

ИД3-2 Умертаев Арслан БПИ227 Вариант 16

Условие:

В гостинице 10 одноместных номеров. Клиенты гостиницы снимают номер на одни или несколько суток (задается при создании клиента). Если в гостинице нет свободных номеров, клиенты не уходят, а устраиваются на рядом с гостиницей на скамейках и ждут в порядке очереди, пока любой из номеров не освободится (других гостиниц в городе нет). Создать многопроцессное приложение, моделирующее работу гостиницы. Каждого клиента и гостиницу (точнее ее администратора) моделировать отдельным процессом.

На 4-5

В программе используются именованные POSIX семафоры для синхронизации доступа к ресурсам (номерам гостиницы) и разделяемая память для хранения информации о состоянии номеров.

В программе предусмотрено удаление семафоров и разделяемой памяти по ее завершению.

Реализовано завершение программы в соответствии с условием задачи.

Процессы и взаимодействие

Гостиница (администратор) - ответственна за управление доступом к номерам через мьютекс (mutex) и семафор на количество свободных номеров (room_sem).

Клиенты (посетители) - реализованы в виде отдельного процесса. Процессы клиентов пытаются занять номер, ожидают на скамейке, если все номера заняты, и освобождают номер после использования.

Использование семафоров и разделяемой памяти

Именованные POSIX семафоры- room_sem: контролирует количество доступных номеров.

Когда клиент освобождает номер, значение семафора увеличивается, позволяя следующему клиенту в очереди войти в номер.

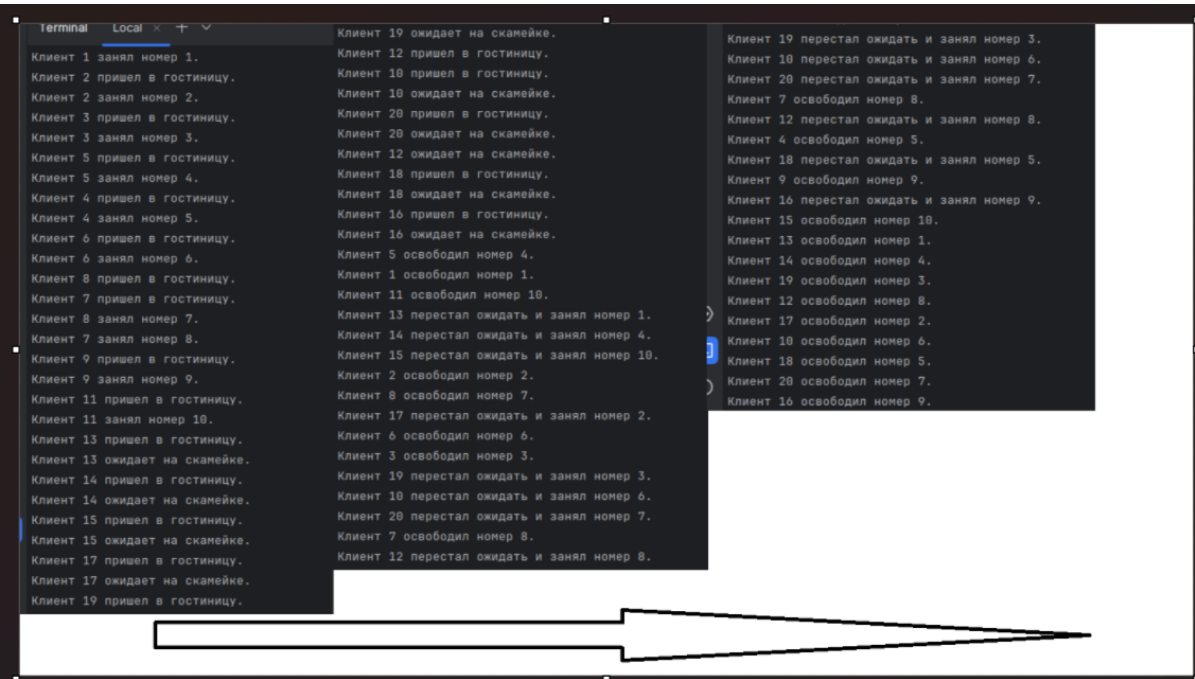
-mutex: защищает доступ к разделяемой памяти, гарантируя, что только один процесс может изменять состояние номеров в один момент времени.

Разделяемая память: Хранит текущее состояние номеров гостиницы, позволяя всем процессам клиентов видеть, какие номера заняты и какие свободны.

При получении сигнала программа корректно завершает работу, освобождая все ресурсы: семафоры удаляются, разделяемая память отключается.

После того как все клиенты завершили свою работу (т.е., заселились и покинули номера), главный процесс закрывает и освобождает все ресурсы, завершая работу программы.

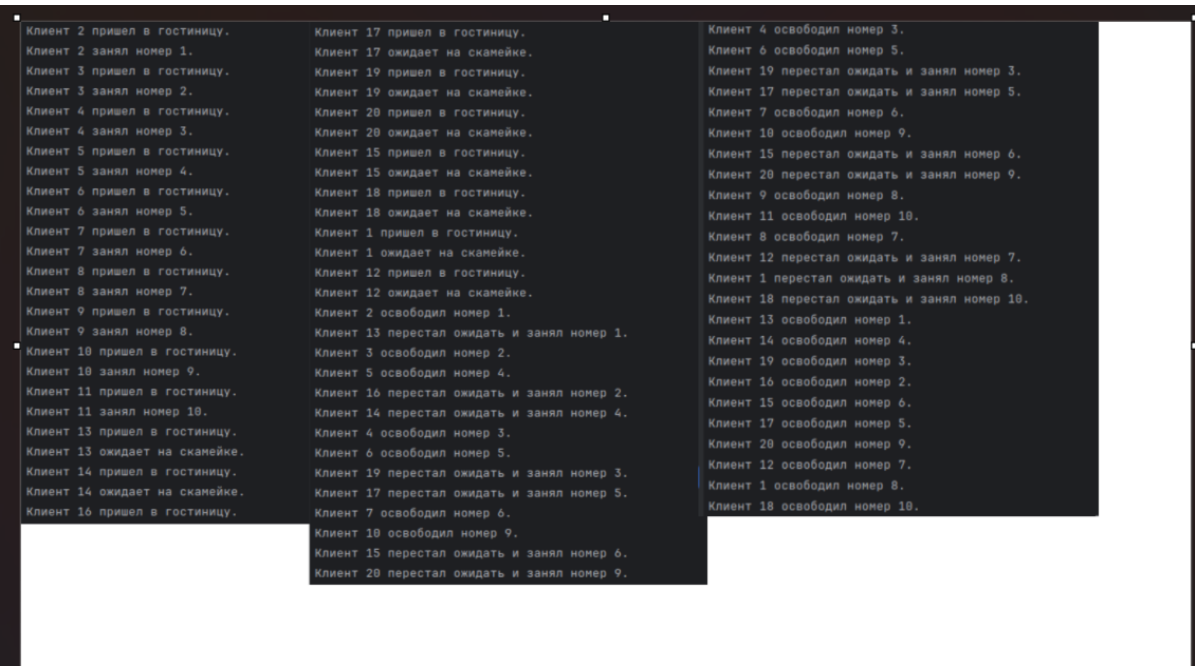
Пример работы



На 6-7

В программе используются разделяемая память и неименованные POSIX семафоры для создания много процессного приложения. Реализация аналогична на 4-5

Пример работы:



На 8

Архитектура программы

Программа состоит из нескольких независимо запускаемых процессов, которые взаимодействуют через семафоры и разделяемую память.

Основные компоненты системы:

admin.c

Отвечает за начальную инициализацию разделяемой памяти, установку всех номеров в состояние "свободно".

client.c

Отвечает за моделирование клиентов гостиницы.- Каждый процесс клиента пытается занять номер, остается в нем некоторое время, а затем освобождает его.

ipc_resources.c

Содержит общие функции и определения, используемые другими частями программы, включая инициализацию разделяемой памяти и семафоров, а также вспомогательные функции для работы с ними.

start_clients.sh

Скрипт на Bash для запуска множества клиентских процессов. Организует параллельный запуск клиентов и управляет их жизненным циклом.

Пример работы:

```
Клиент 6 запущен.  
Клиент 6 занял номер 6.  
Клиент 7 запущен.  
Клиент 7 занял номер 7.  
Клиент 8 запущен.  
Клиент 8 занял номер 8.  
Клиент 9 запущен.  
Клиент 9 занял номер 9.  
Клиент 10 запущен.  
Клиент 10 занял номер 10.  
Клиент 11 запущен.  
Клиент 11 ожидает на скамейке.  
Клиент 12 запущен.  
Клиент 12 ожидает на скамейке.  
Клиент 13 запущен.  
Клиент 13 ожидает на скамейке.  
Клиент 14 запущен.  
Клиент 14 ожидает на скамейке.  
Клиент 13 занял номер 1.  
Клиент 1 освободил номер 1.  
Клиент 15 запущен.  
Клиент 15 ожидает на скамейке.  
Клиент 2 освободил номер 2.  
Клиент 15 занял номер 2.  
Клиент 16 запущен.
```