

Оглавление

1. Задание	2
2. Описание работы планировщика	2
3. Пример работы программы	3
4. Заключение	5

1. Задание

Задачей курсового проекта является разработка программы, которая выполняет эмуляцию работы планировщика задач, используемого в операционных системах. Планировщик задач должен использовать алгоритм планирования типа SJF (shortest job first).

Процессу задается приоритет, время выполнения процесса, а также список захватываемых ресурсов. На каждом такте из очереди готовых процессов выбирается на выполнение процесс в соответствии с алгоритмом планирования. Активный процесс захватывает необходимые ресурсы на некоторое время.

Алгоритм планирования	Число процессоров	Операционная система
SJF не вытесняющее	2	Windows

2. Описание работы планировщика

Стратегия Shortest Job First (SJF, обслуживание самого короткого задания первым) – стратегия диспетчеризации процессора, при которой процессор предоставляется в первую очередь наиболее короткому процессу из имеющихся в системе.

В данной курсовой работе формировании очереди участвуют следующие факторы:

1. Приоритет
 - 1) Высокий (минимальный квант времени 2 секунды)
 - 2) Низкий (минимальный квант времени 1 секунды)
2. Время
3. Количество занимаемых ресурсов (файлов) (тем меньше, тем выше по очереди)

В алгоритме реализована система повышения приоритета, если процесс занял более 50% всех возможных ресурсов системы.

В данном случае с каждым процессом связывается длина его очередного периода активности. Эта длина используется для того, чтобы первым обслужить самый короткий процесс.

Не вытесняющий алгоритм SJF-планирования заключается в том, что процессор предоставляется избранному процессу на минимальный квант времени согласно приоритету процесса. После истечения этого кванта времени планировщик попытается в очереди процессов найти самый малоприоритетный процесс и переключить процессор на него. Минимальное время выполнения малоприоритетного процесса зависит также от его приоритета. После выполнения малоприоритетного процесса алгоритмом планирования осуществляется переход обратно к ранее выполняемому процессу.

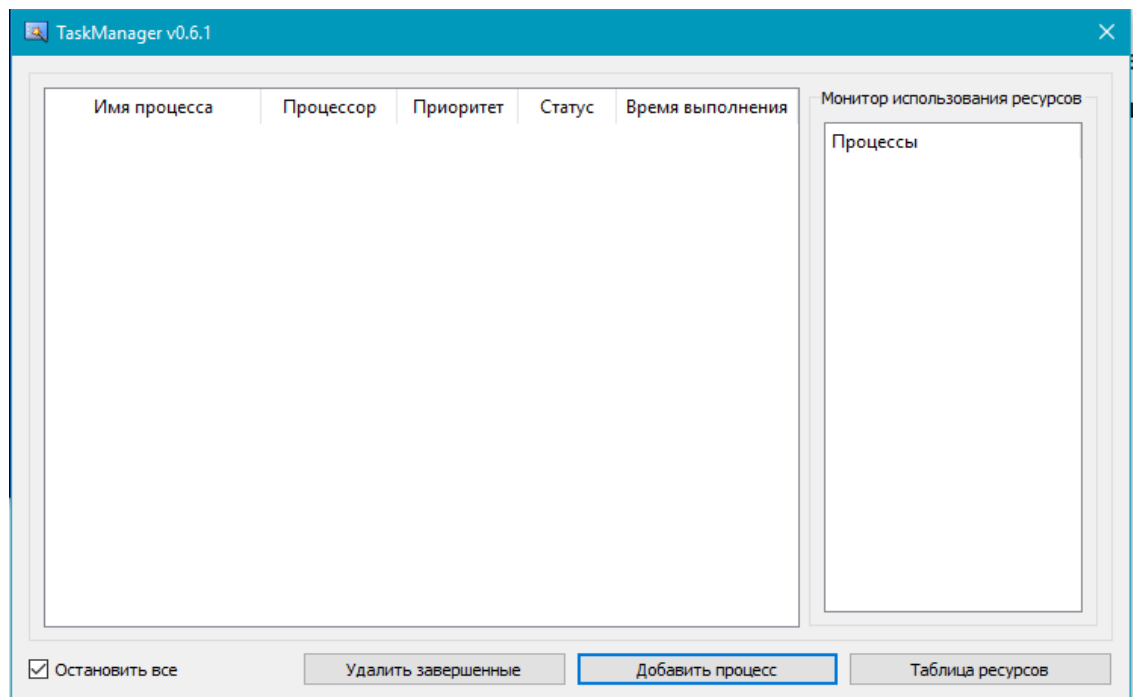
Выбор второго процесса осуществляется последствием анализа текущих выполняемых процессов. Выбран будет тот процесс, который будет находится в конце

очереди, а также он будет не занят другим процессором и файлы, которые он будет захватывать будут свободны.

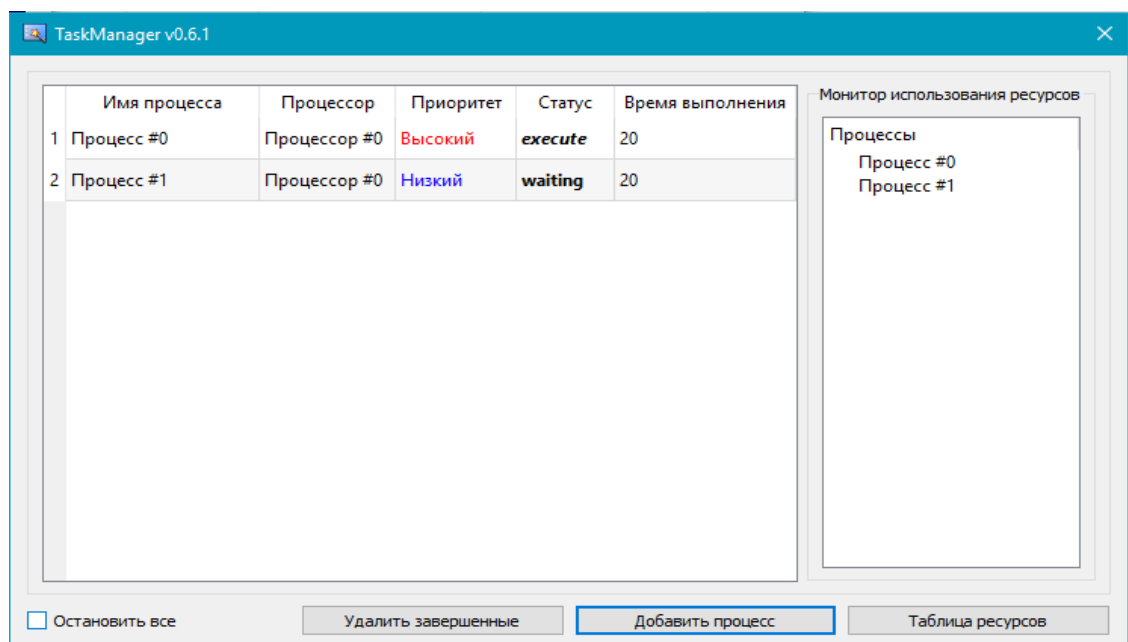
При не вытесняющем SJF-планировании новые процессы записываются в конец очереди, затем сразу же осуществляется перепланирование процессов, в результате чего очередь процессов может измениться. Стоит отметить что после завершения процесса осуществляется перепланирование очереди.

3. Пример работы программы

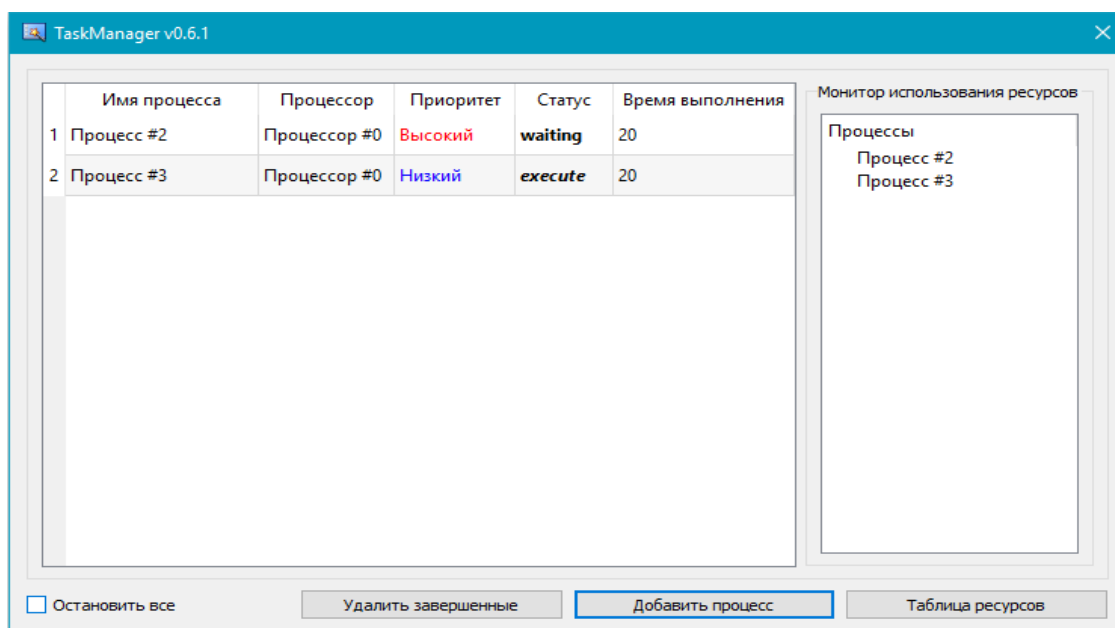
Внешний вид эмулятора:



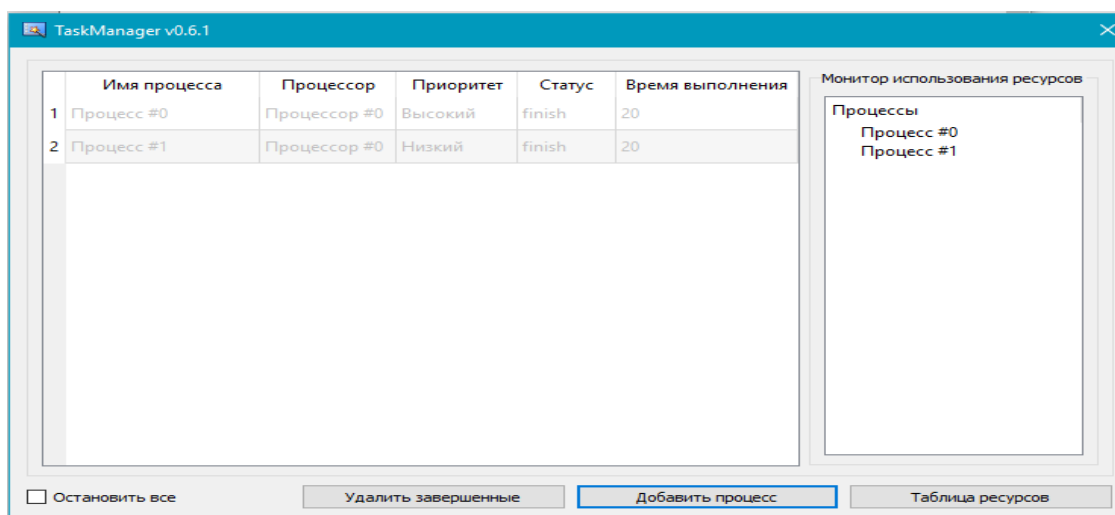
1. Создадим два процесса с разными приоритетами:



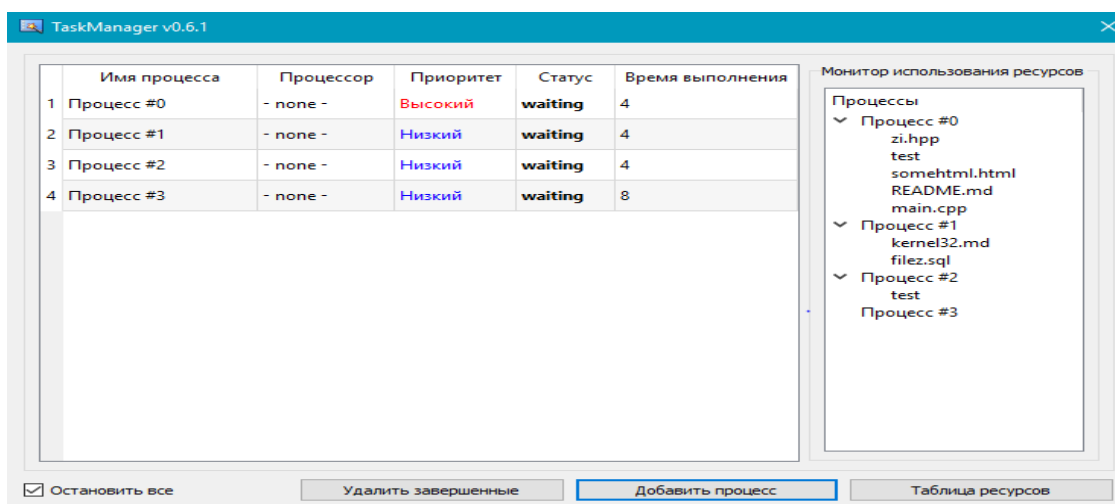
2. Результат программы спустя минимальный квант времени высокого приоритета:



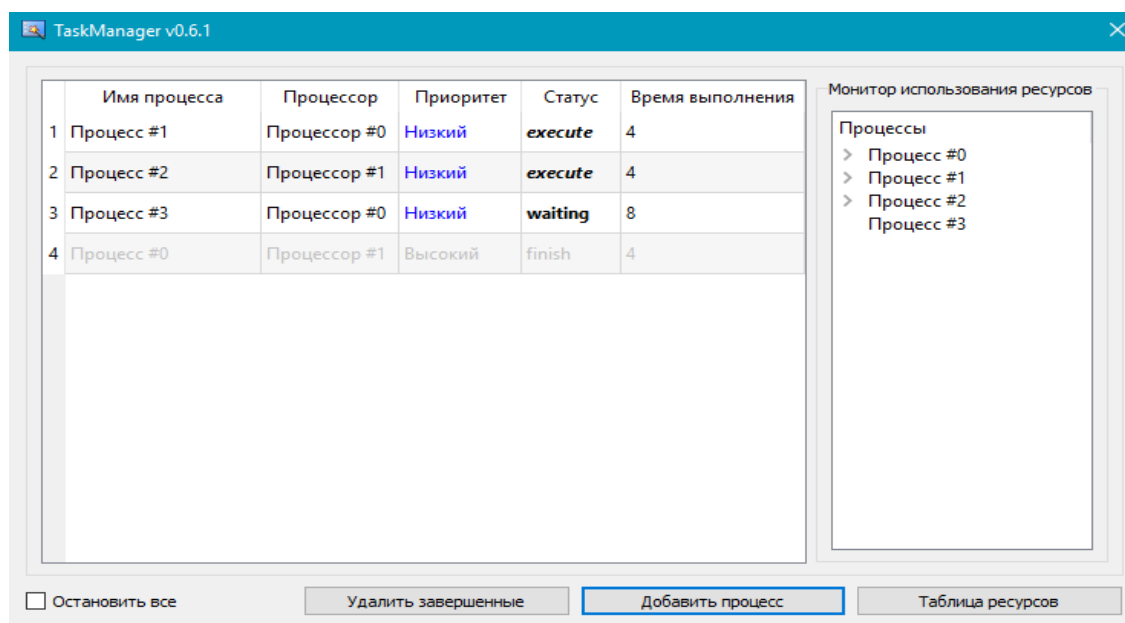
3. Результат выполнения процессов:



1. Пример процессов с ресурсами:



2. Пример работы эмулятора при выполнении нескольких задач:



4. Заключение

В данной курсовой работе была разработана программа, которая выполняет задачу просмотра графических моделей и сцен, построенных в графическом пакете Blender. Задачей курсового проекта является разработка программы, которая выполняет эмуляцию работы планировщика задач, используемого в операционных системах. Планировщик задач должен использовать алгоритм планирования типа SJF (shortest job first).

Процессу задается приоритет, время выполнения процесса, а также список захватываемых ресурсов. На каждом такте из очереди готовых процессов выбирается на выполнение процесс в соответствии с алгоритмом планирования. Активный процесс захватывает необходимые ресурсы на некоторое время.

Эмулятор написан на языке Python v2.7.12 с использованием библиотеки PyQt4. Для работы интерфейсом пользователя использовались модули Qt. Qt - кроссплатформенный инструмент разработки ПО на языке программирования C++. Является свободным программным обеспечением, распространяемой с лицензией GNU LGPL и GNU GPL.

Программа выполняет поставленную задачу и работает без ошибок (для корректных тестовых данных).