УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

**“ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО”**

Кафедра «Информационные технологии»

Лабораторная работа №5

**«Планирование процессов»**

Выполнил: студент гр. ИТП-11

Клевцов А.О

Принял: преподаватель

Карась О. В.

Гомель 2022

**Цель**: рассмотреть различные алгоритмы планирования процессов, предназначенных для достижения различных целей и эффективных для разных классов задач.

**Ход работы**

**Задание 1.** Не вытесняющие алгоритмы планирования процессов.

Выполнить различные алгоритмы планирований: *First-Come, First-Served(FCFS)* (Прямой и обратный), *Round Robin(RR), Shortest-Job-First (SJF)* (не вытесняющий), *Short-Job-First(SJF)* (не вытесняющий приоритетный). Вычислить полное время выполнения всех процессов и каждого в отдельности, время ожидание для каждого процесса. Рассчитать среднее время выполнения процесса и среднее время ожидания. Результаты оформить в виде таблиц, иллюстрирующих работы процессов. В соответствии с вариантом по журналу представлена следующая таблица:

|  |  |  |  |
| --- | --- | --- | --- |
| Процесс | Продолжительность | Время появления в очереди | Приоритет |
| P0 | 3 | 4 | 4 |
| P1 | 7 | 0 | 3 |
| P2 | 1 | 0 | 3 |
| P3 | 3 | 0 | 2 |

1. *FCFS* (Прямой)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 | И | И | И |  |  |  |  |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | И | И | И | И | И | И | И |  |  |  |  |
| P2 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И |  |  |  |
| P3 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И |

По названию алгоритму можно понять и его суть – первый пришел – первый обслужен. Собственно, здесь нет ничего сложного, ведь процессы обрабатываются в порядке своего появления, то есть в начале *P0, P1, P2, P3*.

Полное время выполнения всех процессов составит: 3 + 10 + 11 + 14 =38. Среднее время исполнения составит: (3 + 10 + 11 + 14) / 4 = 9.5. Среднее время ожидания: (0 + 3 + 9 + 11) / 4 =5.75.

1. *FCFS* (Обратный)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И |
| P1 | Г | Г | Г | Г | И | И | И | И | И | И | И |  |  |  |
| P2 | Г | Г | Г | И |  |  |  |  |  |  |  |  |  |  |
| P3 | И | И | И |  |  |  |  |  |  |  |  |  |  |  |

Здесь же все происходит аналогично, разве что в обратном порядке.

Полное время выполнения всех процессов составит: 3 + 4 + 11 + 14 =32. Среднее время исполнения составит: (3 + 4 + 11 + 14) / 4 = 8. Среднее время ожидания: (0 + 3 + 4 + 11) / 4 = 4.5. И в итоге получилось, что обратный FCFS гораздо эффективнее.

1. *Round Robin (RR)*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 | И | И | И | Г | Г | Г | Г | Г | Г | Г | И | И | И | И |
| P1 | Г | Г | Г | И | И | И |  |  |  |  |  |  |  |  |
| P2 | Г | Г | Г | Г | Г | Г | И |  |  |  |  |  |  |  |
| P3 | Г | Г | Г | Г | Г | Г | Г | И | И | И |  |  |  |  |

Суть данного алгоритма в том, что мы даем процессу некоторое время на выполнение, в нашем случае 3 кванта, и если длительность нашего процесса больше, чем квант времени, то он переходит в режим ожидания и позволяет работать другому процессу, и так далее

Полное время выполнения всех процессов составит: 14 + 6 + 7 + 10 = 37. Среднее время исполнения составит: (14 + 6 + 7 + 10) / 4 = 9.25. Среднее время ожидания: (7 + 3 + 6 + 7) / 4 = 5.75.

1. *Shortest-Job-First* (Не вытесняющий)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 | Г | И | И | И |  |  |  |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| P2 | И |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | Г | Г | Г | И | И | И |  |  |  |  |  |  |  |

Согласно данному алгоритму, первым выполняется тот процесс, чья длительность меньше.

Полное время выполнения всех процессов составит: 4 + 14 + 1 + 7 = 26. Среднее время исполнения составит: (4 + 14 + 1 + 7) / 4 = 6.5. Среднее время ожидания: (1 + 7 + 0 + 4) / 4 = 3.

5. *Shortest-Job-First* (Не вытесняющий, приоритетный)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 | Г | Г | Г | Г | И | И | И |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| P2 | И |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | И | И | И |  |  |  |  |  |  |  |  |  |  |

Согласно данному алгоритму, первым выполняется тот процесс, чья длительность меньше, но к этому еще добавляется и приоритет.

Полное время выполнения всех процессов составит: 7 + 14 + 1 + 4 = 26. Среднее время исполнения составит: (7 + 14 + 1 + 4) / 4 = 6.5. Среднее время ожидания: (4 + 7 + 0 + 1) / 4 = 3.

**Задание 2.** Вытесняющие алгоритмы планирования процессов

Выполнить различные алгоритмы планирования *Shortest-Job-First*(вытесняющий), *Shortest-Job-First*(приоритетный) для данных, приведенных в таблице. Рассчитать среднее время выполнения процесса и среднее время ожидания. Результаты оформить в виде таблиц, иллюстрирующих работы процессов.

1. *Shortest-Job-First*(Вытесняющий)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 |  | | | Г | И | И | И |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| P2 | И |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | И | И | И |  |  |  |  |  |  |  |  |  |  |

Вытесняющий похож на обычный *Shortest-Job-First*, за исключением того, что учитывается время, когда пришел процесс. И если, например, в квант времени 3 выполняется процесс продолжительностью 5, и приходит процесс с продолжительностью 3, то он его вытеснит и выполнится, а предыдущий процесс будет его ожидать.

Полное время выполнения всех процессов составит: 4 + 14 + 1 + 4 = 23. Среднее время исполнения составит: (4 + 14 + 1 + 4) / 4 = 5.75. Среднее время ожидания: (1 + 7 + 0 +1) / 4 = 2.25.

1. *Shortest-Job-First* (Вытесняющий, приоритетный)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| P0 |  | | | Г | И | И | И |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И |
| P2 | И |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | И | И | И |  |  |  |  |  |  |  |  |  |  |

Так уж вышло, что с учетом приоритета процессов результаты выполнения никак не поменяются, и будет тот же результат, что и при вытесняющем.

Полное время выполнения всех процессов составит: 4 + 14 + 1 + 4 = 23. Среднее время исполнения составит: (4 + 14 + 1 + 4) / 4 = 5.75. Среднее время ожидания: (1 + 7 + 0 +1) / 4 = 2.25.

1. **Вывод**:в ходе работы были рассмотрены различные алгоритмы планирования процессов, предназначенные для достижения различных целей и эффективные для разных классов задач. Для данных процессов самый эффективный алгоритм оказался *Shortest-Job-First*(Вытесняющий)