Лабораторная работа №2 “Алгоритмы планирования”

Этот проект предполагает реализацию нескольких различных алгоритмов планирования процессов. Планировщику будет назначен заранее определенный набор задач, и он будет планировать задачи на основе выбранного алгоритма планирования. Каждой задаче назначается приоритет и количество времени в течении которого она будет использовать процессор.

Необходимо реализовать следующие алгоритмы планирования:

- «Первым пришел — первым обслужен» (FCFS), при котором задачи планируются в том порядке, в котором они запрашивают ЦПУ.
- Shortest-job-first (SJF), при котором задачи планируются в порядке продолжительности использования ими ЦПУ.
- Приоритетное планирование, при котором задачи планируются на основе приоритета.
- Циклическое планирование (RR), при котором каждая задача выполняется в течение определенного кванта времени.(или оставшуюся часть времени использования ЦПУ).
- Циклическое планирование с приоритетом: задачи планируются в порядке приоритета и используется циклическое планирование для задач с одинаковым приоритетом.

Приоритеты варьируются от 1 до 10, где более высокое числовое значение указывает на более высокий относительный приоритет. Для циклического планирования длина кванта времени составляет 10 миллисекунд.

Реализация проекта.

Проект должен быть реализован на языке Си. Вам предоставляется готовый фреймворк. В фреймворке вспомогательные файлы считывают расписание задач, вставляют задачи в список и вызывают планировщик. Ваша задача просто реализовать каждый алгоритм планирования согласно его описанию.

Расписание задач имеет форму **[имя задачи] [приоритет] [время использования ЦПУ]** со следующим примером формата:

```
T1, 4, 20
T2, 2, 25
T3, 3, 25
T4, 3, 15
T5, 10, 10
```

Таким образом, задача T1 имеет приоритет 4 и загрузку процессора в 20 миллисекунд и так далее. Предполагается, что все задачи поступают одновременно, поэтому алгоритмы планировщика не обязаны поддерживать процессы с более высоким приоритетом, вытесняя процессы с более низким приоритетом. Кроме того,

задачи не обязательно помещать в очередь или список в каком-то определенном порядке.

Существует несколько различных стратегий организации списка задач. Один из подходов — поместить все задачи в один неупорядоченный список, где стратегия выбора задач зависит от алгоритма планирования. Вероятно, при выполнении этого проекта вы сочтете функциональность списка наиболее подходящей.

Детали реализации проекта.

Файл driver.c считывает расписание задач, вставляет каждую задачу в связанный список и вызывает планировщик процессов, вызывая функцию schedule(). Функция schedule() выполняет каждую задачу в соответствии с указанным алгоритмом планирования. Задачи, выбранные для выполнения на ЦП, определяются функцией pick-NextTask() и выполняются путем вызова функции run(), определенной в файле cpu.c. Makefile используется для определения конкретного алгоритма планирования, который будет вызываться драйвером. Например, чтобы скомпилировать планировщик FCFS, мы должны ввести:

```
make fcfs
```

и чтобы запустить полученный планировщик с задачами указанными в файле sheduler.txt, мы должны ввести:

```
./fcfs schedule.txt
```

Дополнительное задание

Подсчитайте среднее оборотное время, время ожидания и время отклика для каждого алгоритма планирования.