Звіт до лабораторної роботи №2

З дисципліни “Операційні системи”

Multiple Queue Scheduling

Вступ до багаторівневого планування черги:

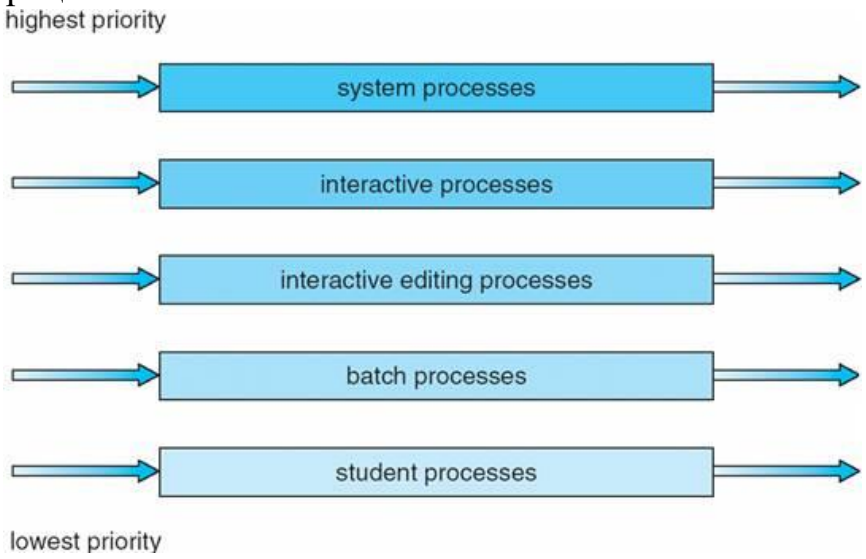
Спільний поділ здійснюється між процесами переднього плану (інтерактивними) і фоновими (пакетними) процесами. Ці два типи процесів мають різні вимоги до часу відгуку, і тому можуть мати різні потреби в плануванні. В додаток, процеси переднього плану можуть мати пріоритет (визначений ззовні) над процесами фонового плану. Алгоритм планування багаторівневої черги розбиває готову чергу на кілька окремих черг. Процеси постійно призначаються до однієї черги, як правило, на основі деяких властивостей процесу, таких як розмір пам’яті, пріоритет процесу або тип процесу.

Кожна черга має свій власний алгоритм планування. Наприклад, окремі черги можуть використовуватися для процесів переднього та фонового плану. Черга переднього плану може бути запланована за допомогою алгоритму FCFS, тоді як фонова черга планується за допомогою алгоритму Round-Robin.

Крім того, серед черг має бути планування, яке зазвичай реалізується як фіксоване- пріоритетне випереджувальне планування (fixed-priority preemptive scheduling). Наприклад, черга переднього плану може мати абсолютний пріоритет над чергою фону.

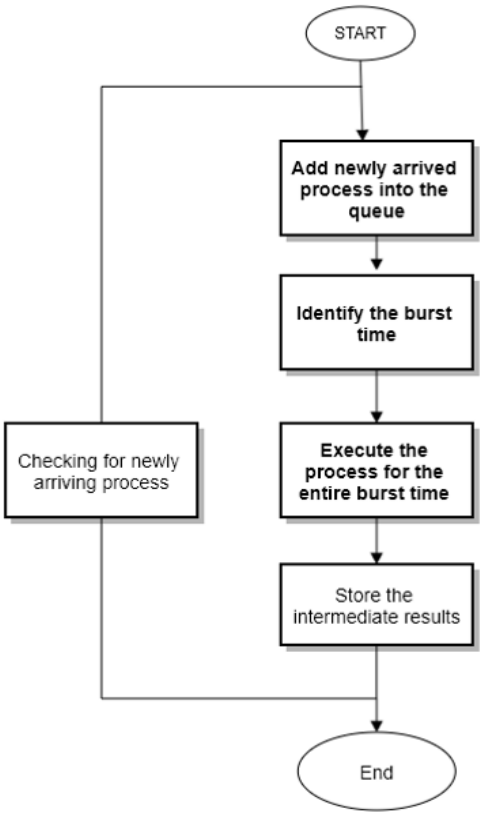
Давайте розглянемо приклад багаторівневого алгоритму планування черги з п’ятьма чергами, наведеними нижче в порядку пріоритету:

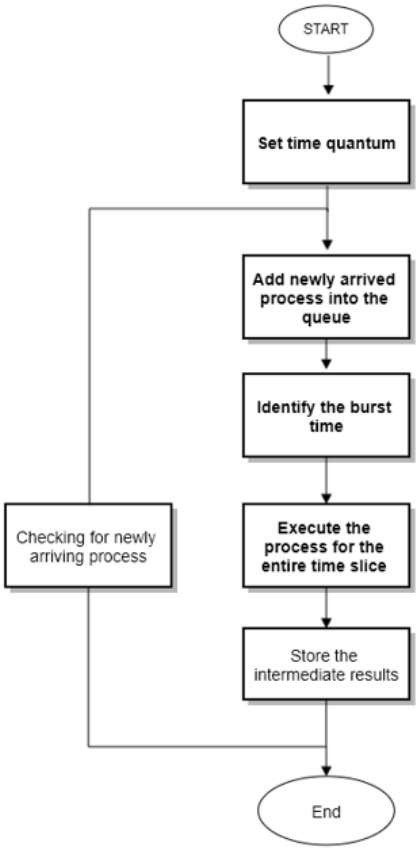
1. Системні процеси
2. Інтерактивні процеси
3. Інтерактивні процеси редагування
4. Пакетні процеси
5. Студентські процеси

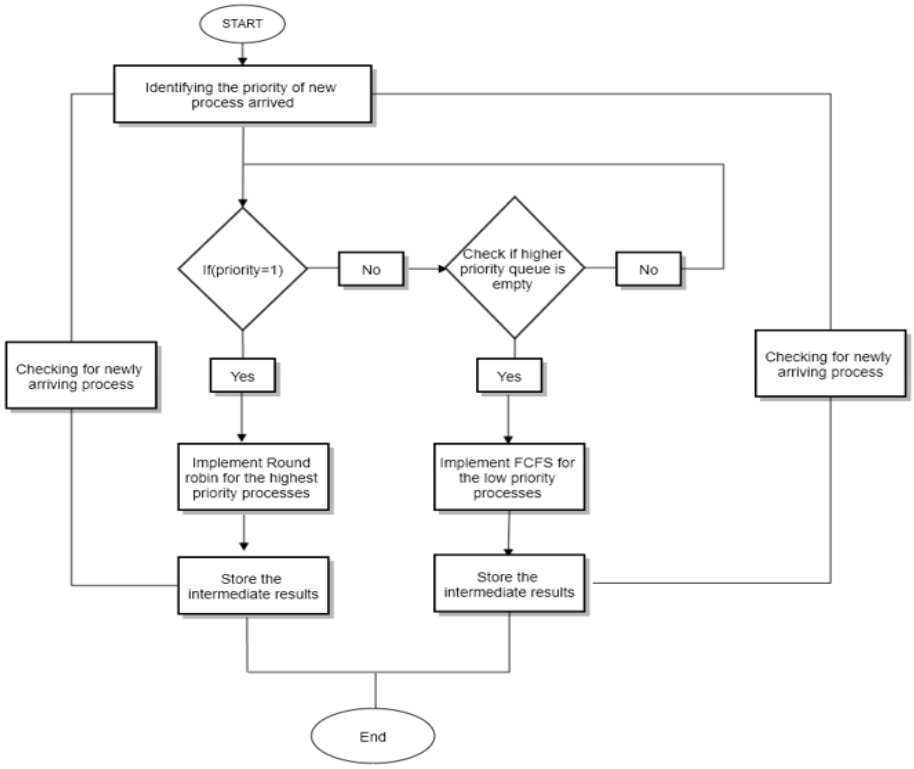


Блок-схеми:

FCFS, Round-Robin та MultipleQueue планування черги без випередження:





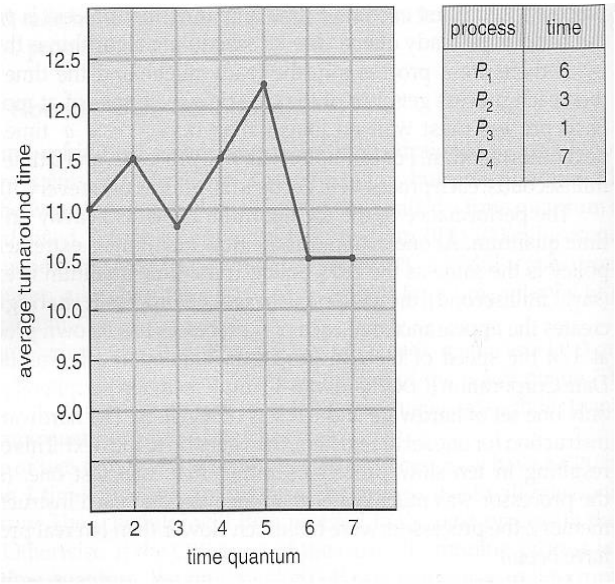


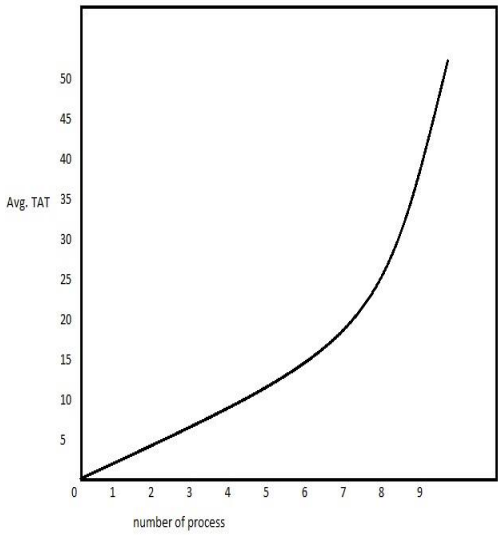
Аналіз складності коду:

- Для кожного циклу for складність обчислюється через n.
- Для кожного вкладеного циклу for загальна складність обчислюється в термінах n^m.

- Для кожного оператора if він множиться на 1.
- Складність всього коду становить (4n (1)) ^2.
- Час виконання коду: 2749 мілісекунд = 0.

Складність залежить від пріоритетності кожного процесу. Він буде динамічним і змінюється для високого та низького пріоритету процесу.





Попередньо ініціалізовані умови (для критеріїв обслуговування):

- Квант часу безпосередньо ініціалізується на 4 в коді. Його можна змінити, якщо спливає будь-який інший квантовий запит.
- Усі масиви, такі як час завершення, час очікування, час пакету, ініціалізуються 500.
- Час прибуття приймається за 0 для всього процесу.
- Пріоритет містить 0 і 1. 1 вказує на високий пріоритет, а 0 — на низький пріоритет.

Доказ того, що алгоритм коректний:
Algorithm:

- BurstTime і Priority кожного процесу були прочитані з .xls файлу та збережені в два масиви розміром 500.
- Процеси були розділені за пріоритетом.
- Процеси з високим пріоритетом були відправлені до 1 черги, а процеси з низьким пріоритетом були відправлені до 2 черги.
- FCFS алгоритм виконувався над першою чергою, якщо всі процеси в 1 черзі завершені, починаємо обробляти другу чергу.
- Round-robin алгоритм виконувався над другою чергою
- Після виконання алгоритмів над обома чергами ми обчислюємо час який було витрачено на очікування і час витрачений на поворот.

Вивід

```
PROCESS |   PRIORITY|   BURST TIME| COMPLETION TIME |   WAITING TIME|   TURNAROUNDTIME

0       |   1       |   4         | 4               |   0           |   4

1       |   0       |   6         | 21              |   15          |   21

2       |   0       |   4         | 15              |   11          |   15

3       |   0       |   5         | 22              |   17          |   22

4       |   1       |   3         | 7               |   4           |   7

Average Waiting Time is: 9.4
Average Turnaround Time is: 13.8
Total time taken for execution of the code is 2749 millisec.
```