
ДОМАШНЕЕ ЗАДАНИЕ №5

БОБРУСКИНА СТАНИСЛАВА БПИ207

ВАРИАНТ 1

Задача о парикмахере

ОПИСАНИЕ ЗАДАНИЯ (вариант 1):

В тихом городке есть парикмахерская. Салон парикмахерской мал, ходить там может только парикмахер и один посетитель. Парикмахер всю жизнь обслуживает посетителей. Когда в салоне никого нет, он спит в кресле. Когда посетитель приходит и видит спящего парикмахера, он будит его, садится в кресло и спит, пока парикмахер занят стрижкой. Если посетитель приходит, а парикмахер занят, то он встает в очередь и засыпает. После стрижки парикмахер сам провожает посетителя. Если есть ожидающие посетители, то парикмахер будит одного из них и ждет пока тот сядет в кресло парикмахера и начинает стрижку. Если никого нет, он снова садится в свое кресло и засыпает до прихода посетителя. Создать многопоточное приложение, моделирующее рабочий день парикмахерской.

В сущности, требуется разработать многопоточное консольное приложение на языке программирования C (*стиль написания – произвольный*).

ОПИСАНИЕ ПРОГРАММЫ

Я сделала дополнительный функционал, а именно:

- Количество мест в комнате ожидания ограничено (если мест в комнате ожидания нет, то клиент уходит)
- Парикмахеров может быть несколько.

Переменные, отвечающие за количество парикмахеров и мест ожидания, находятся в файле `declaration.h`

Таким образом, можно выделить следующие потоки: главный поток - В этом потоке выполняется функция `main`, из которой и стартует приложение; а также у каждого клиента свой поток и у каждого парикмахера свой поток.

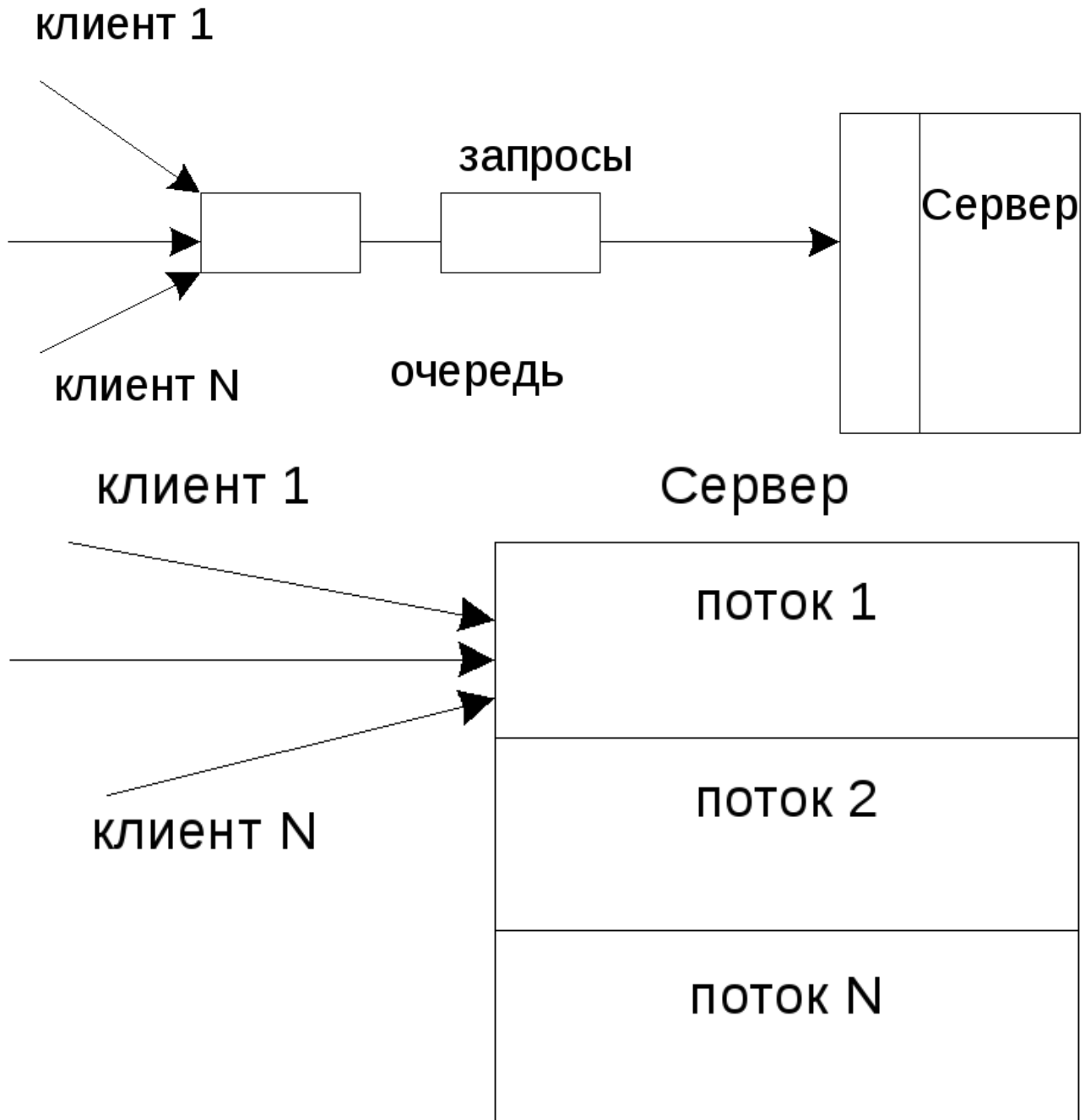
В данной задаче ресурсами являются места ожидания и рабочее кресло. Потоки- клиенты должны получить эти ресурсы строго последовательно: сначала посетитель должен найти место в комнате ожидания и только затем занять очередь к парикмахеру. При этом, предоставление рабочего кресла для обслуживания производит специальный процесс-парикмахер. В этом плане, парикмахера можно интерпретировать как сервер, предоставляющий требуемый сервис.

ОПИСАНИЕ ИСПОЛЬЗОВАННОЙ МОДЕЛИ ВЫЧИСЛЕНИЙ:

Для написания программы я воспользовалась моделью «Клиенты и серверы»

Описание:

Часто необходимо обрабатывать некоторые запросы на обслуживание. Традиционный подход заключается в том, что поток ждет сообщение, обрабатывает его, затем снова ждет и т.д. Другой подход заключается в создании нового потока всякий раз, когда приходит сообщение. Преимущества многопоточной обработки запросов иллюстрируются рис.



Пользовательские потоки не являются потоками ядра, т.е. не диспетчеризуются. Переключение потоков на пользовательском уровне выполняется быстро, т.к. требует вызова ОС. Каждое приложение может выполнять собственную копию исполнительной библиотеки, следовательно, может выбирать требуемую стратегию планирования, а не следует единой встроенной стратегии диспетчера.

Потоки ядра диспетчеризуются. Для каждого пользовательского потока создаётся поток ядра и всё планирование осуществляется ядром на основании единой стратегии. Блокирование и разблокирование потоков выполняется ядром и происходит, когда потоку нужно дождаться завершения операции ввода-вывода, а также в ответ на запросы блокирования и разблокирования, поступающие от исполнительной библиотеки, когда требуется изменение состояния конкретного потока.

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:

- Число заголовочных файлов – 1
- Число модулей реализации – 1
- Общий размер исходных текстов – 113 строк кода.
- Размер исполняемого файла – 6 КБ