Навчально-методична література

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМ. ІВАНА
ПУЛЮЯ

КАФЕДРА КОМП'ЮТЕРНО-
ІНТЕГРОВАНИХ ТЕХНОЛОГІЙ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з курсу

«Основи автоматизованого проєктування складних об'єктів і систем»

для студентів освітнього рівня магістр за спеціальністю 174"Автоматизація, комп'ютерно-інтегровані технології та робототехніка"

Тернопіль
2023

Методичні вказівки до виконання лабораторних робіт з курсу «Основи автоматизованого проектування складних об’єктів і систем» для студентів освітнього рівня магістр за спеціальністю 174 "Автоматизація, комп’ютерно-інтегровані технології та робототехніка" / В.В. Левицький– Тернопіль: ТНТУ, 2023. – 31 с.

Рецензент: к.т.н., доцент Коноваленко І.В.
Відповідальний за випуск: к.т.н., доцент Левицький В.В.

Розглянуто на засіданні кафедри комп’ютерно-інтегрованих технологій і рекомендовано до друку (протокол №6 від 7.12.2023 р.)

Схвалено та рекомендовано до друку НМК факультету прикладних інформаційних технологій та електроінженерії (протокол №4 від 12.12.2023 р.)

ЛАБОРАТОРНА РОБОТА №1

ТЕМА: Оцінка характеристик паралелізму задач

МЕТА: Ознайомлення з основними характеристиками паралелізму задач за ярусно-параллельною формою (ЯПФ) алгоритму, придбання навичок побудови ЯПФ і оцінки її характеристик на прикладі обчислення арифметичного вираження.

1.2. ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1.2.1. Вивчити розділ 1.3 дійсних вказівок

1.2.2. За заданим викладачем варіантом (див. табл.1.1) скласти ЯПФ алгоритму обчислення арифметичного виразу й оцінити основні характеристики паралелізму: довжину і ширину ЯПФ, коефіцієнти прискорення і завантаження. Число процесорів у ОС прийняти рівним ширині ЯПФ.

1.2.3. Оформити звіт по роботі (див. п. 1.4).

1.3. ХАРАКТЕРИСТИКИ ПАРАЛЕЛІЗМУ ЗАДАЧ

Найбільш розповсюдженим типом паралелізму задач, реалізованих на обчислювальних системах, є паралелізм незалежних галузей. Суть цього типу паралелізму полягає у тому, що в алгоритмі рішення задачі на тих чи інших етапах можуть бути виділені незалежні модулі - галузі, що у ОС можуть виконуватися одночасно, тобто паралельно, по незалежних програмах.

Галузь А програми не залежить від галузі Б, якщо одночасно виконуються наступні чотири умови:

1) Між А і Б немає функціональних зв'язків, тобто жодна з вхідних перемінних для Б не є вихідною перемінної галузі А або залежної від галузі А;

2) Між А і Б немає зв'язку по робочих полях пам'яті: А і Б не роблять запису в ті самі комірки пам'яті;

3) А і Б виконуються за різними програмами (чи копіями програм);

4) А і Б незалежні за керуванням, тобто умови виконання галузі Б не повинні залежати від ознак, вироблених при реалізації А чи іншої галузі, що залежить від А.

Зручною графічною моделлю представлення паралелізму незалежних галузей є ярусно-паралельна форма (рис. 1.1). Усі галузі в ЯПФ розбиваються по *N+1* ярусах; у ярус 0 входять галузі, кожна з який не залежить (у вищевказаному змісті) ні від однієї іншої галузі; у *1-й* ярус - галузі, що залежать тільки від галузей 0-го ярусу;... у *i-й* ярус - галузі, що залежать від галузей (*i-1*)-го ярусу і, можливо, від галузей ярусів з номерами, меншими (*i-1*); і т.д. На рис. 1.1 галузі позначені кільцями з відповідними номерами (замість номерів можуть бути інші ідентифікатори), функціональні зв'язки між галузями показані відрізками, причому, якщо відрізок не виходить ні з якого кільця, то він відповідає вхідній перемінній програми; якщо відрізок з'єднує два кільця, скажемо, від *a* до *b*, то цей

відрізок відповідає вихідній перемінний для галузі *a* і вхідній - для *b*; якщо відрізок не входить ні у яке кільце, то він відповідає вихідній змінній програмі. Можливі “розщеплення” відрізка означають, що ті самі змінні є вхідними для двох чи більше галузей.

Кожну галузь для зручності ідентифікації можна пронумерувати подвійним індексом виду *(i, j)*, де *i* -номер ярусу, *j* - номер галузі в цьому ярусі (див. рис. 1.1).

Найважливіша характеристика галузі *(i, j)* - це її “довжина” l_{ij}, що залежно від ситуації інтерпретується або як трудомісткість реалізації галузі, або як час реалізації на конкретному обчислювальному пристрої.

Таблиця 1.1

Варіант	α	β	Арифметичний вираз
01	2	4	*abc + de/f + g (h + k) + m*
02	2	3	*ab + cd + e/(f + g) + (h + k)/m*
03	3	5	*(a + b + c)/d + ef/g + hkm*
04	2	4	*a + b + cd + e/f + gh + k/m*
05	2	3	*(ab + c) (de + f) + (g + h)/(k + m)*
06	3	5	*(ab + cd + e/f + gh)/k + m*
07	2	3	*(abcde + f)/(gh +k/m)*
08	2	4	*a/b + c/d + ef/(g + h) + k/m*
09	3	5	*(a + b)/(c + d) + efg + h/(k + m)*
10	3	5	*(ab + cde)/(fg + h/k) + m*
11	2	5	*(a + b + c)/(d + e + f) (g + hk/m)*
12	2	3	*(a + bc)/(d + ef) + g/(h + km)*
13	2	4	*(a/b + cd + e/f) (gh + k/m)*
14	2	3	*(a + b + c + d/e + fg)/(h + km)*
15	3	5	*a/b + c/d + e/f + ghk + m*
16	2	4	*a/(b + c + d + ef + gh) + k/m*
17	2	3	*(a + bc)/(d + ef) + g/(h + k + m)*
18	3	5	*ab + cd + ef + g/h + k/m*
19	2	3	*(a + b)/(c + d)+(e + f)/(h + k) + m*
20	2	4	*(a + b)(c + d)(e + f)(h + k)/m*
21	3	5	*(ab + cd)/(ef + hk)/m*
22	2	5	*(a + b + c)(d + e/f)(hk + g) m*
23	2	3	*a/b/c/d + (e + f + g + h)k + m*
24	2	4	*(a + b)/c + (d + e)/f + (g + h)k/m*
25	2	5	*(a + bc)(d + ef)/(g + hk) + m*

26	3	4	*(abcd + efgh)/(k/m)*
27	3	5	*(a/b + c/d + e/f + g/h)(k + m)*
28	2	5	*abcd/e + (f + g + h)k + m*
29	2	3	*(a + b)/(cd) + e/f + g/h + k/m*
30	2	4	*(ab + cd)/(e + f + g/h)/k + m*

Формування ЯПФ і оцінювання l_{ij}- досить складний процес, що залежить від самої задачі і від кваліфікації системного програміста. У сучасних автоматизованих системах побудовою ЯПФ займається спеціальний блок ОС або спеціальна програма-диспетчер. Однак, для простих задач ЯПФ можна побудувати вручну.

Маючи ярусно-паралельну форму, легко оцінити наступні основні характеристики паралелізму відповідної задачі:

Ширина *B* ЯПФ*:*

$$B = \max_{0 \le i \le N} \{B_i\}, \qquad (1.1)$$

де: B_i - ширина *i-го* ярусу - кількість галузей у цьому ярусі.

Довжина *L* ЯПФ*:*

$$L = \sum_{i=0}^{N} l_i, \qquad (1.2)$$

де: l_i - довжина *i-го* ярусу:

$$l_i = \max_{1 \le j \le b_i} \{l_{ij}\}. \qquad (1.3)$$

Наприклад, для рис.1.1, легко підрахувати, *B = 5, L = 19*.

Характеристики (1.1) і (1.2) у значній мірі можна вважати атрибутами (невід'ємними властивостями) задачі, поза залежністю від того, на яких ОС вона реалізується.

Розглянемо тепер можливість реалізації ЯПФ на деякої мультипроцесорної ОС. Зробимо наступні припущення:

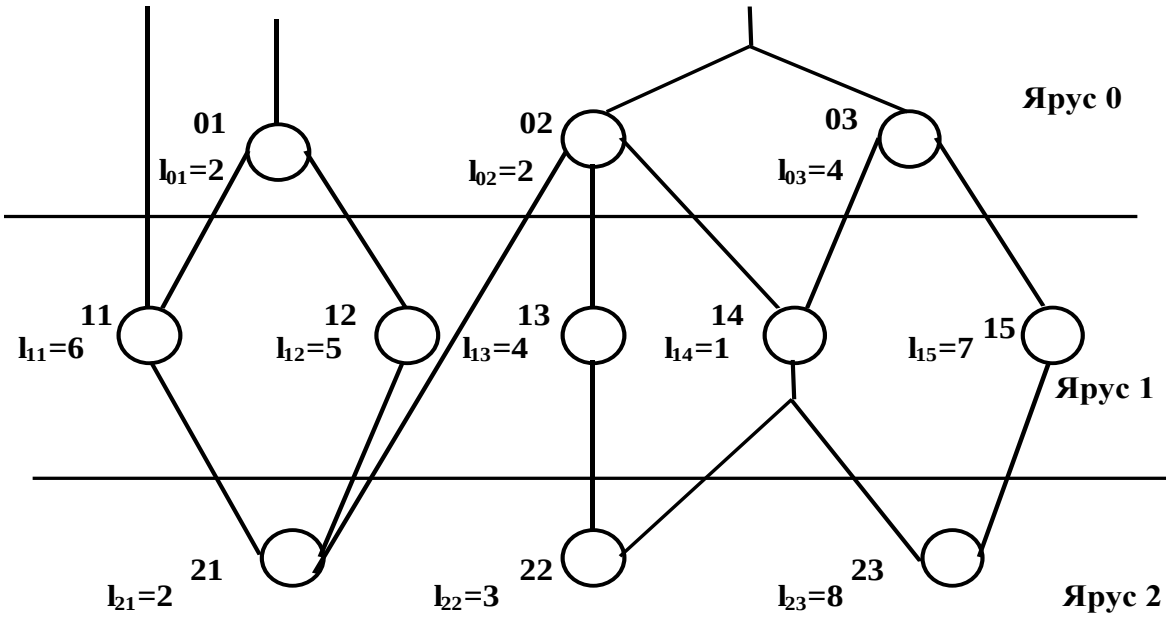


Рисунок 1.1

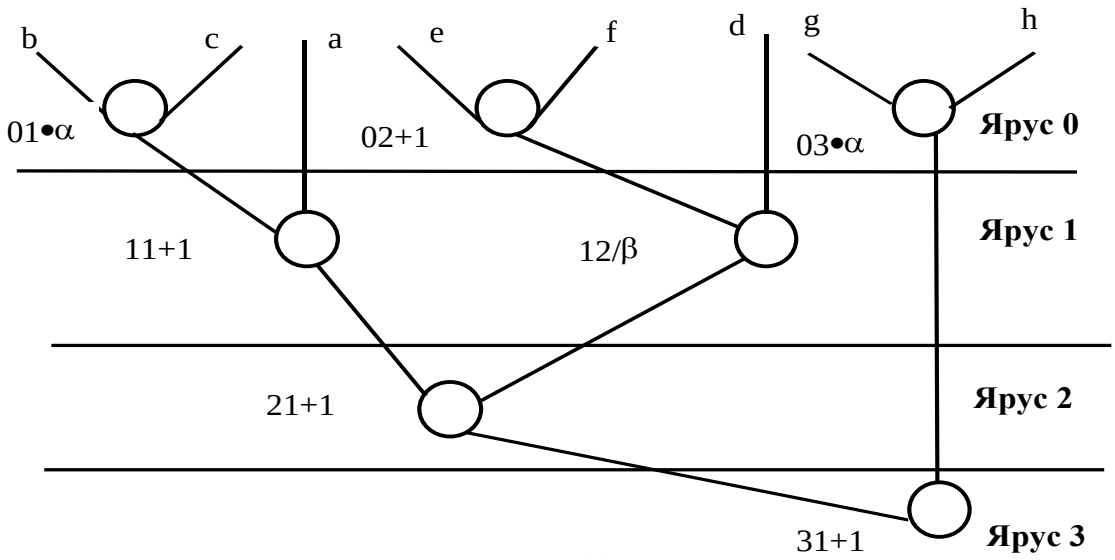


Рисунок 1.2

- час l_{ij} реалізації галузі *ij* не залежить від того, на якому процесорі ОС вона реалізується (тобто, усі процесори функціонально ідентичні);

- число процесорів ОС не обмежено, у тім змісті, що система для реалізації будь-якого ярусу ЯПФ може надати стільки процесорів, скільки необхідно для повного розпаралелювання робіт на даному ярусі;

- ЯПФ на ОС реалізуються синхронно: усі галузі чергового ярусу починають виконуватися одночасно, після завершення робіт найдовшої галузі попереднього ярусу.

Тоді характеристики *B* і *L* набувають наступного сенсу: *B* - мінімальне число процесорів для повного розпаралелювання, а *L* - мінімальний час рішення задачі на ОС при синхронній реалізації ЯПФ:

$$L = L_{min}. \qquad (1.4)$$

З іншого боку, при кількості n процесорів ОС, яка дорівнює 1 (однопроцесорна ОС), час вирішення задачі, очевидно, дорівнює максимальному:

$$L_{\max} = \sum_{i=0}^{N} \sum_{j=1}^{b_i} l_{ij} . \qquad (1.5)$$

Таким чином, час $L(n)$ рішення задачі на n - процесорної ОС, $n \le B$, лежить у межах

$$L_{min} \le L(n) \le L_{max} . \qquad (1.6)$$

Однією з основних проблем мультипроцесорних ОС є проблема вибору архітектури ОС і організації обчислювального процесу. Це слід виконувати таким чином, щоб забезпечити $L(n)$, яке буде максимально наближеним до L_{min} при даному n. Про це мова йтиме в наступних роботах. Розглянемо ще дві корисні характеристики ефективності обчислень.

Коефіцієнт K прискорення обчислень:

$$K = \frac{L_{\max}}{L_{\min}} \qquad (1.7)$$

Коефіцієнт прискорення, очевидно, показує, у скільки разів швидше обчислюється задача на *n-процесорній* системі, порівняно з одно процесорною ОС. Даний коефіцієнт характеризує ступінь підвищення продуктивності при використанні багатопроцесорних ОС.

Про ефективність використання устаткування свідчить коефіцієнт завантаження:

$$\rho = \frac{L_{\max}}{nL(n)} \ , n \le B, \qquad (1.8)$$

який показує частку процесорного часу L_{max}, використаного на вирішення задачі, стосовно загального часу $n(n)$. З (1.8) і (1.7) випливає, що

$$\rho = K/n \ , \ \ n \le B. \qquad (1.9)$$

Розглянемо процес побудови ЯПФ і оцінки характеристик паралелізму на прикладі обчислення арифметичного виразу:

$$a + bc + d/(e + f) + gh. \qquad (1.10)$$

Нехай відомо, що процесор ОС витрачає на виконання операцій множення, ділення і додавання часи t_y, t_d і t_c, пов'язані співвідношеннями:

$t_y = \alpha t_c$,

$t_d = \beta t_c$.

Без обмеження можна прийняти, що $t_c = 1$; тоді $t_y = \alpha$ і $t_d = \beta$. Відомо також, що $\beta > \alpha > 1$.

Використовуючи ці дані, а також визначення незалежності галузей, будуємо ЯПФ виразу (1.10). Вона має вигляд, що показано на рис. 1.2. Безпосередньо з цього рисунка одержимо, використовуючи співвідношення (1.1) - (1.4):

$B = max \{B_0, B_1, B_2, B_3\} = max \{3, 2, 1, 1\} = 3$,

$l_0 = max \{\alpha, 1, \alpha\} = \alpha$,

$l_1 = max\{1, \beta\} = \beta$,
$l_2 = 1$,
$l_3 = 1.$,
Тоді:
$L = l_0 + l_1 + l_2 + l_3 = \alpha + \beta + 2.$
Далі, за формулою (1.4):
$L_{max} = 2\alpha + \beta + 4$.
Якщо число n процесорів ОС не менше 3, то при цьому, згідно (1.4), досягається мінімально можливий час рішення задачі:
$L_{min} = L = \alpha + \beta + 2$.
Коефіцієнт прискорення обчислень, відповідно до (1.7), дорівнює:

$$K = K_{\max} = \frac{2\alpha+\beta+4}{\alpha+\beta+2} = 1 + \frac{\alpha+2}{\alpha+\beta+2}.$$

Коефіцієнт завантаження (1.9), при n = 3:

$$\rho = K / 3 = 1/3 + \frac{\alpha+2}{3(\alpha+\beta+2)} .$$

Наприклад, при α = 2 і β = 3: K = 1,56 і ρ = 0,52.

1.4. ЗМІСТ ЗВІТУ

Постановка задачі, ЯПФ арифметичного виразу, результати розрахунків і висновки.

ЛАБОРАТОРНА РОБОТА №2

ТЕМА: Визначення характеристик ядра МПС із загальною пам’яттю.

МЕТА: Ознайомлення з основними характеристиками центральної частини (ядра) мультипроцесорної системи й освоєння методики їх оцінювання методами теорії масового обслуговування. Визначення характеристик ядра МПС із загальною пам’яттю.

2.2. ПОРЯДОК ВИКОНАННЯ РОБОТИ:

2.2.1. Вивчити розділ 2.3.

2.2.2. За заданим варіантом табл. 2.1. побудувати графік U(N) залежності часу U реакції ядра МПС від числа N процесорів, вважаючи відомими параметри: λ - інтенсивність потоку завдань; Θ - трудомісткість одного завдання; B - швидкодію процесора. За графіком визначити кількість N^* процесорів, що забезпечують обробку завдань з часом реакції, не великим

згідно u^*. Обчислити завантаження R, ймовірність простою p_0 і середнє число l завдань у черзі в N^*- процесорній системі.

2.2.3. Оформити звіт по роботі (див. п. 2.4).

2.3. ХАРАКТЕРИСТИКА ЯДРА МПС

Центральна частина, чи ядро, більшості сучасних мультипроцесорних систем (МПС) має структуру, вигляд якої зображено на рис. 2.1, де: ПР – центральний процесор, МП – модуль оперативної пам'яті. Комутаційна мережа (у різних МПС її називають комутаційною матрицею або системою з перехресними зв'язками) призначена для інформаційного зв'язку – передачі команд, даних між зазначеними пристроями. Якщо комутаційна мережа забезпечує доступ будь-якого пристрою периферійного процесора до будь-якого модуля оперативної пам'яті, то говорять про МПС із загальною пам'яттю.

Однієї із задач системотехнічного проектування МПС із загальною пам'яттю є проблема оцінки продуктивності ядра МПС. Продуктивність λ зазвичай оцінюють середнім числом завдань, що виконує (розв'язує) система за одиницю часу. Продуктивність еквівалентна середньому часу u виконання одного завдання, який, у свою чергу, називають часом реакції ядра МПС.

Оцінювання значення u є складною задачею. На початковому етапі цю задачу вирішують методами імітаційного моделювання з наступною статистичною обробкою результатів імітації. Однак, для наближеної оцінки, на рівні ескізного проектування, можна цілком обійтися аналітичними методами, що у стані забезпечити достатньо високу точність оцінювання результатів експерименту.

Таблиця 2.1

Варіант	λ, $с^{-1}$	Θ, млн.оп.	B, млн.оп./с	u^*, с
01	0.2	120	10	20
02	0.3	130	10	30
03	0.1	90	10	10
04	0.2	110	20	10
05	0.3	120	20	20
06	0.4	120	20	20
07	0.5	120	20	10
08	0.1	130	30	20
09	0.2	130	30	30
10	0.3	130	30	40
11	0.4	130	30	10
12	0.5	100	40	20
13	0.1	120	40	40
14	0.2	130	40	30

15	0.3	140	40	10
16	0.4	150	40	30
17	0.5	110	25	30
18	0.1	130	25	30
19	0.2	140	25	20
20	0.3	110	10	40
21	0.4	140	10	10
22	0.5	120	40	30
23	0.1	120	30	20
24	0.2	130	30	10
25	0.3	130	40	40
26	0.2	110	15	20
27	0.3	120	15	30
28	0.1	80	15	20
29	0.2	100	15	20
30	0.4	120	15	30

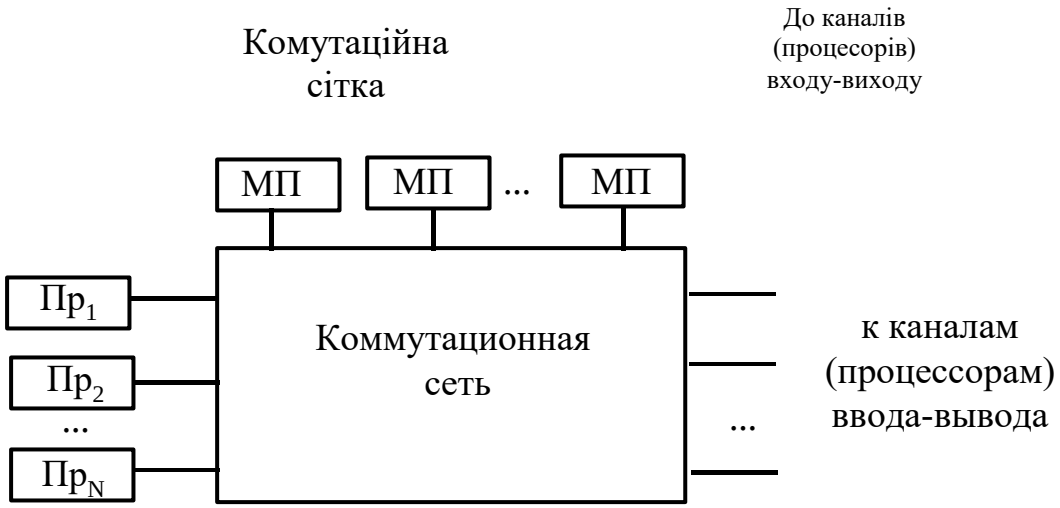


Рисунок 2.1- структура ядра МПС

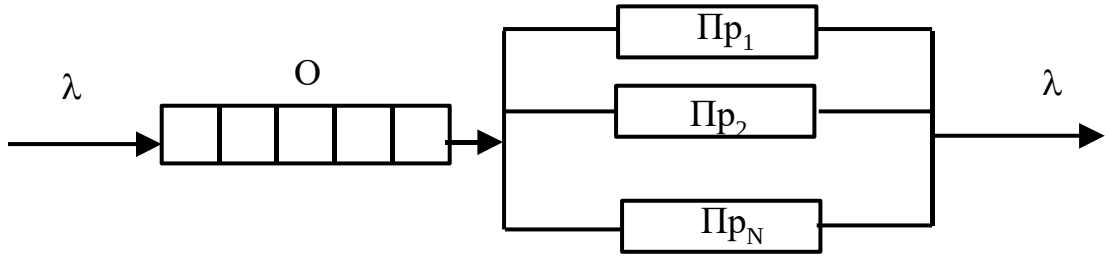


Рисунок 2.2 – Модель ядра МПС із загальною пам’яттю

Разом з оцінкою тимчасової реакції оцінюють й інші робочі характеристики МПС: завантаження, коефіцієнт простою ядра, довжину черги на обробку та ін.

Оцінимо час реакції й інші характеристики ядра МПС аналітично, використовуючи результати теорії масового обслуговування.

Ядро МПС, яке показано на рис. 2.1, можна представити як систему масового обслуговування (рис. 2.2). У цьому випадку розглядають N – канальну систему масового обслуговування (СМО) з чергою без втрат. Модель такої СМО побудуємо, вважаючи справедливими наступні припущення. Програми і вихідні дані для будь-якого завдання, що надходить на обробку в МПС, розміщаються у загальній пам'яті, і виконання кожної програми від початку до кінця виробляється без переривань. Потік завдань, що надходить у ядро через канали (процесори) введення, є найпростішим з інтенсивністю λ. Трудомісткість кожного завдання – випадкова величина, розподілена за показниковим законом із середнім значенням Θ. Усі N процесори ядра функціонально ідентичні і мають швидкодію B. Завдання, що надходять на обробку, перебувають у черзі О (рис. 2.2) і виходять з неї на будь-який вільний процесор після того, як він закінчить обслуговувати попередню вимогу. Припустимо, що всі пристрої МПС працюють без збоїв і відмов.

Характеристики ядра визначимо для МПС, що функціонує в стаціонарному режимі. Для системи (рис. 2.2) умова існування стаціонарного режиму виглядає так:

$$\lambda\Theta < NB \qquad (2.1)$$

і має фізичний зміст: необхідно, щоб сумарна швидкодія NB процесорів ядра перевищувала необхідну для обробки завдань середню швидкість λΘ обчислень.

Якщо умова (2.1) виконується, тоді, відповідно до теорії масового обслуговування, час реакції є середнім часом u перебування вимоги у СМО типу M/M/N/∞ у стаціонарному режимі. Використовуючи відомі формули теорії масового обслуговування, маємо:

$$u = \frac{\Theta}{B}\left[1 + \frac{R^N}{(N-1)!(N-R)^2}\, p_0\right], \qquad (2.2)$$

де:

$$R = \lambda\Theta / B, \tag{2.3}$$

$$p_0 = \frac{1}{\sum_{i=0}^{N} \frac{R^i}{i!} + \frac{R^{N+1}}{N!(N-R)}}. \tag{2.4}$$

Величину *R*, яку визначають з формули (2.3), називають завантаженням або коефіцієнтом мультипрограмування ядра МПС. Ця величина показує середнє число процесорів, зайнятих обробкою завдань у стаціонарному режимі. Як випливає з (2.1), R<N.

Величина p_0 має сенс коефіцієнта простою ядра і є ймовірністю того, що одночасно всі *n* процесорів вільні.

Вкажемо ще на одну корисну характеристику ядра – середнє число *l* завдань, що знаходяться у черзі на обробку. У розглянутій моделі *l* обчислюють за формулою

$$l = u\lambda - R. \tag{2.5}$$

2.4. ЗМІСТ ЗВІТУ

Постановка задачі, графік функції *u(N)*, результати розрахунків і висновки.

ЛАБОРАТОРНА РОБОТА №3

ТЕМА: Оцінка ефективності паралельних ОС.

МЕТА: Ознайомитись з архітектурою паралельних обчислювальних систем класу МКМД, освоїти методи реалізації паралельних алгоритмів на ОС і способів оцінки ефективності паралельних ОС.

3.2. ПОРЯДОК ВИКОНАННЯ РОБОТИ:

3.2.1. Вивчити розділ 3.3 дійсних вказівок.

3.2.2. Використовуючи ЯПФ арифметичного виразу лабораторної роботи №1, обчислити характеристики ефективності її реалізації (коефіцієнт прискорення і завантаження) на В-процесорній ОС у синхронному й асинхронному режимах. Повторити обчислення для (В-1)-процесорної системи. Порівняти результати і зробити висновки.

3.2.3. Оформити звіт по роботі (див. п. 3.4).

3. ОЦІНКА ЕФЕКТИВНОСТІ ПАРАЛЕЛЬНИХ ОС

Паралельними ОС зазвичай називають мультипроцесорні обчислювальні системи класу МКМД, тобто ОС із множинним потоком команд і множинним потоком даних. Такі системи ефективно реалізують задачі з паралелізмом незалежних галузей. Ядро МПС цього класу розглядали у лабораторній роботі №2. Класичними прикладами паралельних ОС є ЕОМ класу “Ельбрус” і системи серії “У” американської фірми BURROUGHS. Приклад сучасної системи – суперкомп'ютер RS/6000

SP фірми IBM. Гранична ефективність ОС досягається на задачах з регулярними ЯПФ. Для регулярної форми ширина B системи дорівнює ширині кожного з N-1 ярусів, а довжина l_j ярусу j дорівнює довжині кожної з B ланок цього ярусу. Легко перевірити, що B-процесорна ОС реалізує ЯПФ із коефіцієнтом прискорення К=В і з завантаженням $\rho = 1$. Ці граничні значення можуть бути досягнуті лише на задачах спеціального класу, що мають регулярну ЯПФ, або на спеціально перетворених програмах рішення, що наближають ЯПФ до регулярної.

На практиці доводиться мати справу з задачами, ЯПФ яких не є регулярними. У цьому випадку проблема полягає у перетворенні форми до виду, найбільш придатного для реалізації ОС. Якщо таке перетворення можливо без порушення змісту самої задачі, тоді у подальшому оцінюють ефективність реалізації програмної ЯПФ на ОС.

Розглянемо процес перетворення ЯПФ і оцінки ефективності ОС на прикладі реалізації арифметичного виразу (лабораторна робота №1):

$$a + b \cdot c + d / (e + f) + g \cdot h .$$

ЯПФ цього виразу показано на рис. 3.1, де $\alpha = 2$ і $\beta = 3$.

Якщо приймають 3-процесорну паралельну ОС, тоді її без усяких перетворень можна реалізувати так, як показано на рис. 3.2, де прийнято: $t_з=1$, $t_у=2$ і $t_д=3$.

Рисунок подібного роду зазвичай називають діаграмою Ганта. Нагадаємо, що діаграма (рис. 3.2) відповідає синхронної реалізації ЯПФ. З цього рисунка безпосередньо видно, що L=7, L_{max}=11. Тоді коефіцієнт прискорення K=11/7=1,56, а коефіцієнт завантаження $\rho=K/3$=0,52.

(Зазначимо, що діаграма на рис. 3.2 дозволяє обчислити ρ, використовуючи саме поняття завантаження як співвідношення сумарного корисного часу процесорів, яке дорівнює L_{max} =11, до загального витраченого часу, тобто 7·3=21).

Розглянемо тепер випадок 2-процесорної ОС. На перший погляд здається, що задача не може бути паралельно реалізована, оскільки ярус 0 вимагає наявності 3-х процесорів. Однак, як легко побачити, перетворена форма, зображена на рис. 3.3, еквівалентна ЯПФ (рис. 3.1) у змісті задачі (3.1). У той же час для реалізації перетвореної ЯПФ достатньо 2-х процесорів. Процес реалізації показано на рис. 3.4. З діаграми випливає, що при синхронній реалізації ЯПФ тривалість вирішення задачі дорівнює 8, коефіцієнт прискорення K=11/8=1,38, а завантаження ρ=11/(2·8)=0,69.

Дотепер мали справу із синхронною реалізацією ярусів ЯПФ задачі. Ефективність ОС можна підвищити, якщо зняти обмеження синхронності, а саме: дозволити процесору, що звільнився на якому-небудь ярусі, почати реалізацію галузі наступного ярусу негайно, не чекаючи звільнення інших процесорів. Проілюструємо результат зняття цього обмеження на прикладі реалізації ЯПФ (рис.3.3).

Відповідну діаграму показано на рис.3.5. Доведено, що K=11/7 =1,56, а ρ=11/(2·7) = 0.79. Тобто, у даному випадку досягнуто таке ж прискорення,

як і у 3-процесорній системі, з одночасним майже 2- кратним збільшенням завантаження ОС.

Слід зазначити, що таке можливо лише в моделях обчислювальних процесів, які не враховують витрати процесорного часу на підготовку корисного процесу і на різні затримки, пов'язані з пересиланням файлів, команд і даних в оперативну пам'ять. У випадках, де ці витрати малі, порівняно з довжиною галузей ЯПФ, асинхронний процес виявляється ефективнішим. Зокрема, в обчислювальних комплексах серії “Ельбрус” і Spark передбачена саме асинхронна реалізація ярусів ЯПФ, тобто динамічний оптимальний розподіл завдань – ланок між процесорами й іншими ресурсами системи.

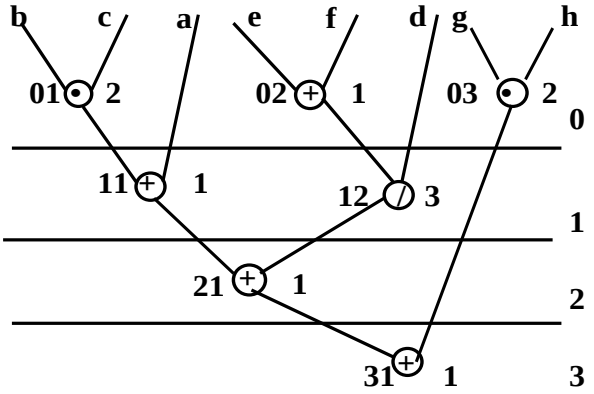


Рисунок 3.1

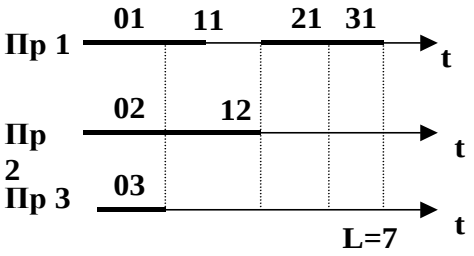


Рисунок 3.2

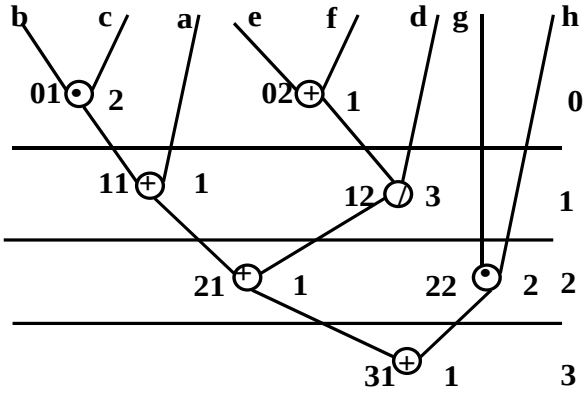


Рисунок 3.3

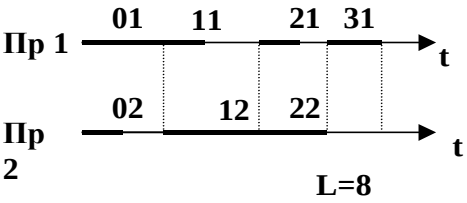


Рисунок 3.4

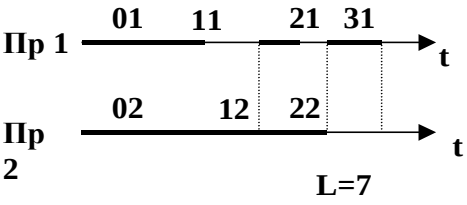


Рисунок 3.5

3.4. ЗМІСТ ЗВІТУ

Постановка задачі, ЯПФ, діаграми Ганта для синхронних і асинхронних реалізацій ЯПФ *B* і (*B-1*)-процесорними ОС, результати розрахунків і висновки.

ЛАБОРАТОРНА РОБОТА №4

ТЕМА: Оцінювання ефективності конвеєрних ОС.

МЕТА: Ознайомлення із принципом реалізації конвеєрних обчислювальних систем, освоєння методів підготовки задач для конвеєра процесорів і придбання навичок оцінювання ефективності конвеєрних ОС.

4.2. ПОРЯДОК ВИКОНАННЯ РОБОТИ:

4.2.1. Вивчити розділ 4.3 дійсних вказівок

4.2.2. Використовуючи методику розділу 4.3, побудувати конвеєр процесорів для обчислення заданого арифметичного виразу й оцінити характеристики цього конвеєра щодо характеристик послідовної синхронної реалізації ЯПФ арифметичного виразу.

4.2.3. Оформити звіт по роботі (див. п.4.4).

4.3. ОЦІНКА ЕФЕКТИВНОСТІ КОНВЕЄРА ПРОЦЕСОРІВ

Конвеєрні обчислювальні системи, зокрема, реалізують принцип обробки МКОД (множинний потік команд - одиночний потік даних), і з цієї причини їх часто ототожнюють тільки з ОС цього класу, хоча, поняття конвеєрної обробки ширше поняття МКОД.

Пояснимо загальний принцип конвеєрної обробки даних на наступному прикладі. Нехай потрібно обчислити функцію *F(x)* для послідовності x^1, x^2, ... даних по алгоритму

$$F(x) = f_3(f_2(f_1(x))) , \qquad (4.1)$$

де: f_1, f_2 і f_3 - задані функції.

При цьому відомо, що на операцію f_i ОС витрачає час τ_i, $i = 1,2,3$. Тут як аргумент x, так і функції f_i і F можуть бути як скалярними, так і векторними.

При реалізації послідовного алгоритму (4.1) результат *F(x)* з'явиться на виході ОС протягом часу:

$$\tau_n = \sum_i \tau_i \qquad (4.2)$$

Тепер припустимо, що є три процесори, а процесор *i* спеціалізується на виконанні операції f_i. Зв'яжемо процесори у *конвеєр* за схемою (рис. 4.1) і визначимо **такт конвеєра**, $\tau_{до}$, як:

$$\tau_к = \max \{\tau_i\}. \qquad (4.3)$$

У кожному такті будемо подавати на вхід конвеєра, тобто на вхід процесора 1, черговий набір даних. Тобто у першому такті - x^1, у другому - x^2 і т.д. У результаті одержимо схему, чи діаграму обчислень, показану на табл. 4.1. Зі схеми видно, що перший результат – $F(x^1)$, з'явиться на виході конвеєра після 3 тактів, тобто через час $3\tau_к$ (іноді його називають **часом**

розвантажування). Тоді кожен наступний результат – $F(x^2)$, $F(x^3)$ і т.д. – у кожному наступному такті, тобто через час $\tau_к$. При досить довгій послідовності $x^1, x^2, ...$ можна не враховувати час розвантажування, і тоді ефективність конвеєра, яку оцінюють коефіцієнтом прискорення K, можна визначити за формулою:

$$K = \tau_n / \tau_к. \qquad (4.4)$$

Наведені положення (4.1)-(4.4), можна узагальнити на випадок довільного числа n процесорів у конвеєрі. У цьому випадку, підставляючи у (4.4) значення τ_n і $\tau_к$ з (4.2) і (4.3), одержимо наступне:

$$K = \sum_{i=1}^{n} \tau_i / \max\{\tau_i\} \le n \max\{\tau_i\},$$

тобто:

$$K \le n, \qquad (4.5)$$

а завантаження конвеєра:

$$\rho = K/n \le 1. \qquad (4.6)$$

Граничне значення $K_{max} = n$ досягається при однаковій тривалості τ_i обробки даних у всіх процесорах конвеєра і при "суцільній" завантаженості усіх процесорів на всіх тактах роботи. У цьому випадку коефіцієнт завантаження ρ дорівнює 1.

У реальних конвеєрних ОС цього не можна досягнути внаслідок припинення роботи конвеєра в результаті виконання команд умовного і безумовного переходів. Реальне прискорення можна оцінити за формулою:

$$K = n(1 + \sum_{i=1}^{n} i p_i), \qquad (4.7)$$

де: p_i - ймовірність припинення роботи конвеєра на i тактів. Завантаження ρ визначають за формулою: $\rho = K/n$.

У цьому випадку завантаження дорівнює:

$$\rho = 1/(1 + \sum_{i=1}^{n} i p_i). \qquad (4.8)$$

Побудуємо конвеєрну ОС для обчислення відомого арифметичного виразу (див. п. 4.1):

$$a + bc + d/(e + f) + gh\,. \qquad (4.9)$$

Ефективність конвеєрної ОС оцінимо щодо тривалості послідовної синхронної реалізації ЯПФ цього виразу двопроцесорною паралельною ОС із попередньої роботи. Вважаємо, що тривалість операцій додавання,

множення і ділення рівні відповідно 1, 2 і 3. Згадана ЯПФ скопійована у лівій частині рис. 4.2. Для цієї форми, у позначеннях типу (4.1), можна записати:

для вектора *x* вихідних даних:

$$x = (a, b, c, d, e, f, g, h),$$

для вектор - функції, що реалізує ярус 0:

$$f_1(x) = (bc, a, e + f, d, g, h),$$

також, використовуючи позначення ЯПФ:

$$01 = bc, \quad 02 = e + f,$$

отримаємо:

$$f_1(x) = (01, a, 02, d, g, h),$$

тоді: $\tau_1 = 2$.

Ярус 1 ЯПФ реалізується вектор-функцією:

$$f_2(f_1) = (11, 12, g, h),$$

і тривалість операції f_2 дорівнює: $\tau_2 = 3$.

Ярус 2 реалізується вектор-функцією:

$$f_3(f_2) = (21, 22)$$

з тривалістю: $\tau_3 = 2$.

Ярус 3 реалізується скалярною функцією:

$$f_4(f_3) = f_4(31) = F(x)$$

з тривалістю: $\tau_4 = 1$.

Отже, час τ_n послідовної реалізації ярусів ЯПФ, відповідно до (4.2), буде визначатись за формулою:

$$\tau_n = \sum_{i=1}^{4} \tau_i = 8 .$$

Побудуємо конвеєр. Спочатку, визначимо такт, $\tau_{до}$, конвеєра:

$$\tau_k = \max_{1 \le i \le 4} \{\tau_i\} = \max\{2,3,2,1\} = 3.$$

Зазначимо, що операція розподілу є елементарною, тому величину $\tau_к$ зменшити неможливо. Отже, коефіцієнт прискорення конвеєра у даному випадку, згідно (4.4), не може перевищувати величини:

$$K = 8/3 = 2,67 . \qquad (4.10)$$

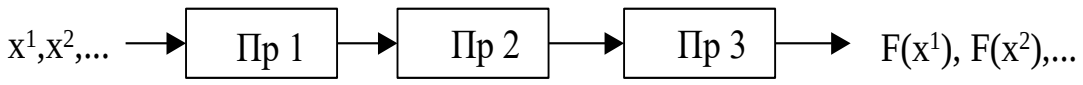


Рисунок 4.1

Таблиця 4.1

Такт / Функція	1	2	3	4	5
f_1	$f_1(x^1)$	$f_1(x^2)$	$f_1(x^3)$	$f_1(x^4)$	$f_1(x^5)$
f_2		$f_2(f_1(x^1))$	$f_2(f_1(x^2))$	$f_2(f_1(x^3))$	$f_2(f_1(x^4))$
f_3			$f_3(f_2(f_1(x^1)))$	$f_3(f_2(f_1(x^2)))$	$f_3(f_2(f_1(x^3)))$
Вихід F(x)				$F(x^1)$	$F(x^2)$

Визначимо структуру конвеєра. Найпростіше для кожного ярусу ЯПФ призначити спеціалізований процесор. Тоді конвеєр буде містити $n = 4$ процесори і завантаження ОС становитиме:

$$\rho = K/n = 2{,}67/4 = 0{,}66. \qquad (4.11)$$

При цьому, як показують співвідношення (4.5) і (4.6), значення K і ρ є далекими від граничних.

Характеристики конвеєра можна поліпшити, якщо для операцій ярусів 2 і 3 ЯПФ призначити один процесор (див. рис. 4.2). Тоді, як неважко бачити, Пр3 виконає всі ці операції за час $\tau_3 = 3$, тобто в межах такту конвеєра, і в той же час завантаження конвеєра зросте до величини

$$\rho = K/3 = 0{,}89\ ,$$

при тому ж самому коефіцієнту прискорення (4.10).

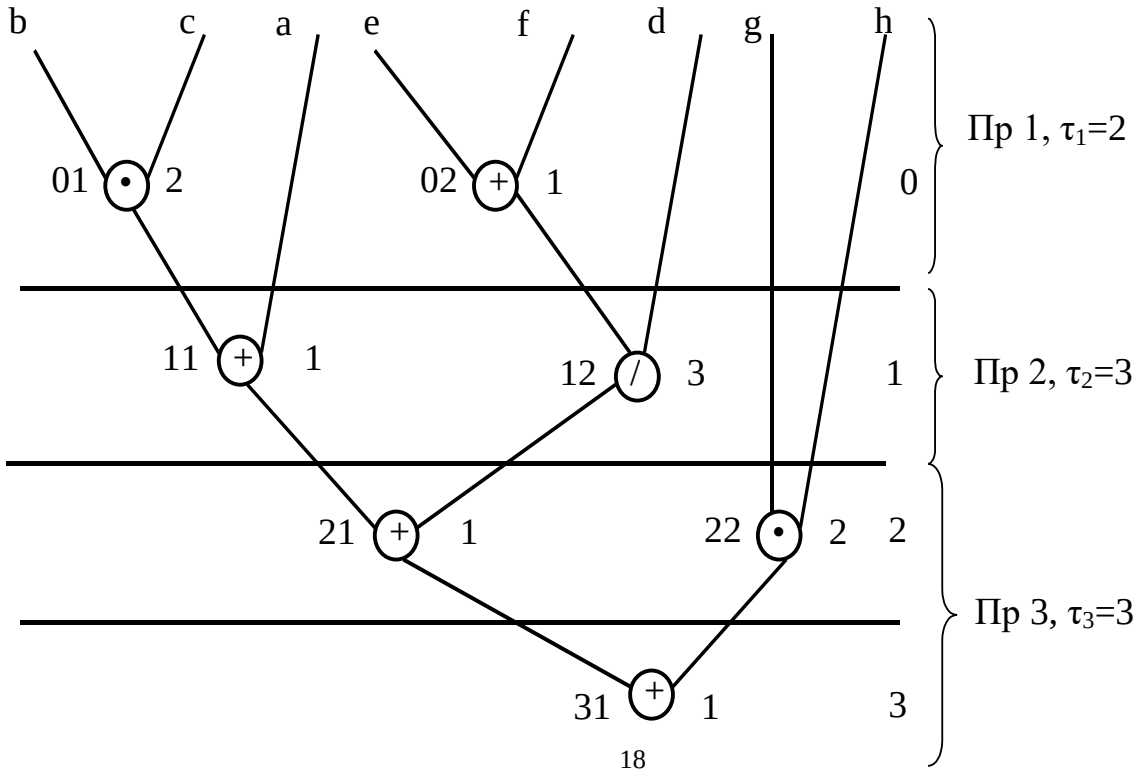


Рисунок 4.2

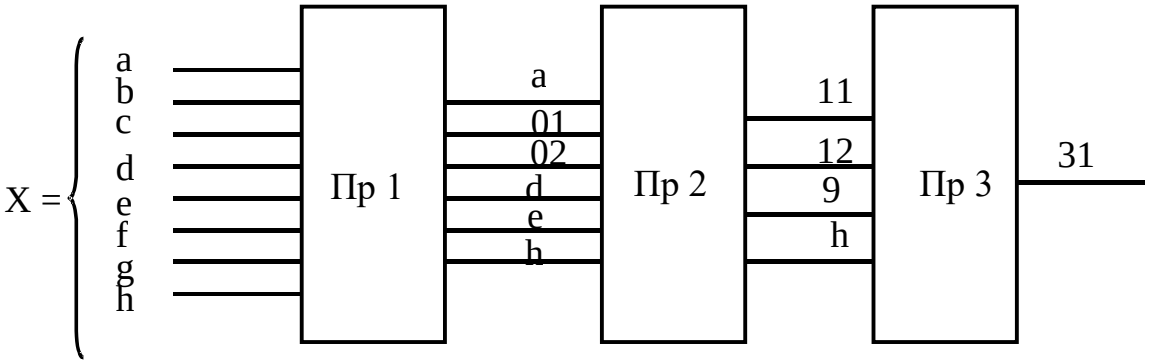


Рисунок 4.3

Отже, у роботі запропоновано 3-процесорну конвеєрну ОС. Структуру конвеєра показано на рис. 4.3. Процесори Пр 1, Пр 2 і Пр 3 реалізують відповідно вектор-функції:

$$f_1(x) = (01, a, 02, d, g, h),$$
$$f_2(f_1) = (11, 12, g, h),$$
$$f_3(f_2) = (21, 22, 31),$$

з виходом 31 = *F(x)*. Тривалість такту конвеєра дорівнює 3, коефіцієнт прискорення стосовно послідовної реалізації ЯПФ дорівнює 2,67, завантаження конвеєра - 0,89.

Введемо поняття *абсолютного* коефіцієнта прискорення. Під абсолютний коефіцієнтом прискорення будемо розуміти відношення часу реалізації арифметичного виразу на ОС класу ОКОД до часу конвеєра. Абсолютний коефіцієнт у даному випадку дорівнює:

$$11/3 = 3{,}67.$$

4.4. ЗМІСТ ЗВІТУ

Постановка задачі, опис її вирішення, результати, включаючи структуру конвеєра і систему відповідних вектор-функцій, висновки.

ЛАБОРАТОРНА РОБОТА №5

ТЕМА: Розрахунок швидкодії процесора і параметрів типового завдання обчислювальної системи.

МЕТА: Придбання навичок розрахунку основних параметрів типового завдання з потоку, що підлягає обробці на однопроцесорній обчислювальній

системі, і визначення необхідної швидкодії процесора при заданому часі реакції системи.

5.2. ПОРЯДОК ВИКОНАННЯ РОБОТИ:

5.2.1. Вивчити розділ 5.3 дійсних вказівок.

5.2.2. По заданому викладачем варіанту розрахунку обчислити параметри (5.1) - (5.6) типового завдання ОС.

5.2.3. Визначити нижню оцінку (5.8) швидкодії *B* процесора ОС, а також мінімально необхідну швидкодію (5.9) процесора при різних часах перебування завдань у системі. Побудувати графік B(ω*) і зробити висновки.

5.2.4. Оформити звіт по роботі (див. розділ 5.4).

5.3. РОЗРАХУНОК ПАРАМЕТРІВ ТИПОВОГО ЗАВДАННЯ І НЕОБХІДНОЇ ШВИДКОДІЇ ПРОЦЕСОРА ОС

Зазвичай обчислювальна система, що підлягає проектуванню, призначена для вирішення деякого потоку завдань, параметри яких більш-менш відомі. У даній роботі значення параметрів завдань вважають відомими і вибирають з відповідних таблиць стосовно заданого варіанту.

Типовим завданням потоку називають завдання, параметри якого усереднені за значеннями параметрів усіх завдань потоку.

Визначимо швидкодію процесора ОС як кількість процесорних операцій, що виконуються ним за одиницю часу.

5.3.1. ВИЗНАЧЕННЯ ПАРАМЕТРІВ ТИПОВОГО ЗАВДАННЯ

Обчислювальний процес у ОС може бути представлений простою моделлю (рис. 5.1). Виконання кожного завдання починається з запиту користувача ОС на вирішення відповідної задачі $З_i$ (i= I,...,M) і закінчується видачею відповіді користувачу у вигляді результатів вирішення. У процесі виконання завдання проходить етапи рахунку (обробка в процесорі зі звертаннями до оперативної пам'яті), що чергуються з етапами звертання до файлів $F_1,...,F_N$, розташованими у зовнішній пам'яті ОС.

Передбачається, що ОС створюється для вирішення відомого набору M задач $З_i$ $(i = I,...,M)$. Кожна задача $З_i$ характеризується інтенсивністю λ_i потоку запитів на її вирішення і середньою кількістю θ_i процесорних операцій, необхідних для виконання відповідного завдання. Взаємодія задачі $З_i$ з файлом F_j характеризується середнім числом N_{ij} звертань до файлу у процесі виконання завдання.

Варіанти перерахованих даних для розрахунків по лабораторній роботі приведені у таблицях 5.1 і 5.2.

Параметри типового завдання обчислюють з використанням наступних припущень і співвідношень. Передбачається, що процес виконання завжди починається етапом рахунку, і що процес є марковським. Тоді справедливі наступні співвідношення:

- інтенсивність λ потоку завдань:

$$\lambda = \sum_{i=1}^{M} \lambda_i \ ; \qquad (5.1)$$

- число θ_i процесорних операцій при виконанні типового завдання:

$$\theta = \frac{1}{\lambda} \sum_{i=1}^{M} \lambda_i \theta_i \ ; \qquad (5.2)$$

- число D_j звертань до файлу F_j для типового завдання:

$$D_i = \frac{1}{\lambda} \sum_{i=1}^{M} \lambda_i N_{ij} \ , \qquad j = 1, \ldots , N; \qquad (5.3)$$

- число D звертань до файлів для типового завдання:

$$D = \sum_{j=1}^{N} Dj \ , \qquad (5.4)$$

- імовірність p_j використання файлу F_j:

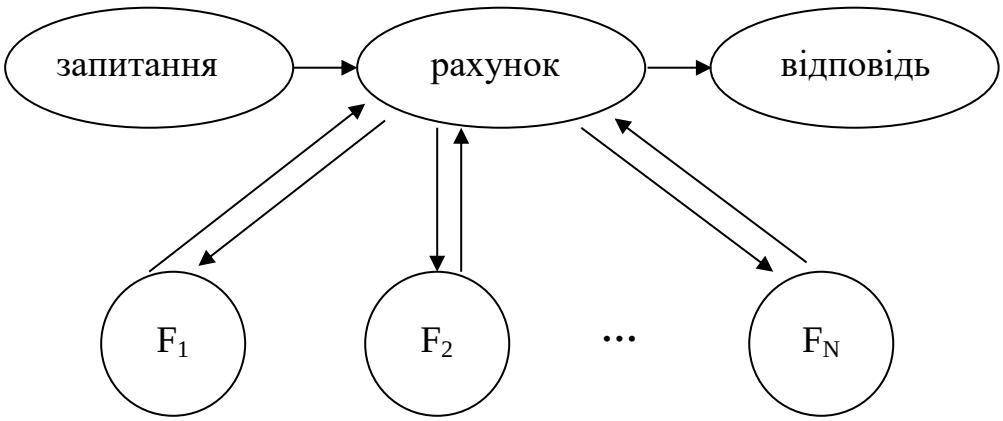
$$p_j = D_j / D \ , \quad j = 1, \ldots , N \ , \qquad (5.5)$$

- середня трудомісткість $\theta_{про}$ етапу рахунку:

$$\theta_{про} = \theta / (D + 1), \qquad (5.6)$$

де:

$(D + 1)$ - кількість етапів рахунку при виконанні типового завдання.



Звертання до файлів

Рисунок 5.1

Таблиця 5.1

Інтенсивність λ_i надходження завдань, $с^{-1}$

Примітка: у таблиці наведено значення $\lambda \cdot 100$

№	№	λ_i	№	λ_i	№	λ_i	№	λ_i	№	λ_i

вар.	задачі		задачі		задачі		задачі		задачі	
1	**1**	**1**	**20**	**2**	**4**	**1**	**16**	**1**	**10**	**1**
2	**2**	**9**	**19**	**3**	**5**	**3**	**15**	**1**	**12**	**1**
3	**3**	**8**	**18**	**4**	**6**	**1**	**14**	**2**	**11**	**1**
4	**4**	**7**	**17**	**5**	**7**	**1**	**13**	**1**	**9**	**2**
5	**5**	**6**	**16**	**6**	**8**	**2**	**12**	**2**	**19**	**1**
6	**6**	**5**	**15**	**7**	**9**	**2**	**20**	**2**	**2**	**2**
7	**7**	**4**	**14**	**8**	**10**	**1**	**19**	**1**	**1**	**4**
8	**8**	**3**	**13**	**9**	**1**	**3**	**1**	**1**	**10**	**2**
9	**9**	**2**	**12**	**10**	**2**	**5**	**17**	**1**	**6**	**1**
10	**10**	**1**	**11**	**11**	**3**	**4**	**16**	**2**	**9**	**1**
11	**11**	**1**	**10**	**2**	**14**	**2**	**15**	**2**	**8**	**3**
12	**12**	**9**	**9**	**3**	**15**	**6**	**13**	**1**	**4**	**1**
13	**13**	**8**	**8**	**4**	**16**	**3**	**12**	**3**	**3**	**2**
14	**14**	**7**	**7**	**5**	**17**	**2**	**11**	**4**	**2**	**2**
15	**15**	**6**	**6**	**6**	**18**	**3**	**10**	**1**	**1**	**4**
16	**16**	**5**	**5**	**7**	**19**	**2**	**9**	**1**	**12**	**5**
17	**17**	**4**	**4**	**8**	**20**	**1**	**8**	**3**	**11**	**5**
18	**18**	**3**	**3**	**9**	**11**	**3**	**7**	**3**	**13**	**3**
19	**19**	**2**	**2**	**10**	**12**	**5**	**6**	**2**	**15**	**3**
20	**20**	**1**	**1**	**11**	**13**	**5**	**5**	**3**	**9**	**2**
21	**1**	**10**	**20**	**2**	**14**	**6**	**10**	**1**	**8**	**3**
22	**2**	**9**	**19**	**3**	**15**	**5**	**9**	**2**	**7**	**4**
23	**3**	**8**	**18**	**4**	**16**	**1**	**8**	**1**	**11**	**7**
24	**4**	**7**	**17**	**5**	**10**	**2**	**7**	**3**	**13**	**6**
25	**5**	**6**	**16**	**6**	**11**	**4**	**6**	**4**	**3**	**4**
26	**6**	**5**	**15**	**7**	**12**	**1**	**5**	**1**	**2**	**3**
27	**7**	**4**	**14**	**8**	**13**	**3**	**16**	**2**	**1**	**2**
28	**8**	**3**	**13**	**9**	**14**	**1**	**15**	**1**	**12**	**3**
29	**9**	**2**	**12**	**10**	**15**	**1**	**13**	**3**	**11**	**4**
30	**10**	**1**	**11**	**11**	**16**	**2**	**12**	**4**	**13**	**1**

Таблиця 5.2

Середнє число θ_i процесорних операцій (десятки мільйонів) і середні числа N_{ij} звертань до файлів

№ зада-чі	θ_i, дес. млн	N_{ij}									
		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
1	1	10	5	-	-	-	-	2	1	-	-
2	2	-	8	5	3	-	-	-	-	3	-
3	3	-	-	10	-	5	-	-	-	-	2
4	4	12	4	-	-	-	3	-	2	-	-
5	5	-	15	8	-	6	-	4	-	-	-
6	6	8	-	8	-	-	7	-	-	3	1
7	7	10	-	-	5	-	-	1	-	2	-
8	8	-	12	6	-	8	-	-	2	-	2
9	9	10	5	-	9	-	-	-	-	-	3
10	10	-	15	-	-	-	10	3	-	4	-
11	1	12	-	8	10	-	-	-	2	2	1
12	2	15	10	-	-	8	-	1	-	1	-
13	3	-	20	5	-	-	8	-	4	-	-
14	4	5	-	15	7	-	-	2	-	3	-
15	5	-	10	20	-	-	10	-	4	-	3
16	6	-	15	25	6	4	-	3	-	2	-
17	7	30	10	-	8	-	10	-	-	-	5
18	8	20	-	25	-	12	-	-	-	4	1
19	9	-	40	-	15	-	-	4	-	-	2
20	10	30	-	20	-	10	5	-	8	-	-

5.3.2. РОЗРАХУНОК МІНІМАЛЬНОЇ ШВИДКОДІЇ ПРОЦЕСОРА

Розглядаючи процесор як систему масового обслуговування, неважко одержати наступну елементарну оцінку його швидкодії *B* для обслуговування завдань без утворення черги. Відомо, що для цього повинна виконуватись умова:

$$\rho < 1,$$

де:

ρ - завантаження процесора потоком завдань,

$\rho = \rho_1 + ... + \rho_м$, i $\rho_i = \lambda_i \theta_i / B$, $i = 1, ... , M$.

Звідси одержимо:

$$B > B^* ,$$

де:

$$B^* = \sum_{i=1}^{M} \lambda_i \theta_i \qquad (5.7)$$

B^* – нижня оцінка необхідної швидкодії. Помітимо, що величину B^* можна обчислити, використовуючи (5.2), як:

$$B^* = \lambda \theta \qquad (5.8)$$

Величини λ і θ визначено раніше за формулами (5.1) і (5.2).

Величина B^* є мінімально необхідною швидкодією, при якій у системі ще можливий стаціонарний режим. При цьому час ω перебування завдання у системі, хоча і є кінцевим, однак може приймати дуже великі значення. Однак, цей час не повинен перевищувати деяку граничну задану величину ω^*. Тоді має місце співвідношення: $B > B(\omega^*)$,

де:

$$B(\omega^*) = \frac{B^*}{2} + \sqrt{\frac{B^{*2}}{4} + \frac{1}{\omega^*} \sum_{i=1}^{M} \lambda_i \theta_i^2} \ . \qquad (5.9)$$

Тоді B^* визначають за формулами (5.7) чи (5.8). З (5.9) випливає, що при $\omega^* \to \infty$ величина $B(\omega^*)$ дорівнює B^*. Водночас при будь-якому кінцевому ω^* завжди $B(\omega^*) > B^*$. Зазвичай для швидкої оцінки B використовують графік функції $B(\omega^*)$, вибираючи B при заданому ω^* і заокруглюючи отримане значення у велику сторону з точністю до мільйонів оп/с.

5.4. ЗМІСТ ЗВІТУ

Постановка задачі, розрахункові дані, графік функції $B(\omega^*)$ і висновки.

ЛАБОРАТОРНА РОБОТА №6

ТЕМА: Побудова стохастичних мережних моделей системи оперативної обробки.

МЕТА: Оволодіння навичками розрахунку обчислювальної системи мінімальної конфігурації, що обробляє потік завдань в оперативному режимі.

6.2. ПОРЯДОК ВИКОНАННЯ РОБОТИ:

6.2.1. Вивчити розділ 6.3 дійсних вказівок.

6.2.2. Використовуючи розрахункові дані лабораторної роботи №5, визначити склад та структуру обчислювальної системи мінімальної конфігурації і побудувати відповідну стохастичну мережеву модель системи.

6.2.3. Оформити звіт по роботі (див. п.6.4)

6.3. ПОБУДОВА ОБЧИСЛЮВАЛЬНОЇ СИСТЕМИ ОПЕРАТИВНОЇ ОБРОБКИ МІНІМАЛЬНОЇ КОНФІГУРАЦІЇ

6.3.1. Основні положення

Найважливішою характеристикою системи оперативної обробки (СОО) є час реакції системи – середній час між моментом надходження завдання на обробку і моментом видачі відповіді користувачу. Для оцінки часу реакції, а також інших характеристик СОО використовують різні моделі, серед яких найбільше поширення отримала модель у вигляді розімкнутої стохастичної мережі масового обслуговування (надалі – просто мережі).

Подібна мережа може бути відображена графом, у якому вузлами є деякі системи масового обслуговування (СМО) S_i, а дугами показують зв'язки між СМО (рис. 6.1).

Кожна СМО може представляти окремий пристрій СОО (наприклад, процесор в однопроцесорній системі) або сукупність однотипних пристроїв (наприклад, нагромаджувачів на магнітних дисках). В останньому випадку говорять про багатоканальної СМО з загальною чергою (рис. 6.2), де кожному пристрою j відповідає обслуговуючий прилад $П_j$, а вимоги (тобто, завдання) вибирають з загальної черги О відповідно до дисципліни обслуговування.

У мережі повинні бути відображені тільки ті пристрої, які найбільше впливають на процес виконання завдання користувача. До цих пристроїв відносять процесор (Пр), селекторні канали (СК) і нагромаджувачі, які використовують у складі зовнішньої пам'яті системи: на магнітних дисках (НМД) і магнітних стрічках (НМС). Кожен з пристроїв реалізує деякі етапи виконання завдання: процесор виконує етапи рахунку, а нагромаджувачі разом зі СК – етапи взаємодії з файлами. Технічно етап звертання до файлу складається з двох фаз: підготовчу і передачі інформації. Підготовча фаза, чи фаза доступу, складається з механізму доступу на заданий циліндр – для НМД, і у підведенні носія – для НМС. Фаза передачі інформації – це безпосередньо обмін даними між оперативною і зовнішньою пам'яттю через

СК. У моделі СОО зручно вважати, що фаза доступу виконується самим нагромаджувачем без участі СК, а фаза передачі інформації – тільки селекторним каналом.

З врахуванням цих припущень найпростішу модель СОО можна представити у вигляді мережі (рис. 6.1). У цій мережі вузол S_0 є джерелом вимог (завдань), інтенсивність λ надходження яких не залежить від кількості завдань, що вже знаходяться у СОО (це справедливо в реальних системах, де час міркування користувача є набагато більшим від часу реакції системи).

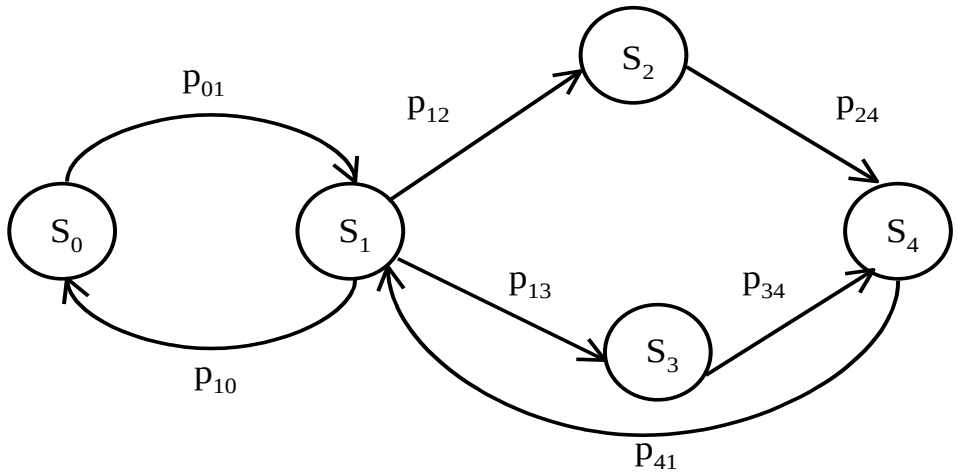


Рисунок 6.1

Потік завдань є найпростішим з відомим значенням λ. Вузол S_1 - це процесор (або сукупність процесорів у мультипроцесорній системі), який, сприйнявши завдання від джерела S_0, виконує етап рахунку. Потім з ймовірністю p_{10} вимога повертається до джерела (задача вирішена), з

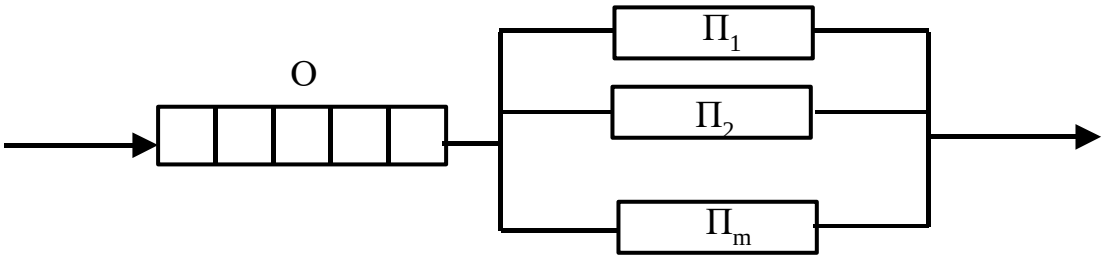


Рисунок 6.2

ймовірністю p_{12} направляється у вузол S_2, що є сукупністю НМД, або з ймовірністю p_{13} – у вузол S_3 (сукупність селекторних каналів), після чого повертається у процесор S_1. Очевидно, у цій інтерпретації у мережі (рис. 6.1) $p_{01} = p_{24} = p_{34} = p_{41} = 1$. Кожний з вузлів $S_1 ... S_4$ може бути СМО, зображеної на рис. 6.2. Дисципліна вибору вимог з черги для всіх СМО – найпростіша (у порядку надходження).

Задачею даної лабораторної роботи є визначення мінімального складу (кількості пристроїв) СОО, необхідного для обробки потоку завдань, розрахованих у лабораторній роботі №5.

Ця задача вирішується в декілька послідовних етапів, опис яких наведено у п.п. 6.3.2-6.3.4.

6.3.2. Розміщення файлів у нагромаджувачах зовнішньої пам'яті

Мета даного етапу – раціональне розміщення файлів даних, визначених відповідним варіантом (л.р. №5), у нагромаджувачах НМД і НМС. Кожен тип нагромаджувача характеризується своїм часом доступу $\nu_Д$ і ν_c відповідно, причому $\nu_c >> \nu_Д$.

Нагромаджувач з розміщеним у ньому файлом F_j можна розглядати як деяку СМО, у яку надходить потік запитів з інтенсивністю λ_j, і кожен такий запит обслуговується в середньому за час ν_j. Тоді умова існування стаціонарного режиму даної СМО:

$$\lambda_j \nu_j < 1. \qquad (6.1)$$

Інтенсивність λ_j оцінюють у такий спосіб. З попередньої лабораторної роботи нам відомі середня кількість D_j звертання до файлу F_j при виконанні типового завдання. Якщо λ – інтенсивність потоку завдань у СОО, тоді: $\lambda_j = \lambda D_j$. Підставляючи це значення у (6.1) відносно ν_j, одержимо $\nu_j < \nu_j^*$, де величина:

$$\nu_j^* = 1/\lambda \qquad (6.2)$$

може розглядатися як максимально допустимий час звертання до файлу F_j. Таким чином, файл F_j може розміщатися в розглянутому нагромаджувачі, якщо тільки час доступу цього нагромаджувача є меншим від ν_j^*.

Звідси випливає наступний алгоритм розміщення файлів. Для кожного файлу F_j, $j=1,...,N$, знаходять величину ν_j^* за формулою (6.2). Далі перевіряють умову: $\nu_Д > \nu_j^*$. Якщо вона виконується, тоді файл розміщують в НМД. У іншому випадку перевіряється умова $\nu_Д < \nu_j^* \le \nu_c$. Якщо вона виконується, тоді файл розміщують в НМС. Якщо ж і вона не виконується (тобто, $\nu_c \ge \nu_j^*$), тоді даний файл не може бути повністю розміщений у НМС, і його необхідно розділити на деяку кількість n_j частин. n_j вибирають з формули (6.1) за умови стаціонарності: $\lambda_j \nu_Д / n_j < 1$. Звідси: $n_j > n_j^*$, де:

$$n_j^* = \lambda D_j \nu_Д. \qquad (6.3)$$

Значення n_j вибирають як найближче до n_j^* зверху натуральне число.

Величини $\nu_Д$ і ν_C для кожного варіанта задані в таблиці 6.1.

Таблиця 6.1

Номер варіанту	Середній час доступу до даних (с)		Швидкість передачі даних (Кбайт/с)		Емкість накопичувача (Мбайт)	
	НМД	НМС	НМД	НМС	НМД	НМС
1	0,05	1,0	200	100	4,0	10
2	0,06	1,5	190	90	5,0	12
3	0,07	2,0	180	80	5,5	15
4	0,08	2,5	170	75	6,0	20
5	0,09	3,0	160	70	6,5	22
6	0,10	3,0	150	60	7,7	25
7	0,11	2,5	140	55	7,5	30
8	0,12	2,0	130	50	8,0	25
9	0,13	1,5	120	40	8,5	22
10	0,14	1,0	110	45	9,0	20

6.3.3. Визначення параметрів мінімальної конфігурації СОО

Мінімальною називається конфігурація СОО, у якій існує стаціонарний режим при обробці завдань, і всі необхідні для роботи файли розміщені в нагромаджувачах зовнішньої пам'яті.

Найважливішими параметрами СОО є: швидкодія процесора, кількість НМД, кількість НМС і кількість селекторних каналів. Швидкодію процесора уже визначено у роботі №5. Обчислимо кількість НМД, НМС і СК.

При визначенні кількості нагромаджувачів зовнішньої пам'яті необхідно задовольнити умови існування стаціонарного режиму і повного розміщення файлів у нагромаджувачах одночасно.

Розглянемо спосіб оцінки необхідної кількості, m_c, НМС. Сукупність НМС можна розглядати як m_c - канальну СМО, на вхід якої надходить потік вимог з інтенсивністю λ_c. У цьому випадку обслуговується кожну вимога в середньому за час ν_c. Стаціонарний режим існує, якщо:

$$\lambda_c \nu_c < 1.$$

Інтенсивність λ_c звертань до файлів, розміщених у НМС, дорівнює:

$$\lambda_c = Dp_c\lambda,$$

де: p_c - ймовірність звертання до “стрічкового” файлу,

$$p_c = \sum_{F_j \in C} p_j,$$

і С - кількість “стрічкових” файлів, тобто файлів, розміщених на НМС. Звідси випливає обмеження на кількість НМС:

$$m_c \geq \lambda D \nu_c \sum_{F_j \in C} p_j . \qquad (6.4)$$

Крім (6.4), необхідно ще задовольнити умову, при якій сумарна ємність НМС, що використовують у СОО, не є меншою від необхідної сумарної довжини стрічкових файлів, тобто:

$$m_c \geq \sum_{F_j \in C} p_j G_j / G_c, \qquad (6.5)$$

де:

G_j - довжина файлу F_j,

G_c - ємність однієї НМС.

Умови (6.4) і (6.5) задовольняються одночасно, якщо за m_c вибрати величину:

$$m_c = \max\left\{\left[\lambda D \nu_c \sum_{F_j \in C} p_j\right], \left[\sum_{F_j \in C} p_j G_j / G_c\right]\right\}. \qquad (6.6)$$

Міркуючи аналогічно, можна одержати і формулу для визначення кількості $m_Д$ нагромаджувачів на дисках:

$$m_Д = \max\left\{\left[\lambda D \nu_Д \sum_{F_j \in Д} p_j\right], \left[\sum_{F_j \in Д} p_j G_j / G_Д\right]\right\}, \qquad (6.7)$$

де:

Д - файли, розміщені на НМД, $G_Д$ - ємність одного НМД.

Обчислимо кількість m_K селекторних каналів СОО. СМО характеризується інтенсивністю $\lambda_к$ надходження вимог і середнім часом $\nu_к$ їх обслуговування. Тоді умова стаціонарності:

$$\lambda_к \nu_к / m_K < 1. \qquad (6.8)$$

Очевидно, що $\lambda_к = \lambda_с + \lambda_Д = D\lambda$. Тоді $\nu_к$ можна визначити усередненням за часом передачі даних у НМС і НМД:

$$\nu_к = \frac{g_с}{\nu_с} p_с + \frac{g_Д}{\nu_Д} p_Д, \qquad (6.9)$$

де:
$g_с$ і $\nu_с$ - середня довжина запису і швидкість передачі даних у НМС;
$g_Д$ і $\nu_Д$ - ті ж величини для НМД.

Середні довжини $g_С$ і $g_Д$ можна визначити як:

$$g_с = \sum_{F_j \in С} g_j \frac{p_j}{p_с}, \quad g_Д = \sum_{F_j \in Д} g_j \frac{p_j}{p_Д}. \qquad (6.10)$$

Підставляючи (6.10) у (6.9) і виконуючи нерівність (6.8) відносно $m_К$, одержимо:

$$m_k > \lambda D \left(\frac{1}{\nu_С} \sum_{F_j \in С} g_j p_j + \frac{1}{\nu_Д} \sum_{F_j \in Д} g_j p_j \right). \qquad (6.11)$$

Отже, як $m_К$ можна взяти найближче до правої частини нерівності (6.11) зверху ціле число.

Величини $\nu_с$, $\nu_Д$, $G_с$ і $G_Д$ вибирають для розрахунку з таблиці 6.1 відповідно до заданого варіанту, довжини G_j файлів і записів g_j у них – з таблиці 6.2.

Таблиця 6.2

Файли	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
Довжина файлу (Мбайт)	0.5	1.0	1.0	1.5	1.5	2.0	2.5	3.0	4.0	5.0
Середня довжина запису (Кбайт)	5	8	15	6	14	18	10	15	20	25

6.3.4. Побудова стохастичної мережевої моделі СОО мінімальної конфігурації

Стохастична мережева модель СОО вважається побудованою тоді, коли визначена структура мережі і наступні параметри:

- інтенсивність λ вхідного потоку вимог;
- матриця $p = (p_{ij})$ ймовірностей передач;
- кількість m_i обслуговуючих каналів кожної СМО S_i, що входять у мережу;
- трудомісткість θ_i однократного обслуговування у СМО S_i;
- швидкодія V_i одного каналу у СМО S_i.

У нашому випадку структура мережі визначається з рис. 6.1. Тобто:

S_0 - джерело вимог, інтенсивність λ якого визначена у л. р. №5;

S_1 - процесор СОО;

S_2 і S_3 - відповідно сукупність НМД і НМС;

S_4 - сукупність селекторних каналів.

Матриця p ймовірностей передач для такої мережі має вигляд:

$$P = \begin{array}{c} \\ S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{array} \begin{array}{c} \begin{array}{ccccc} S_0 & S_1 & S_2 & S_3 & S_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ p_{10} & 0 & p_{12} & p_{13} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Обчислимо ймовірності p_{10}, p_{12} і p_{13}. Ймовірність p_{10} - це ймовірність завершення виконання завдання на черговому етапі рахунку. Вважаючи, що завершення виконання типового завдання може відбутися на кожному з (D+1) етапів рахунку з рівною ймовірністю, одержимо $p_{10} = 1/(D+1)$. Якщо рахунок продовжується (що відбувається з ймовірністю $1-p_{10} = D/(D+1)$), тоді з ймовірністю $p_Д$ за етапом рахунку випливає звертання до НМД, або з ймовірністю p_c - до НМС. Отже, $p_{12} = p_Д D/(D+1)$, $p_{13} = p_c/(D+1)$.

Параметри (m_i, ν_i) систем S_i мережі наступні. У системі S_1 (процесор): $m_1 = 1$, $\nu_1 = \theta_1/\nu_1$, де θ_1 – трудомісткість одного етапу рахунку типового завдання, ν_1 - швидкодія процесора. Ці параметри визначено у л.р. №5. У системах S_2, S_3 і S_4 $m_2 = m_Д$, $m_3 = m_c$, $m_4 = m_K$, а середні часи обслуговування рівні відповідно $\nu_2 = \nu_Д$, $\nu_3 = \nu_c$ і $\nu_4 = \nu_K$.

6.4. ЗМІСТ ЗВІТУ

Постановка задачі, порядок і результати розрахунків за п.п. 6.3.2-6.3.4, структура СОО мінімальної конфігурації, висновки.