвери дискового контролера, драйвери кореневої файлової системи, а при необхідності й модулі підтримки NFS, LVM і RAID) і спеціальний скрипт (init або linuxrc), який і запускається на ранніх стадіях завантаження. Одна з його основних функцій полягає в пошуку розділу або файлу, де заходиться основна файлова система. Саме він збирає софтові RAID-масиви, активує LVM і т.п. Потім, коли основна файлова система знайдена, він переключає корінь системи на неї (а на ранніх стадіях роль тимчасової кореневої системи грає саме initramfs). Далі запускається системний init, який і веде подальше завантаження системи. Швидкість завантаження таких систем, як правило, не дуже висока. Для створення initramfs створені певні інструменти, такі як mkinitrd або більш сучасний Dracut.

В даній роботі пропонується новий підхід до створення початкової файлової системи, який полягає в наступному:

*1. Розмістити основну файлову систему (у стисненому вигляді як squashfs) безпосередньо у initramfs. (Уперше ця ідея була запропонована професором Барі Каулером, основним розробником дистрибутива Purru Linux. Але авторам не відомі якісь її існуючі реалізації)*

*2. Початкову команду init, яка зазвичай являє собою скрипт, написати мовою C.*

Початкова команда init повинна виконувати такі дії:

*1. Завантаження необхідних модулів ядра. В більшості випадків таких модулів буде не більше трьох (fuse, squashfs, unionfs або aufs). Завантаження модуля може здійснюватись за допомогою наступної процедури:*

```
void LoadModule(const char *ModPath){
    int fd;
    void *image;
    struct stat st;
    fd=open(ModPath,O_RDONLY);
    fstat(fd, &st);
    image = mmap(NULL, st.st_size, PROT_READ, MAP_PRIVATE, fd, 0);
    close(fd);
    init_module(image, st.st_size, "");
}

```

2. Монтування основної файлової системи у форматі *squashfs* може здійснюватись наступним чином:

```
fdev=open(devname,O_RDWR);
lodev=open("/dev/loop0",O_RDWR);
ioctl(lodev, LOOP_SET_FD, fdev);
close(lodev);
mount("/dev/loop0", "/d_root", "squashfs", 0, NULL);

```

3. Створення тимчасової файлової системи у пам'яті та сумісне монтування двох файлових систем за допомогою *unionfs* можна реалізувати наступним чином:

```
mount("none", "/d_rw", "tmpfs", 0, size);
mount("none", "/d_new", "unionfs", 0, "dirs=/d_rw=rw:/d_root=ro");

```

4. Остання дія полягає в заміні кореневого каталогу на каталог */d\_rw* та виконання команди */sbin/init* як початкового процесу:

```
chdir("/d_new");
stat("/", &st);
rootdev = st.st_dev;
stat(".", &st);
statfs("/", &stfs);
if(mount(".", "/", NULL, 8192, NULL)) return 13;
chroot(".");
chdir("/");
execl("/sbin/init", "init", (char*)NULL);

```

Якщо скопіювати таку програму з використанням *dietlibc* (стандартна бібліотека мови програмування С, що розроблена Феліксом вон Лейтнером з метою компіляції та лінкування програм у виконуваний файл якомога меншого розміру), то розмір результуючої програми буде 2-3 Кб. Її слід розмістити в кореневому каталозі разом з основною файловою системою, яка в даному випадку знаходиться у файлі *Y.sfs*. Також потрібно створити три порожні каталоги */d\_root*, */d\_rw*, */d\_new*, а в каталозі */dev* слід створити файл блочного пристрою *loop0*. Цю файлову систему слід помістити в архів *сріо*.

Цей підхід має наступні переваги:

1. Під час завантаження не потрібно шукати основну файлову систему. Це спрощує і прискорює процес завантаження;
2. Спрощення завантаження по мережі (для завантаження потрібно лише 2 файли);
3. Після завантаження носій може бути видалений, тому що вся основна файлова система буде знаходитися в оперативній пам'яті у стисненому вигляді.

Для демонстрації цього підходу були перероблені дистрибутиви Purru Linux та Fat Dog. Змінені таким чином дистрибутиви продемонстрували вищу швидкість завантаження та високу надійність в роботі.

**Технологія AJAX у інформаційних системах автоматизованого збору та опрацювання даних**  
Словінський О.В., Словінська О.Д.

*Житомирський державний університет імені Івана Франка, alexslv@zu.edu.ua*

The report highlights the main benefits and the description of methods and techniques using AJAX technology to develop information systems of automated data collection and data processing. We consider the introduction of technology in information system "Master" at Zhytomyr State University named after Ivan Franko.

Сучасний світ характеризується широким використанням систем опрацювання інформації, які ґрунтуються на застосуванні розвинутих інформаційних технологій, зокрема локальної комп'ютерної мережі. Новітні інформаційні системи об'єднують в собі інформаційні процеси, які задовольняють потреби в доступі до інформації на всіх рівнях управління та прийняття рішень. Для ефективного використання інформації важливими завданнями стають способи її накопичення, реєстрації, обробки і передачі. Сьогодні архітектура клієнт-сервер є домінуючою концепцією у створенні розподілених мережних інформаційних систем та застосувань і передбачає взаємодію та обмін даними між ними. Одним з найважливіших завдань, що вирішується при проектуванні інформаційних систем, є створення зручного і відповідного цілям інформаційної системи призначеного для користувача інтерфейсу. Звичайно задача ергономічності не формалізується, але в той же час вона є дуже істотною. Користувачі часто судять про якість системи в цілому, виходячи з якості інтерфейсу користувача. Більш того, від його якості залежить ефективність інформаційної системи в цілому.

Останнім часом широкого розповсюдження набула технологія AJAX, яка представляє собою сучасний підхід до побудови інтерфейсу користувача. Дана технологія передбачає відправку запитів на сервер у фоновому режимі і довантаження потрібних користувачу даних, при цьому не перезавантажується вся веб-сторінка. AJAX – це скоріш за все концепція використання декількох суміжних технологій, не самостійна технологія в чистому вигляді. Підхід до розробки призначених для користувача зручних інтерфейсів за допомогою AJAX поєднає декілька основних методів і прийомів:

- використання Dynamic HTML для динамічної зміни вмісту сторінки та надання інтерактивності;