

Лабораторна робота 1

МОДЕЛІ ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

Мета роботи: вивчення моделей обчислювальних процесів.

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 ймовірнісна, 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 множину операторів на два класи: основні оператори, які об'єднують в собі функціональні оператори та оператори переходу (позначимо V_a) $S_o = \{V_{ai}, \dots, V_{am}\}$; оператори звернення до файлів (позначимо V_b) $S_H = \{V_{bi}, \dots, V_{bm}\}$. 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 ; $\theta_1, \dots, \theta_h$ – середня кількість байтів, що передається за одне звернення до файлів F_1, \dots, F_h відповідно. Сукупність операторів та зв'язків між ними бачимо на графі алгоритму (рис. 1), в якому індексами i та j відмічена ймовірність переходу P_{ij} з вершини V_i до вершини V_j . Оскільки обчислювальний процес не може призупинитись в довільній, за винятком кінцевої K , вершині, то з імовірністю 1 відбудеться перехід до будь-якого оператора алгоритму. З урахуванням цього для будь-якого оператора V_i ($i=1, 2, \dots, k-1$) повинна виконуватись умова $P_{ij}=1$. Імовірності P_{ij} залежать від ймовірності виконання умови, що перевіряється оператором i з метою вибору шляху переходу.

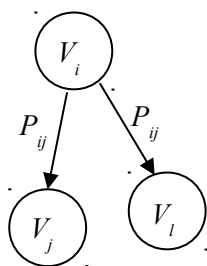


Рис. 1

1.

Наприклад, нехай оператор i породжує перехід до оператора j при від'ємному значенні деякої змінної X та до оператора l при додатному значенні X (рис. 2). Якщо відомо, що величина X рівномірно розподілена в діапазоні $(-1, 3)$, то з імовірністю 0,25 її знак є від'ємним та з імовірністю 0,75 – додатним. З цього витікає, що перехід до оператора j відбувається з ймовірністю $P_{ij}=0,25$ та перехід до оператора l – з імовірністю $P_{il}=0,75$.

Для операторів циклу ймовірність переходу визначається кількістю циклів. Наприклад, оператор циклу повинен виконуватися 9 разів, в цьому випадку ймовірність виходу з циклу $P_{ij}=0,1$, а ймовірність виконання циклу $P_{ij}=0,9$. Якщо за оператором i обов'язково виконується оператор j , то $P_{ij}=1$.



Визначаємо середнє число процесорних операцій, що виконуються при

одному прогоні алгоритму:

Після цього визначаємо середнє число звернень до файлу F_h :

та середню кількість інформації, що передається при одному зверненні до файла F_h :

$$\theta_h = \frac{1}{N_h} \sum_{i \in V} n_i l_i (h = 1, \dots, H), \quad (3)$$

де l_i – середня кількість інформації, яка передається при виконанні оператора i .

Значення N_1, \dots, N_{k-1} визначаються розв'язком системи лінійних алгебраїчних рівнянь, канонічний запис яких має вигляд:

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:

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

де N – сума середнього числа N_i звернень до основних операторів(S_o), тобто:

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

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

Вихідні дані визначаються молодшою цифрою номеру за списком. Вона визначає блок-схему алгоритму. При цьому цифрам 0, 1, 2 відповідає блок-схема, наведена на рис. 3, цифрам 3, 4, 5 – блок-схема, наведена на рис. 4, цифрам 6, 7, 8 – блок-схема, наведена на рис. 5, а цифрі 9 – блок-схема, наведена на рис. 6.

Хід виконання роботи

1. Вивчити теоретичні відомості.
2. Отримати допуск до роботи.
3. Вибрати за методичними вказівками варіант завдання.
4. Занести до звіту блок-схему, яка передбачена завданням.
5. Скласти звіт по лабораторній роботі.
6. Зробити висновки по роботі.
7. Відповісти на запитання для самоперевірки.

Запитання для самоперевірки

1. Дати визначення поняттю “трудомісткість алгоритмів”.
2. Для чого використовують поняття “трудомісткість алгоритму”?
3. У вигляді чого можна подати обчислювальний процес?
4. Сформулювати вимоги до моделі обчислювального процесу.
5. Які розрізняють типи операторів алгоритму?
6. Якими параметрами можна охарактеризувати трудомісткість алгоритму?

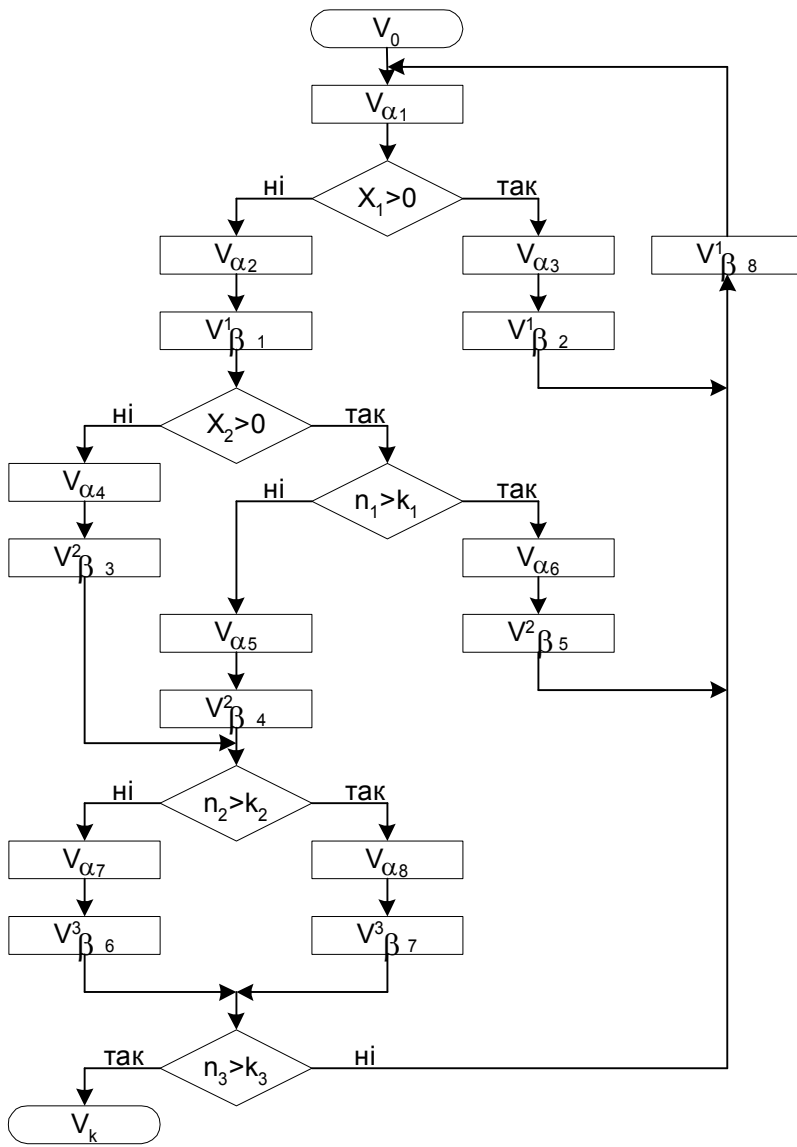


Рис. 3

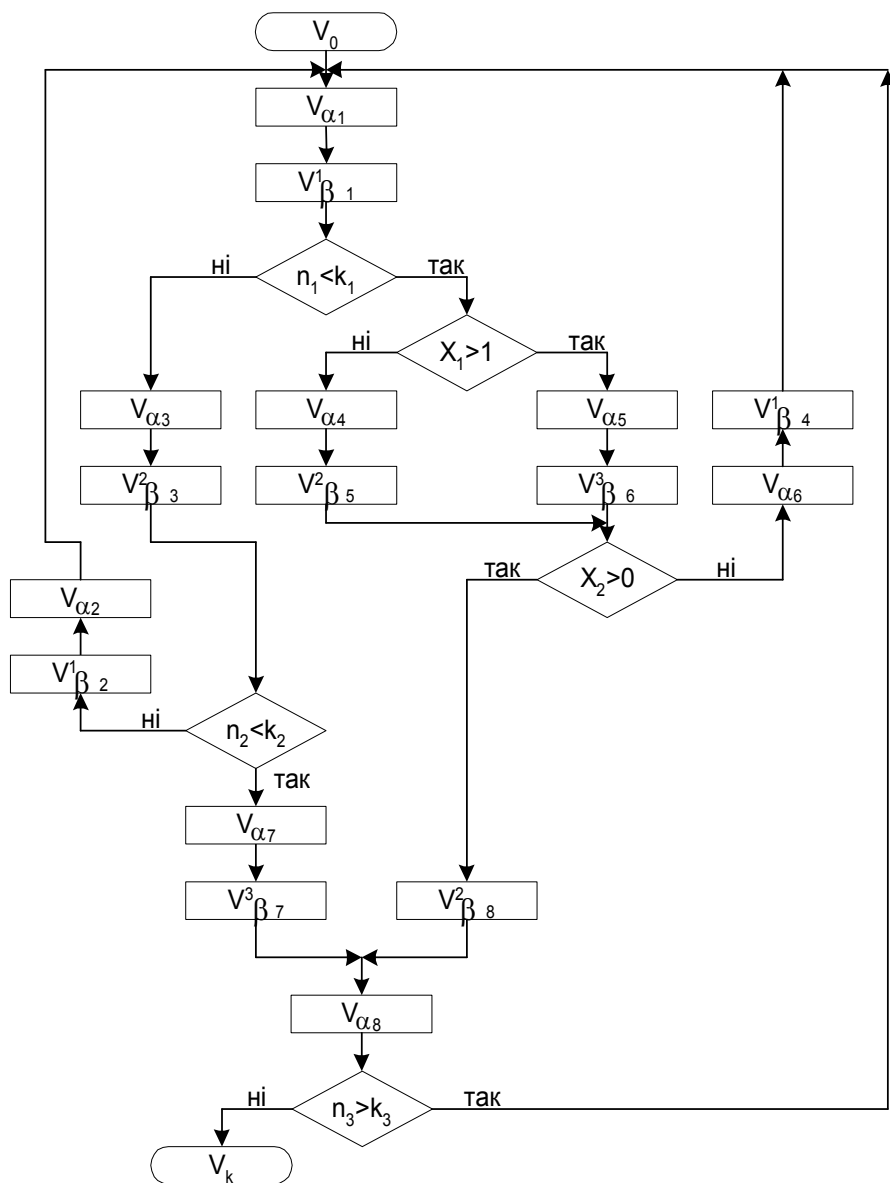


Рис. 4

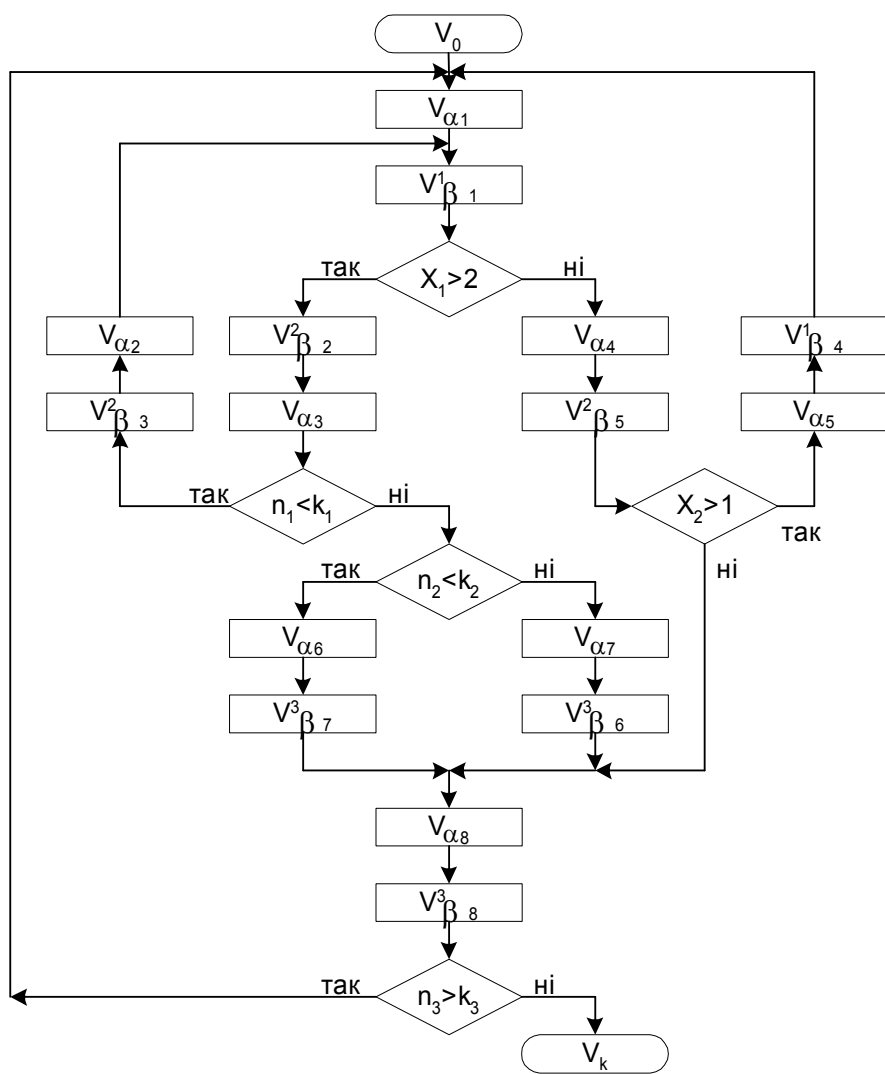


Рис. 5

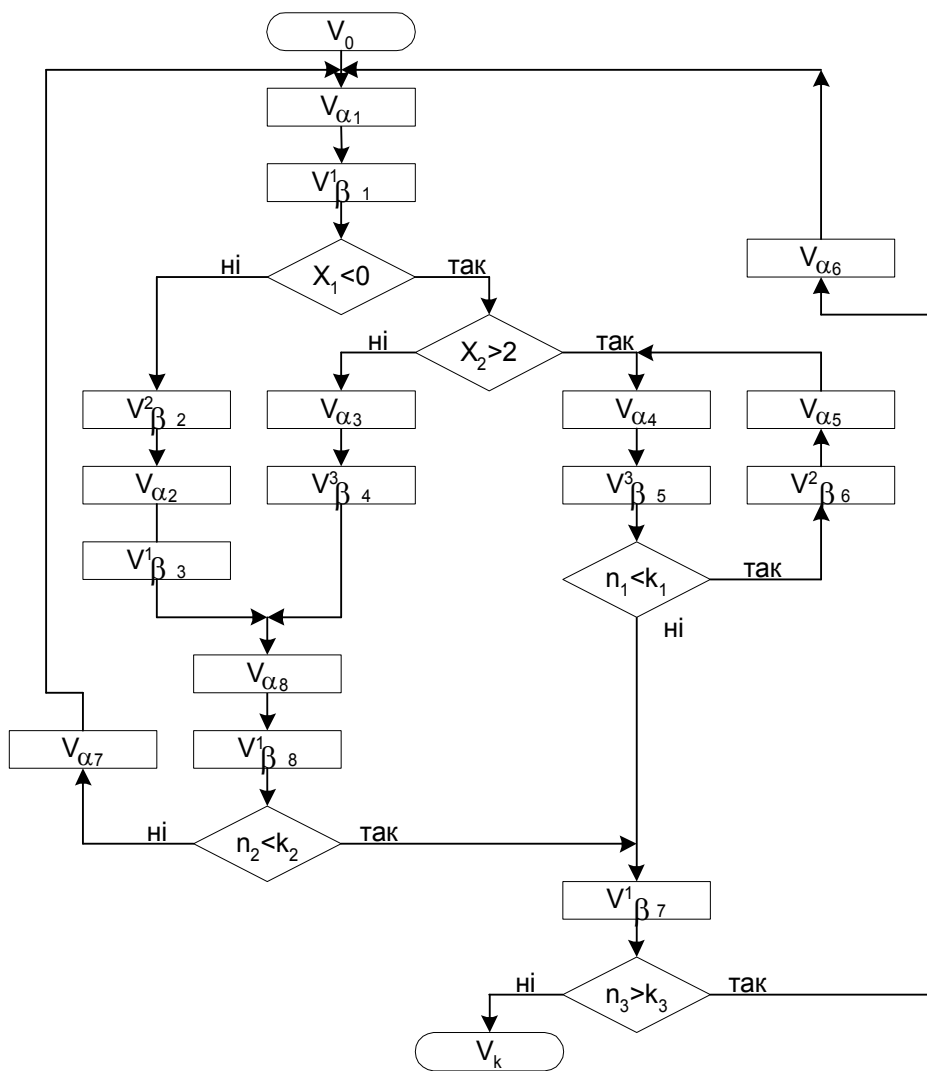


Рис. 6