УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

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

Кафедра «Информационные технологии»

Лабораторная работа №5

**«Планирование процессов»**

Выполнил: студент гр. ИТП-11

Воробьев В.А.

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 | 9 | 3 | 3 |
| P1 | 2 | 0 | 1 |
| P2 | 2 | 0 | 1 |
| P3 | 3 | 4 | 3 |

**1. FCFS** (Прямой)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| P0 | И | И | И | И | И | И | И | И | И |  |  |  |  |  |  |  |
| P1 | Г | Г | Г | Г | Г | Г | Г | Г | Г | И | И |  |  |  |  |  |
| P2 | 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 | 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: 9 + 11 + 13 + 16 = 49. Среднее время исполнения составит: (9 + 11 + 13 + 16) / 4 = 12.25. Среднее время ожидания: (0 + 9 + 11 + 13) / 4 = 8.25.

**2. FCFS** (Обратный)

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

Здесь же все происходит аналогично, разве что в обратном порядке. Полное время выполнения всех процессов составит: 3 + 5 + 7 + 16 = 31. Среднее время исполнения составит: (3 + 5 + 7 + 16) / 4 = 7.75. Среднее время ожидания: (0 + 3 + 5 + 7) / 4 = 3.75. И в итоге получилось, что обратный FCFS гораздо эффективнее.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| P0 | И | И | И | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И |
| P1 | Г | Г | Г | И | И |  |  |  |  |  |  |  |  |  |  |  |
| P2 | Г | Г | Г | Г | Г | И | 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. В нашем же случае, процесс P0 выполняется 9 единиц времени, хотя на выполнение отведено лишь 3, поэтому он передает управление процессу P1. Процессы P1 и P2 занимают 2 и 3 единицы времени, что хватает им, чтобы выполниться за 1 раз. После выполнения P1 передает P3, а P3 в свою очередь P0, который все это время ждал своей очереди.

Полное время выполнения всех процессов составит: 16 + 5 + 7 + 10 = 38. Среднее время исполнения составит: (16 + 5 + 7 + 8) / 4 = 9.5. Среднее время ожидания: (7 + 3 + 5 + 7) / 4 = 5.5.

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

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

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

Полное время выполнения всех процессов составит: 16 + 2 + 4 + 7 = 29. Среднее время исполнения составит: (16 + 2 + 4 + 7) / 4 = 7.25. Среднее время ожидания: (7 + 0 + 2 + 4) / 4 = 3.25.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| P0 | Г | Г | Г | Г | Г | Г | Г | И | И | И | И | И | И | И | И | И |
| P1 | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 | Г | Г | И | И |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 | Г | Г | Г | Г | 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: P1, P2, P3, P0. Так уж совпало, что с этими данными с учетом приоритета процесса мы получим одинаковый результат.

Полное время выполнения всех процессов составит: 16 + 2 + 4 + 7 = 29. Среднее время исполнения составит: (16 + 2 + 4 + 7) / 4 = 7.25. Среднее время ожидания: (7 + 0 + 2 + 4) / 4 = 3.25.

**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 |
| P0 |  |  |  | Г | Г | Г | Г | И | И | И | И | И | И | И | И | И |
| P1 | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 | Г | Г | И | 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.

В нашем случае процесс P1 и P2 приходят в момент времени 0, и при этом никаких других процессов пока нет. В момент времени 3 приходят процессы P0 и P3, но они выполняются дольше, чем P2, поэтому они ожидают. После его выполнения меньшим по времени является процесс P3, а не P1, и поэтому выполняется именно он.

Полное время выполнения всех процессов составит: 13 + 2 + 4 + 4 = 23. Среднее время исполнения составит: (13 + 2 + 4 + 4) / 4 = 5.75. Среднее время ожидания: (4 + 0 + 2 +1) / 4 = 1.75.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процесс | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| P0 |  |  |  | Г | Г | Г | Г | И | И | И | И | И | И | И | И | И |
| P1 | И | И |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 | Г | Г | И | 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: 13 + 2 + 4 + 4 = 23. Среднее время исполнения составит: (13 + 2 + 4 + 4) / 4 = 5.75. Среднее время ожидания: (4 + 0 + 2 +1) / 4 = 1.75.

**Вывод**: в ходе работы были рассмотрены различные алгоритмы планирования процессов, предназначенные для достижения различных целей и эффективные для разных классов задач.