МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем

и технологий

**Отчет**

**по лабораторной работе № 16**

по дисциплине: ”Системное программирование”

на тему: ***”*** **ИСПОЛЬЗОВАНИЕ ОБЩИХ ФАЙЛОВ**, БУФЕРИЗАЦИЯ, СИНХРОНИЗАЦИЯ***”***

Выполнил**:** студент группы *10702121* Ахраменко Н. В.

Козлов М. А.

Василевский А. А.

Коржицкий Д. В.

Принял**:** пр. Мисякова В. А.

Минск 2023

# Лабораторная работа №16.

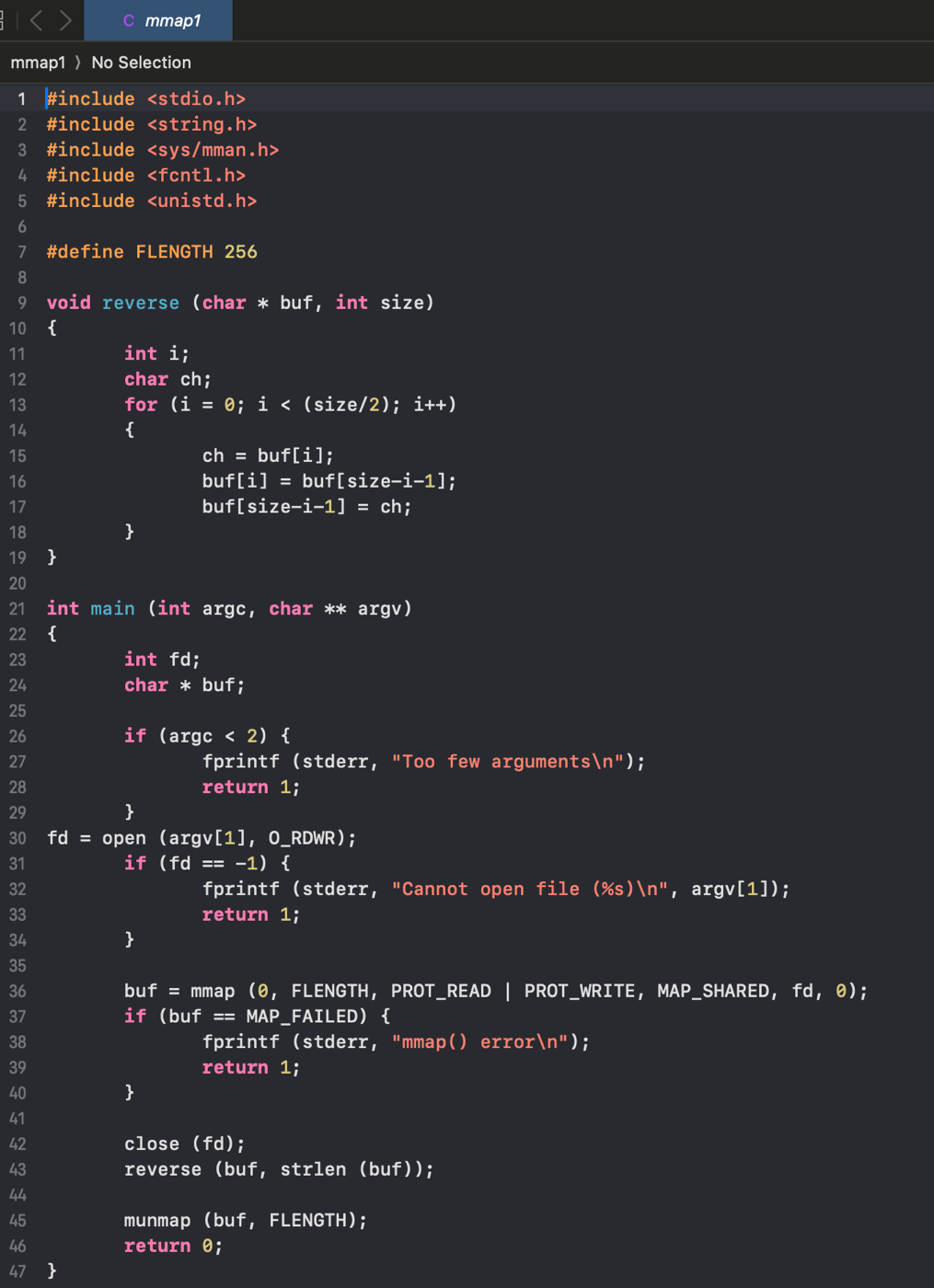
**Цель работы:** Освоить механизм взаимодействия между процессами на основе использования общей памяти.

## Задание 1

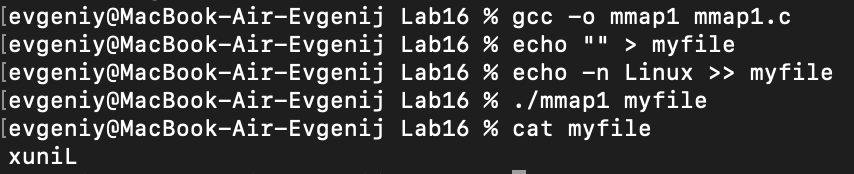
Создайте программу, которая демонстрирует использование системного вызова mmap(), перезаписывая содержимое файла "задом наперед".

**Решение**

**Программа mmap1.c**



**Проверяем работу в консоли**

****

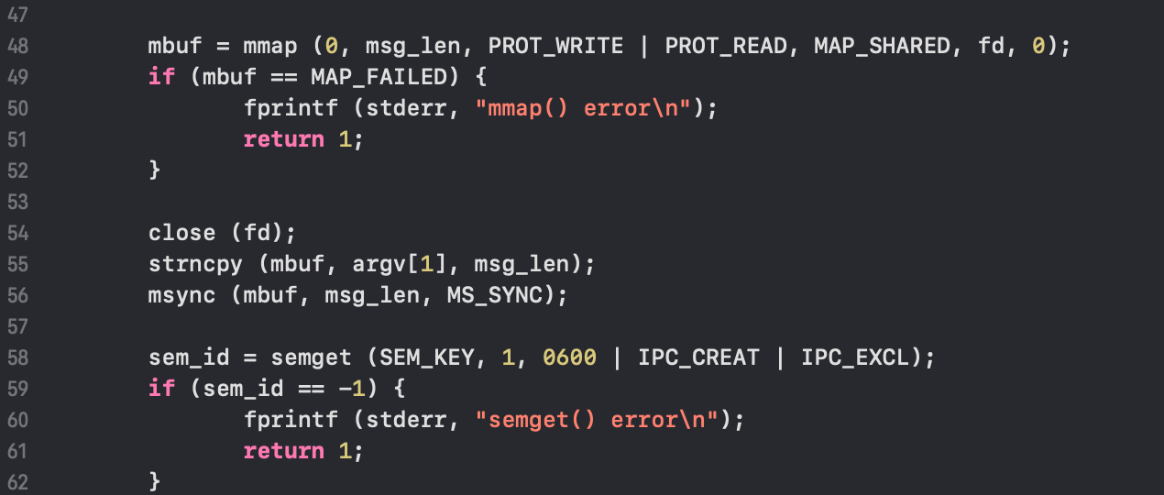
## Задание 2.

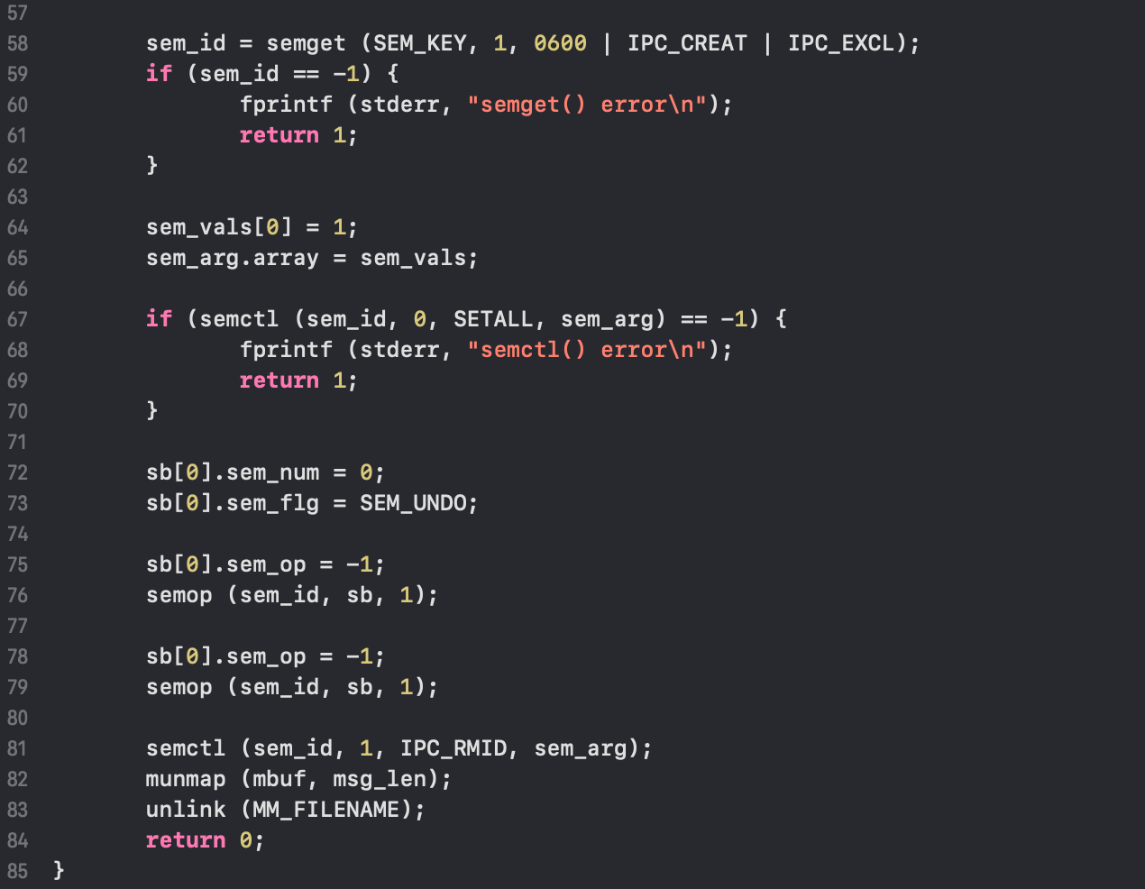
Организуйте межпроцессное взаимодействие на основе механизма отображения файлов в памяти. Для разделения доступа к общему файлу применяется семафор.

**Программа mmap2-owner.c**

В этом файле семафор используется для синхронизации доступа к общему файлу в памяти.

****

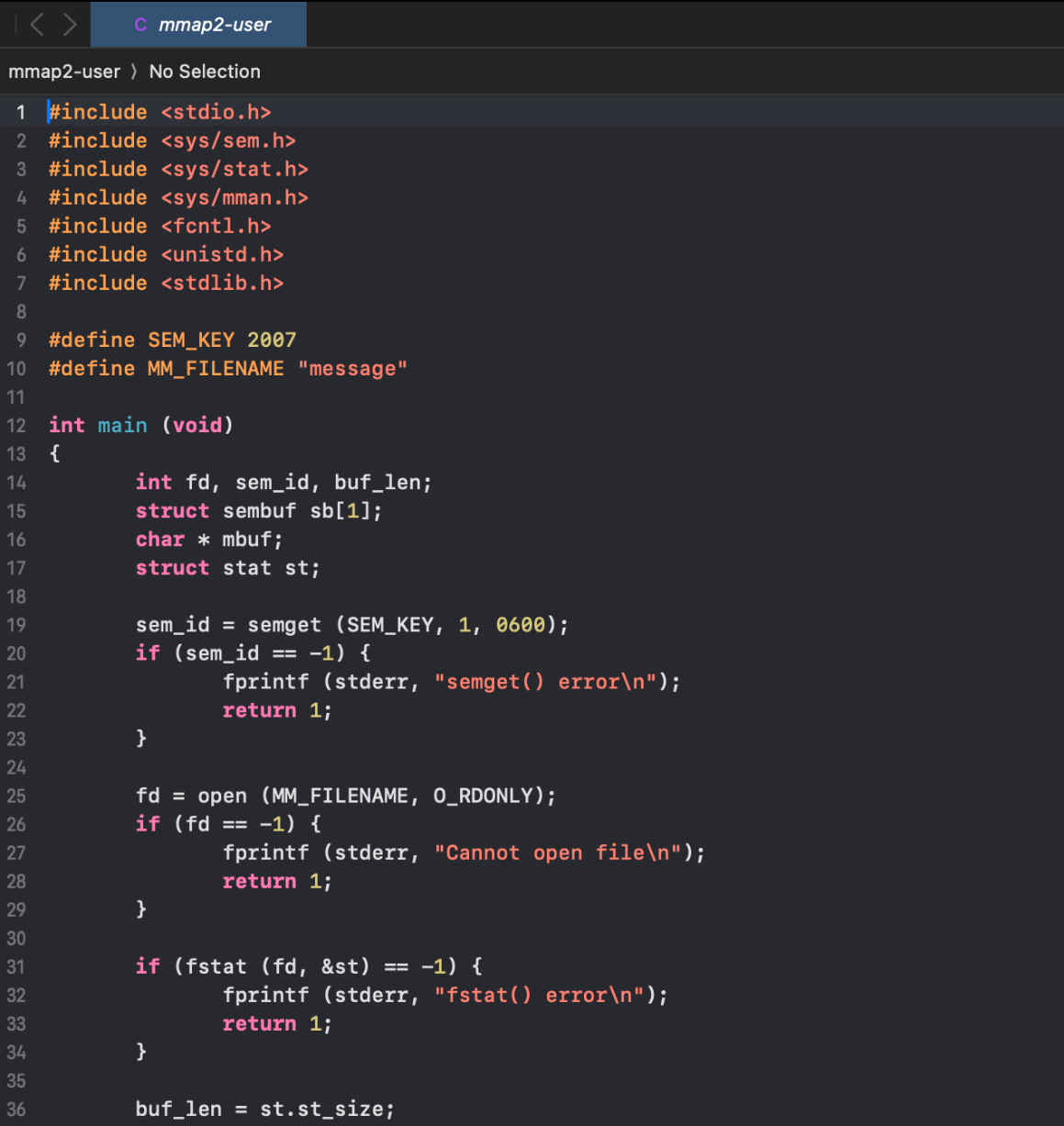
****

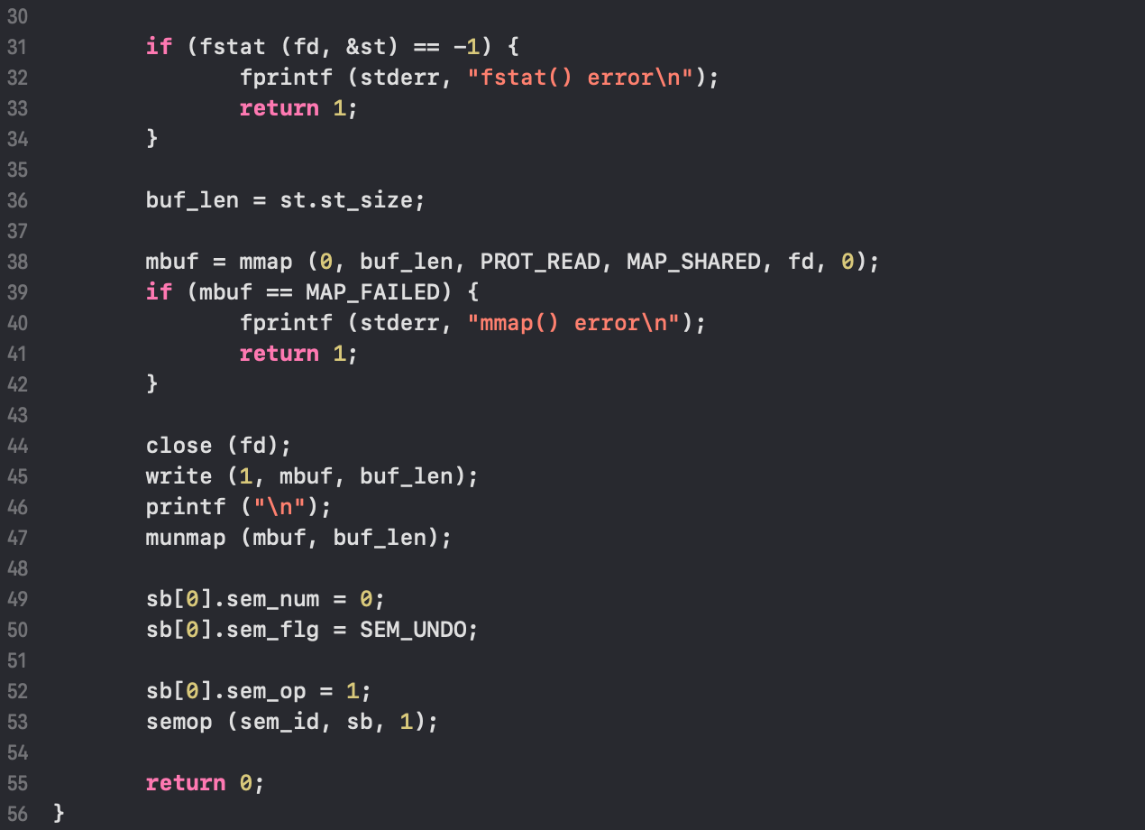
****

Здесь создается семафор с ключом SEM\_KEY, и затем семафор используется для управления доступом к разделяемому файлу в памяти. Сначала семафор уменьшается (*sem\_op = -1*), блокируя доступ других процессов.

**Программа mmap2-user.c**

В этом файле снова используется семафор для синхронизации.

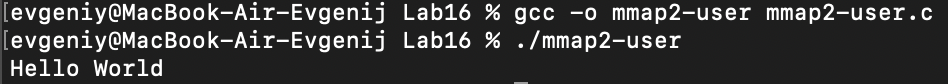
****

****

Здесь семафор увеличивается (*sem\_op = 1*), разблокируя доступ к разделяемому файлу в памяти для других процессов.

**Проверяем работу в консоли**





**Вывод:** Освоили механизм взаимодействия между процессами на основе использования общей памяти.

# Контрольные вопросы:

1. **Что такое *разделяемая память*?**

Разделяемая память - это область памяти, которую могут использовать несколько процессов. Это позволяет процессам обмениваться данными без использования промежуточных механизмов, таких как каналы IPC.

1. **Что такое *синхронизация процессов*?**

Синхронизация процессов - это механизм, который обеспечивает правильное выполнение нескольких процессов, которые могут взаимодействовать друг с другом. Это необходимо для предотвращения состояний гонки и обеспечения атомарности операций.

1. **Какие системные вызовы используются для работы с разделяемой памятью? Какие атрибуты они имеют, какие значения возвращают? Какие библиотеки необходимы для их подключения.**

Системные вызовы, используемые для работы с разделяемой памятью, включают shmget(), shmat(), shmdt() и shmctl(). Они используются для создания идентификатора разделяемого сегмента памяти, присоединения этого сегмента к адресному пространству процесса, отсоединения сегмента от адресного пространства и управления разделяемыми сегментами памяти. Для их использования необходимо подключить библиотеку sys/ipc.h и sys/shm.h.

1. **Что такое *семафоры*, для чего они используются?**

Семафоры - это переменные, используемые для контроля доступа к общим ресурсам. Они предотвращают одновременный доступ к ресурсу, что может привести к состоянию гонки.

1. **Какие системные вызовы используются для работы с семафорами? Какие атрибуты они имеют, какие значения возвращают?**

Системные вызовы, используемые для работы с семафорами, включают semget(), semop() и semctl(). Они используются для создания набора семафоров, выполнения операций над семафорами и управления набором семафоров.

1. **Опишите примерный сценарий работы с разделяемой памятью с использованием семафоров.**

Работа с разделяемой памятью с использованием семафоров обычно включает в себя следующие шаги: создание семафора с помощью semget(), создание разделяемой памяти с помощью shmget(), присоединение разделяемой памяти к адресному пространству процесса с помощью shmat(), изменение значения семафора с помощью semop() для обеспечения эксклюзивного доступа к разделяемой памяти, выполнение операций над разделяемой памятью, изменение значения семафора с помощью semop() для освобождения доступа к разделяемой памяти, отсоединение разделяемой памяти от адресного пространства процесса с помощью shmdt().

1. **Что Вы знаете о синхронизации и семафорах в языках более высокого уровня *C#* или *Java*?**

В языках высокого уровня, таких как C# и Java, семафоры и синхронизация также используются для координации действий многопоточных программ. Они обеспечивают контролируемый доступ к общим ресурсам и предотвращают состояния гонки.

1. **Что такое *динамически изменяемые структуры данных*, приведите примеры?**

Динамически изменяемые структуры данных - это структуры данных, которые могут изменять свой размер во время выполнения программы. Примеры включают связанные списки, стеки, очереди, хэш-таблицы и деревья.

1. **Какими средствами языка С обеспечивается динамическое выделение памяти?**

В языке C динамическое выделение памяти обеспечивается с помощью функций malloc(), calloc(), realloc() и free().

1. **Что такое «*утечка памяти*» и из-за чего она происходит?**

Утечка памяти - это ситуация, когда программа выделяет память, но не освобождает ее после использования. Это может привести к исчерпанию памяти и снижению производительности системы.

1. **Что такое «*Сборщик мусора*» в языках более высокого уровня *C#* или *Java*, по какому принципу (алгоритму) он работает.**

Сборщик мусора в языках высокого уровня, таких как C# или Java, автоматически освобождает память, которая больше не используется программой. Он работает по принципу отслеживания объектов, которые больше не доступны для программы, и освобождения памяти, занимаемой этими объектами.