

Домашнее задание 8. Мусаев Умахан Рашидович.

БПИ234.

Условие:

До 8 баллов

Модифицировать последнюю рассмотренную программу таким образом, чтобы данные в буфер записывались по кольцу и читались из него в том же порядке. Синхронизацию читателя и писателя при этом можно не менять. Сохранить также возможность работы одного писателя и одного читателя.

Опционально +1 балл

Использовать в программе с кольцевым буфером два отдельных мьютекса, реализуемых посредством POSIX-семафоров. Отдельно для читателей и писателей (аналогично решению в программе по курсу "Архитектура ВС", реализованному для многопоточного приложения).

Опционально +1 балл

Переработать код так, чтобы он допускал подключение до двух читателей при одном писателе с возможным завершением (как и реализовано в примере) всех процессов по завершению одного из них.

Предоставить только один вариант программ, учитывающих все реализованные ситуации. Опциональные решения можно реализовать как одну из альтернатив, если решение не планируется на 10 баллов. То есть каждая из них не зависит от другой.

Дополнительно к отчету, описывающему проделанную работу, предоставить исходные тексты программ, решающих поставленную задачу на искомую оценку.

Отчет:

Структура проекта:

- `sharedmemory.h` - заголовочный файл с определениями общих констант и структуры разделяемой памяти.
- `client.cpp` - код писателя, который генерирует данные и записывает их в буфер.
- `server_hw8.cpp` - код читателя, который считывает данные из буфера и выводит их.

Shared Memory позволяет разным процессам совместно использовать область памяти для обмена данными.

Семафоры POSIX используются для синхронизации доступа к разделяемым ресурсам между процессами.

Структура разделяемой памяти:

```
struct SharedMemoryBuffer {  
    int buffer[BUFFER_SIZE]; // Кольцевой буфер  
    size_t write_index; // Индекс записи  
    size_t read_index; // Индекс чтения  
    size_t reader_count; // Количество текущих читателей  
    bool data_available; // Флаг наличия данных  
    bool writer_active; // Флаг активности писателя  
};
```

- `buffer` - массив фиксированного размера, реализующий кольцевой буфер.
- `write_index` - индекс следующей позиции для записи.
- `read_index` - индекс следующей позиции для чтения.
- `reader_count` - текущее количество активных читателей.
- `data_available` - флаг, указывающий, есть ли данные для чтения.
- `writer_active` - флаг, указывающий, активен ли писатель.

Семафоры:

- **SEM_WRITER_NAME** - семафор для синхронизации доступа писателя к буферу.
- **SEM_READER_NAME** - семафор для синхронизации доступа читателей к буферу и управления количеством читателей.

Client.cpp:

-Инициализация:

- Создает или открывает разделяемую память.
- Отображает разделяемую память в адресное пространство.
- Инициализирует индексы и флаги в структуре `SharedMemoryBuffer``.
- Создает или открывает семафоры писателя и читателя.

Основной цикл:

- Генерирует случайные числа.
- Захватывает семафор писателя.
- Пишет данные в буфер в позицию **write_index**.
- Обновляет **write_index** с учетом цикличности.
- Устанавливает флаг **data_available**.
- Освобождает семафор писателя.
- Пауза для имитации реальной работы.

Завершение работы:

- Устанавливает флаг **writer_active** в false, уведомляя читателей о завершении работы.
- Закрывает семафоры и разделяемую память.

Server_hw8.cpp:

Инициализация:

- Открывает существующую разделяемую память.
- Отображает разделяемую память в адресное пространство.
- Открывает семафоры писателя и читателя.

Регистрация читателя:

- - Захватывает семафор читателя.
- - Проверяет, не превышен ли максимальный лимит читателей (**MAX_READERS**).
- - Увеличивает счетчик **reader_count**.
- - Освобождает семафор читателя.

Основной цикл:

- Проверяет флаги **writer_active** и **data_available**.
- Захватывает семафор читателя.
- Если данные доступны, считывает их из буфера в позиции **read_index**.
- Обновляет **read_index** с учетом цикличности.
- Если все данные прочитаны, сбрасывает флаг **data_available**.
- Освобождает семафор читателя.
- Пауза для имитации реальной работы.

Завершение работы:

- Захватывает семафор читателя.
- Уменьшает счетчик reader_count.
- Освобождает семафор читателя.
- Если это был последний читатель и писатель завершил работу, освобождает разделяемую память и семафоры.

Инструкция по запуску:

Компиляция:

1. `g++ client.cpp -o client`
2. `g++ server_hw8.cpp -o server_hw8`

Запуск программы:

1. `./client` (в 1 терминале)
2. `./server_hw8` (в 2 терминале)
3. `./server_hw8` (в 3 терминале)

Пример работы программы:

```
umahan@50cent: ~  
umahan@50cent:~$ ./client  
Писатель записал: 907  
Писатель записал: 830  
Писатель записал: 631  
Писатель записал: 635  
Писатель записал: 481  
Писатель записал: 445  
Писатель записал: 564  
Писатель записал: 833  
Писатель записал: 401  
  
umahan@50cent:~$ ./server_hw8  
Читатель (639) прочитал: 907  
Читатель (639) прочитал: 830  
Читатель (639) прочитал: 631  
Читатель (639) прочитал: 635  
Читатель (639) прочитал: 481  
Читатель (639) прочитал: 445  
Читатель (639) прочитал: 564  
Читатель (639) прочитал: 833  
Читатель (639) прочитал: 401
```

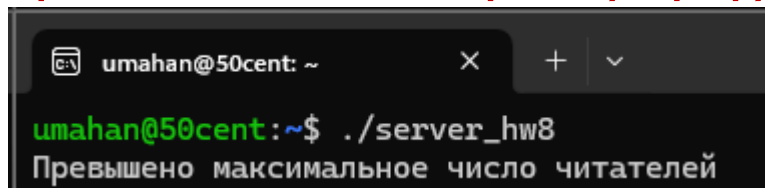
Два сервиса:

```
umahan@50cent: ~  
umahan@50cent:~$ ./client  
Писатель записал: 196  
Писатель записал: 870  
Писатель записал: 610  
Писатель записал: 69  
Писатель записал: 422  
Писатель записал: 684  
Писатель записал: 747  
Писатель записал: 600  
Писатель записал: 781  
  
umahan@50cent:~$ ./server_hw8  
Читатель (645) прочитал: 196  
Читатель (645) прочитал: 870  
Читатель (645) прочитал: 610  
Читатель (645) прочитал: 69  
Читатель (645) прочитал: 484  
Читатель (645) прочитал: 386  
Читатель (645) прочитал: 366  
Читатель (645) прочитал: 135  
Читатель (645) прочитал: 21  
  
umahan@50cent:~$ ./server_hw8  
Читатель (646) прочитал: 422  
Читатель (646) прочитал: 684  
Читатель (646) прочитал: 747  
Читатель (646) прочитал: 600  
Читатель (646) прочитал: 781  
Читатель (646) прочитал: 456  
Читатель (646) прочитал: 728  
Читатель (646) прочитал: 964  
Читатель (646) прочитал: 112
```

Завершение работы:

```
Писатель записал: 97  
Писатель записал: 605  
Писатель записал: 45  
Писатель записал: 714  
Писатель записал: 536  
Писатель записал: 889  
Писатель записал: 476  
Писатель записал: 413  
Писатель записал: 988  
umahan@50cent:~$ |  
  
Читатель (660) прочитал: 605  
Читатель (660) прочитал: 45  
Читатель (660) прочитал: 714  
Читатель (660) прочитал: 536  
Читатель (660) прочитал: 889  
Читатель (660) прочитал: 476  
Читатель (660) прочитал: 413  
Читатель (660) прочитал: 988  
Сервер завершил работу и освободил ресурсы  
umahan@50cent:~$ |
```

При попытке подключиться третьему серверу, выводится ошибка:



```
umahan@50cent: ~  
umahan@50cent:~$ ./server_hw8  
Превышено максимальное число читателей
```

The image shows a terminal window with a dark background. The title bar at the top reads 'umahan@50cent: ~' and includes window control buttons. The terminal content shows the user 'umahan' at host '50cent' in the home directory. They have executed the command './server_hw8', which resulted in the Russian error message 'Превышено максимальное число читателей' (Maximum number of readers exceeded).