Задание к лабораторной работе №1 (осень 2015). Планирование процессов.
Задача: изучить основные алгоритмы планирования процессов.

Алгоритмы планирования

Существует достаточно большой набор разнообразных алгоритмов *планирования*, которые предназначены для достижения различных целей и эффективны для разных классов задач. Рассмотрим следующие.

First-Come, First-Served (FCFS)

Простейшим алгоритмом *планирования* является алгоритм, который принято обозначать аббревиатурой ***FCFS*** по первым буквам его английского названия – First-Come, First-Served (первым пришел, первым обслужен). Представим себе, что процессы, находящиеся в состоянии готовность, выстроены в очередь. Когда процесс переходит в состояние готовность, он, а точнее, ссылка на его *PCB* помещается в конец этой очереди. Выбор нового процесса для исполнения осуществляется из начала очереди с удалением оттуда ссылки на его *PCB*. Очередь подобного типа имеет в программировании специальное наименование – FIFO, сокращение от First In, First Out (первым вошел, первым вышел).

Такой алгоритм выбора процесса осуществляет *невытесняющее планирование*. Процесс, получивший в свое распоряжение процессор, занимает его до истечения. После этого для выполнения выбирается новый процесс из начала очереди.

Круговое, карусельное, Round Robin (RR)

Модификацией *алгоритма FCFS* является алгоритм, получивший название *Round Robin* (*Round Robin* – это вид детской карусели в США) или сокращенно ***RR***. По сути дела, это тот же самый алгоритм, только реализованный в режиме *вытесняющего планирования*. Можно представить себе все множество готовых процессов организованным циклически – процессы сидят на карусели. Карусель вращается так, что каждый процесс находится около процессора небольшой фиксированный *квант времени*,
обычно 10 – 100 миллисекунд. Пока процесс находится рядом с процессором, он получает процессор в свое распоряжение и может исполняться. Реализуется такой алгоритм так же, как и предыдущий, с помощью организации процессов, находящихся в состоянии готовность, в очередь FIFO. Планировщик выбирает для очередного исполнения процесс, расположенный в начале очереди, и устанавливает таймер для генерации прерывания по истечении определенного *кванта времени*. При выполнении процесса возможны два варианта.

- Время непрерывного использования процессора, необходимое процессу, меньше или равно продолжительности *кванта времени*. Тогда процесс по своей воле освобождает процессор до истечения *кванта времени*, на исполнение поступает новый процесс из начала очереди, и таймер начинает отсчет *кванта* заново.

- Продолжительность остатка текущего *CPU* процесса больше, чем *квант времени*. Тогда по истечении этого *кванта* процесс прерывается таймером и помещается в

конец очереди процессов, готовых к исполнению, а процессор выделяется для использования процессу, находящемуся в ее начале.

Shortest process next (SPN)

При рассмотрении алгоритмов *FCFS* и *RR* видно, насколько существенным для них является порядок расположения процессов в очереди процессов, готовых к исполнению. Если короткие задачи расположены в очереди ближе к ее началу, то общая производительность этих алгоритмов значительно возрастает. Если бы мы знали время исполнения следующих процессов, находящихся в состоянии готовность, то могли бы выбрать для исполнения не процесс из начала очереди, а процесс с минимальной длительностью. Если же таких процессов два или больше, то для выбора одного из них можно использовать уже известный нам *алгоритм FCFS*. Квантование времени при этом не применяется. Описанный алгоритм получил название "кратчайшая работа первой

SPN -алгоритм невытесняющий. При *невытесняющем SPN – планировании* процессор предоставляется избранному процессу на все необходимое ему время, независимо от событий, происходящих в вычислительной системе.

Shortest remaining time (SRT)

SRT – *вытесняющий вариант SJF – планирования.* В этом варианте планирования планировщик выбирает процесс с наименьшим остающимся временем. Учитывается появление новых процессов в очереди готовых к исполнению (из числа вновь родившихся или разблокированных) во время работы выбранного процесса. Если время работы вновь появившегося процесса меньше, чем остаток времени выполнения исполняющегося, то исполняющийся процесс вытесняется новым.

См. Столлингс В. Операционные системы. Внутреннее устройство и реализация. Часть 4.

В соответствии с вариантом задания реализовать программную имитацию реализации алгоритма планирования (2 шт.).

Программно подготовить исходные данные, представляющие собой пары чисел, где первое число – время поступления процесса, а второе – время его выполнения. Очевидно, что последовательность первых значений – возрастающая (временные парадоксы запрещены).
Например:

Время появления:	0	4	6	7	11	13	19	24
Длительность выполнения	4	7	8	4	5	8	2	5

Диапазон времен выполнения, и среднее время между поступающими процессами взять из варианта задания.
Для полученных данных определить среднее время оборота процессов и среднее время ожидания процессов, в соответствии с реализованными алгоритмами планирования.

Сделать вывод о предпочтительном алгоритме планирования, считая критерием эффективности минимальное среднее время оборота процессов для нечетных вариантов, и минимальное среднее время ожидания для четных вариантов.

№ варианта	Алгоритм 1	Алгоритм 2	Диапазон длительностей выполнения	Среднее время между поступающими процессами.
1	**FCFS**	**RR(2)**	4÷8	6±3
2	**SPN**	**RR(4)**	5÷12	9±3
3	**SRT**	**SPN**	3÷9	4±3
4	**FCFS**	**SPN**	3÷7	4±3
5	**FCFS**	**SRT**	2÷10	8±4
6	**RR(4)**	**SPN**	3÷12	10±3
7	**SRT**	**RR(4)**	8÷15	16±3
8	**FCFS**	**SRT**	2÷12	13±3
9	**RR(2)**	**SRT**	4÷8	6±4
10	**FCFS**	**SPN**	3÷10	9±3
11	**SPN**	**RR(1)**	3÷6	4±3
12	**SRT**	**RR(3)**	4÷8	5±3
13	**FCFS**	**RR(5)**	10÷20	12±3
14	**RR(2)**	**SRT**	3÷11	7±3
15	**SRT**	**RR(2)**	4÷8	6±4
16	**FCFS**	**RR(4)**	6÷12	7±3
17	**FCFS**	**RR(1)**	3÷12	10±3
18	**SRT**	**RR(3)**	4÷10	10±5
19	**RR(2)**	**SRT**	4÷8	6±3
20	**FCFS**	**SPN**	4÷8	6±3

По данной лабораторной работе представить отчет:

- Набор исходных данных.
- Величину времени оборота или времени ожидания (в соответствии с вариантом) для каждого из реализованных алгоритмов планирования.
- Максимальное количество процессов в очереди – для желающих получить отличную оценку.