

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України «Київський політехнічний
інститут»

Методичні вказівки
до виконання домашніх контрольних робіт з курсу
КОМП'ЮТЕРНІ СИСТЕМИ
для студентів напрямку підготовки 6.050102
“Комп'ютерна інженерія”
заочної форм навчання

Рекомендовано
кафедрою обчислювальної техніки
Протокол № від 2012 р.
Завідувач кафедри

_____ Луцький Г.М.
(підпис)

Київ 2012

МОДЕЛЮВАННЯ ЧАСОВИХ ХАРАКТЕРИСТИК КОМП'ЮТЕРНИХ СИСТЕМ

Загальні теоретичні відомості

Перед тим, як спроектувати обчислювальну систему, аналізують, які функції буде виконувати дана система, які прикладні задачі вона буде вирішувати. Після визначення кола задач, що вирішуються, оцінюють трудомісткість алгоритмів. Поняття “трудомісткість алгоритмів” використовують для оцінки потреб у часі, пов'язаних з періодом роботи сукупності пристроїв обчислювальних систем (ОС).

Трудомісткість алгоритмів – це кількість обчислювальної роботи, яка необхідна для реалізації цих алгоритмів на обчислювальній системі. Трудомісткість оцінюється кількістю операцій, що виконуються з метою обробки вводу та виводу інформації в процесі розв'язування задачі.

Інформація, що стосується задачі, поділяється на програму та дані. Останні містять в собі вихідні дані, проміжні величини та кінцеві результати. Дані можуть бути розташовані як в оперативній, так і в зовнішній пам'яті.

Оперативна та зовнішня пам'ять мають відмінності у методах доступу, швидкісних характеристиках. Затрати часу на звернення до оперативної пам'яті пропорційні кількості інформації, що читається або записується. Витрати часу на звернення до зовнішньої пам'яті в основному визначаються часом пошуку області даних та в меншому ступені залежать від кількості інформації, що передається. Тому для зменшення витрат часу на обмін інформації з зовнішньою пам'яттю за кожне звернення доцільно передавати достатньо велику кількість інформації. Операція звернення до зовнішньої пам'яті розглядається зазвичай як операція вводу-виводу.

Складність обчислень оцінюється трудомісткістю, яка, в свою чергу, визначається кількістю операцій, що виконуються з метою обробки, вводу та виводу інформації в процесі розв'язку задачі. Трудомісткість алгоритму – величина ймовірна, оскільки число операторів, що виконуються при різних вхідних даних, може бути різною величиною. Саме тому повна характеристика трудомісткості передбачає опис кількості операцій, що виконуються за одну реалізацію алгоритму, випадковими величинами.

Використання статичних методів визначення кількості операцій – складний та довготривалий процес, тому трудомісткість алгоритмів звичайно описується методами математичної статистики, наприклад, за допомогою математичного опису числа операцій, що виконуються.

При вирішенні задач аналізу та синтезу ОС виникає необхідність в описі властивостей обчислювальних процесів. Обчислювальний процес подають у вигляді моделі, яка несе в собі інформацію тільки про властивості самого процесу. З урахуванням відомостей про трудомісткість алгоритму можна сформулювати такі вимоги до моделі, яка повинна:

- 1) визначати порядок проходження алгоритмів запитів на кожен з видів обслуговування – рахування та ввід-вивід інформації;
- 2) визначати трудомісткість обслуговування запитів – кількість операцій, яку повинен виконати процесор при обслуговуванні запиту на рахування, та кількість символів інформації, що вводиться або виводиться;
- 3) відповідати реальним процесам з точністю до збігу, як найменше, математичних очікувань їх однойменних характеристик.

На початковому етапі побудови моделі необхідно оцінити трудомісткість алгоритму [4].

При оцінці трудомісткості обчислювальний процес можна подати у вигляді послідовності операторів, при цьому розрізняють три типи операторів: функціональні, переходу та оператори звернення до файлів. Функціональні оператори задають сукупність обчислювальних операцій. Оператори переходу задають правила вибору одного з можливих шляхів розвитку обчислювального процесу. Оператори звернення до файлів відображують процес обміну інформацією із зовнішніми пристроями. Для оцінки трудомісткості алгоритму необхідно розбити множини операторів на два класи: основні оператори, які об'єднують в собі функціональні оператори та оператори переходу (позначимо V_a) $S_O = \{V_{a1}, \dots, V_{am}\}$; оператори звернення до файлів (позначимо V_b) $S_H = \{V_{b1}, \dots, V_{bn}\}$. Трудомісткість алгоритму можна охарактеризувати такою сукупністю параметрів:

Q – середня кількість процесорних операцій, які виконуються за одну реалізацію алгоритму; N_1, \dots, N_h – середній час звернення до файлів F_1, \dots, F_h ;

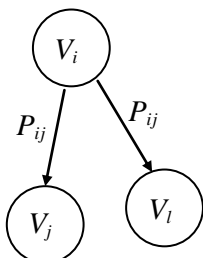


Рис. 1

$$P_{1(k-1)}n_1 + P_{2(k-1)}n_2 + \dots + (P_{(k-1)(k-1)} - 1)n_{k-1} = 0.$$

У виразі (1) підсумовування виконується для всіх вершин, які належать до основних операторів, у виразах (2), (3) – для всіх вершин, що відображають звернення до даного файлу.

При формуванні коефіцієнтів системи (4) необхідно звернути увагу на те, що коефіцієнти P_{ij} записуються не по рядках, а по стовпцях, тобто, наприклад, на місці другого елемента першого рядка стоїть елемент з індексом 21, а не коефіцієнт з індексом 12 матриці переходів.

Визначивши за формулами (1), (2), (3) величини θ , N_h , θ_h можна визначити середню трудомісткість етапу рахування, на основі якої виконується оцінка оптимальної швидкодії процесора.

Середня трудомісткість етапу рахування визначається за формулою: де N – сума середнього числа N_i звернень до основних операторів (S_o), тобто:

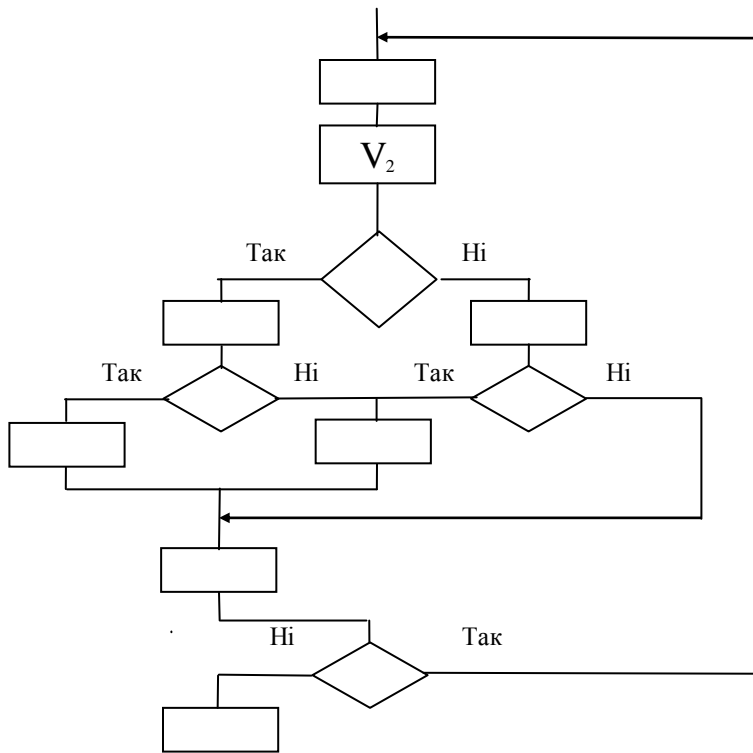
$$\theta_0 = \frac{\theta}{N}, \quad (5)$$

$$N = \sum_{V_{0i}} n_i. \quad (6)$$

Вихідні дані: - схема алгоритму (рис. 3, 4, 5, 6).

Вихідні дані визначаються останньою цифрою залікової книжки. Вона визначає схему алгоритму. При цьому цифрам 0, 1, 2 відповідає схема, наведена на рис. 3, цифрам 3, 4, 5 – схема, наведена на рис. 4, цифрам 6, 7, 8 – схема, наведена на рис. 5, а цифрі 9 – схема, наведена на рис. 6.

Приклад: Реалізацію алгоритму можна подати у вигляді направленого графа, вершини якого відповідають операторам [3]. Ребра графа відмічаються ймовірностями переходів від i -ї вершини до j -ї вершини. Граф наведено на рис. 7.



Для цього графа матрицю ймовірностей переходу задано в табл. 1, елемент P_{ij} якої визначає ймовірність переходу із стану i в стан j .

Рис. 7

Матриця ймовірнісних переходів

Таблиця 1

	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_k
V_1		1						
V_2			0,25	0,75				
V_3					0,5		0,5	
V_4					0,2	0,3		
V_5							1	
V_6							1	
V_7	0,9							0,1

Для простоти припустимо, що всі оператори алгоритму – основні та $k_i=1$ для всіх $i=1, \dots, k$. На підставі табл. 1 та формули (4) складаємо систему з семи лінійних алгебраїчних рівнянь

$$\begin{cases}
 -N_1 + 0,9 N_7 = -1, \\
 N_1 - N_2 = 0, \\
 0,25 N_2 - N_3 = 0, \\
 0,75 N_2 - N_4 = 0, \\
 0,5 N_3 + 0,2 N_4 - N_5 = 0, \\
 0,8 N_4 - N_6 = 0, \\
 0,5 N_3 + N_5 + N_6 - N_7 = 0.
 \end{cases} \quad (7)$$

Розв'язуючи систему (7) знаходимо значення N_1, \dots, N_{k-1} . Підставляючи отримані значення у формулу (1), отримаємо:

$$\theta = \sum_{i=1}^l k_i N_i = 39,75.$$

Якщо при виконанні будь-яких операторів відбувається звернення до файлів, то необхідно визначити ще величини N та θ . Визначивши таким чином ці величини, можна визначити середню трудомісткість етапу рахування.

Вихідні дані:

а) схема алгоритму (рис. 3, 4, 5, 6); б) k_i – кількість операцій, що складають V_{ai} оператор (табл. 2); в) L_i – середня кількість інформації, що передається при виконанні V_i оператора звернення до файлу, де m – номер файлу, до якого відбувається звертання (табл. 3); г) області зміни параметрів X_i та N_i (табл. 4).

Вихідні дані визначають за двома останніми цифрами залікової книжки. Остання цифра залікової книжки визначає область зміни параметрів. Передостання цифра залікової книжки визначає значення k_i та L_i .

Число операцій, що складають V_{ai} оператор (k_i)

Таблиця 2

Кількість операторів V_a	Номер варіанта									
	0	1	2	3	4	5	6	7	8	9
V_{a1}	20	20	50	30	60	20	40	80	30	10
V_{a2}	30	30	40	10	60	100	20	40	60	80
V_{a3}	50	30	20	30	40	60	30	20	100	200
V_{a4}	20	30	50	20	30	30	10	80	90	35
V_{a5}	50	50	30	20	10	50	30	70	20	20
V_{a6}	30	20	10	30	100	30	20	60	70	45
V_{a7}	100	10	20	50	40	20	100	30	40	50
V_{a8}	20	40	10 0	100	20	40	50	300	200	100

Середня кількість інформації, що передається при виконанні V_{bi} оператора звернення (L_i)

Таблиця 3

Кількість інформації V_b	Номер варіанта									
	0	1	2	3	4	5	6	7	8	9
V_{b1}	500	700	800	250	900	250	800	300	500	400
V_{b2}	250	800	100	500	250	1000	100	200	400	300
V_{b3}	120	500	250	150	100	700	500	250	800	500
V_{b4}	800	100	150	1000	700	250	900	200	100	200
V_{b5}	100	600	800	200	500	1000	250	800	200	700
V_{b6}	600	900	700	100	400	400	400	500	900	300
V_{b7}	900	600	900	400	800	900	100	100	600	900
V_{b8}	400	700	600	200	900	400	600	400	400	100

Області зміни параметрів X_i та N_i

Таблиця 4

Параметри	Номер варіанта									
	0	1	2	3	4	5	6	7	8	9
X_1	-1,+3	-2,+3	-2,+3	0,+4	-2,+2	0,+5	0,+6	-1,+4	1,+7	-3,+1
X_2	-1,+1	-1,+4	-2,+2	-3,+1	-3,+2	-2,+4	0,+5	-1,+3	-2,+4	-2,+3
K_1	10	20	30	10	50	10	20	10	20	25
K_2	20	10	15	20	40	20	10	20	10	10
K_3	10	30	10	30	10	30	10	20	10	20

N_i – середнє число попадань обчислювального процесу у стан $S_i (1, \dots, k_i)$.

Хід виконання роботи

1. Вивчити теоретичні відомості.
2. Отримати допуск до роботи.
3. Вибрати за методичними вказівками варіант завдання.
4. Використовуючи дані, визначити:
 - а) середню кількість операцій, яка виконується за один прогін алгоритму;
 - б) середню кількість звернень до кожного з файлів;
 - в) середню кількість інформації, яка передається при одному звертанні до файлу;
 - г) середню трудомісткість етапу рахування.
5. Скласти звіт по роботі.
6. Зробити висновки по роботі.

Примітка. Для зменшення розмірності системи лінійних рівнянь доцільно об'єднувати послідовні ланцюжки операторів в один узагальнений оператор.

