**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине: «Операционные системы»

на тему: **«**Планирование процессов**»**

Выполнил: студент гр. ИТП-11

Трацевский И.С.

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 | 5 | 0 | 2 |
| P1 | 8 | 3 | 1 |
| P2 | 2 | 1 | 3 |
| P3 | 4 | 4 | 3 |

**1. FCFS** (Прямой)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | И | И | И | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И |  |  |  |  |  |  |
| P2 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И |  |  |  |  |
| P3 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | 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: 5 + 13 + 15 + 19 = 52. Среднее время исполнения составит: (5 + 13 + 15 + 19) / 4 = 13. Среднее время ожидания: (0 + 5 + 13 + 15) / 4 = 8.25.

**2. FCFS** (Обратный)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И |
| P1 | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И |  |  |  |  |  |
| P2 | Г | Г | Г | Г | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | И | И | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Здесь же все происходит аналогично, разве что в обратном порядке. Полное время выполнения всех процессов составит: 4 + 6 + 14 + 19 = 43. Среднее время исполнения составит: (4 + 6 + 14 + 19) / 4 = 10.75. Среднее время ожидания: (0 + 4 + 6 + 14) / 4 = 6. И в итоге получилось, что обратный FCFS эффективнее.

**3. Round Robin** (**RR**)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | И | И | И | Г | Г | Г | Г | Г | Г | Г | Г | И | И |  |  |  |  |  |  |
| P1 | Г | Г | Г | И | И | И | Г | Г | Г | Г | Г | Г | Г | И | И | И | Г | И | И |
| P2 | Г | Г | Г | Г | Г | Г | И | И |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | Г | Г | Г | Г | Г | 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. В нашем же случае, процесс P0 выполняется 5 единиц времени, хотя на выполнение отведено лишь 3, поэтому он передает управление процессу P1. Процесс P1 занимает 8 единиц времени, и ему не хватает времени, чтобы выполниться за 1 раз. После P1 передает P2, а тот выполняется полностью, после очередь переходит к P3, после в свою очередь P0, который все еще ждал своей очереди, и так далее по кругу.

Полное время выполнения всех процессов составит: 13 + 19 + 8 + 17 = 57. Среднее время исполнения составит: (13 + 19 + 8 + 17) / 4 = 14.25. Среднее время ожидания: (8 + 11 + 6 + 13) / 4 = 9.5.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | Г | Г | Г | Г | Г | Г | И | И | И | И | И |  |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И |
| P2 | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | Г | И | И | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |

Согласно данному алгоритму, первым выполняется тот процесс, чья длительность меньше. И если отсортировать по порядку процессы, то мы получим следующее: P2, P3, P0, P1.

Полное время выполнения всех процессов составит: 2 + 6 + 11 + 19 = 38. Среднее время исполнения составит: 38/ 4 = 9.5. Среднее время ожидания: (6 + 11 + 0 + 2) / 4 = 4.75.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | И | И | И | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И |
| P2 | Г | Г | Г | Г | Г | И | И |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | Г | Г | Г | Г | Г | Г | И | И | И | И |  |  |  |  |  |  |  |  |

Согласно данному алгоритму, первым выполняется тот процесс, чья длительность меньше, но к этому еще добавляется и приоритет. И если отсортировать по порядку процессы, то мы получим следующее: P0, P2, P3, P1.

Полное время выполнения всех процессов составит: 5 + 19 + 7 + 11 = 32. Среднее время исполнения составит: (5 + 19 + 7 + 11) / 4 = 8. Среднее время ожидания: (0 + 5 + 7 + 11) / 4 = 5.75.

**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 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | И | Г | Г | И | Г | Г | Г | Г | И | И | И |  |  |  |  |  |  |  |  |
| P1 |  |  |  | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И |
| P2 |  | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 |  |  |  |  | И | 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: 11 + 16 + 2 + 4 = 33. Среднее время исполнения составит: (11 + 16 + 2 + 4) / 4 = 8.25. Среднее время ожидания: (6 + 8 + 0 +0) / 4 = 3.5.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| P0 | И | И | И | Г | Г | Г | Г | Г | Г | Г | Г | И | И |  |  |  |  |  |  |
| P1 |  |  |  | И | И | И | И | И | И | И | И |  |  |  |  |  |  |  |  |
| P2 |  | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И |  |  |  |  |
| P3 |  |  |  |  | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И | И | И |

Тот же принцип что и Shortest-Job-First(Вытесняющий), но с приоритетом.

Полное время выполнения всех процессов составит: 13 + 8 + 14 + 15 = 50. Среднее время исполнения составит: (13 + 8 + 14 + 15) / 4 = 12.5. Среднее время ожидания: (8 + 0 + 12 +11) / 4 = 3.5.

**Вывод**: в ходе работы были рассмотрены различные алгоритмы планирования процессов, предназначенные для достижения различных целей и эффективные для разных классов задач. В процессе выполнения лабораторной работы для предоставленных данных наиболее эфективным оказался **Shortest-Job-First** (Не вытесняющий, приоритетный).