**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

# Кафедра Вычислительной техники

**ОТЧЕТ**

# по лабораторной работе № 2.1

**по дисциплине «Операционные системы» Тема: «Управление файловой системой»**

Студент гр. 3311

Пасечный

Л.В.

Преподаватель

Тимофеев

А. В.

Санкт-Петербург

2025

# Введение

**Цель работы:**

Исследовать механизмы управления виртуальной памятью

# Постановка задачи:

Сервер создает файл на диске и проецирует его в память, далее записывает информацию и ждет,

пока клиент не прочтет, затем отменяет проецирование и удаляет файл.

Клиент открывает файл, проецирует и ждет доступности чтения, затем читает и выводит результат,

в завершении работы отменяет проецирование.

Сделать меню в каждой программе:

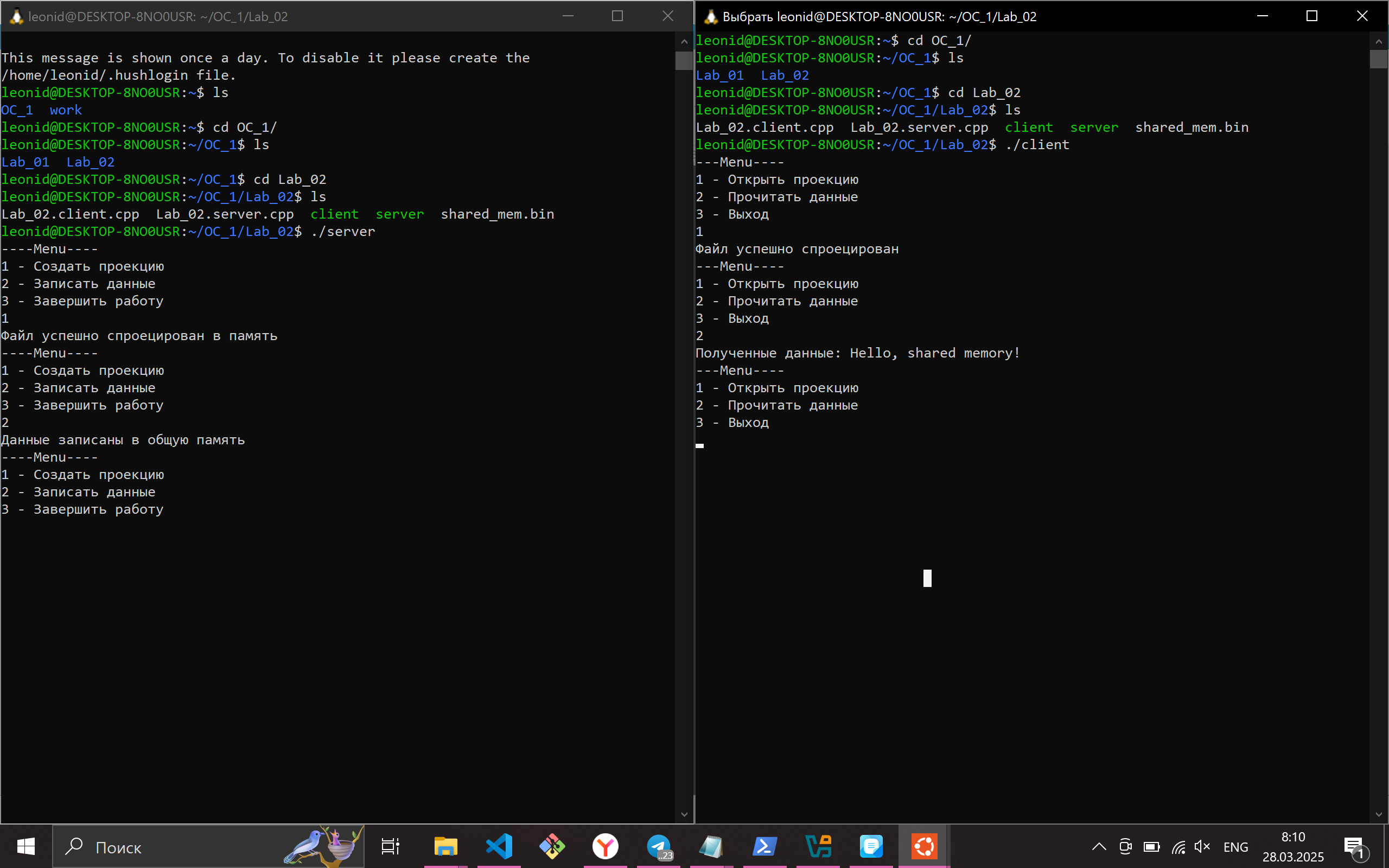
1. пункты меню Сервера – «выполнить проецирование», «записать данные», «завершить

работу»;

1. пункты меню Клиента – «выполнить проецирование», «прочитать данные», «завершить

# работу».

# Результаты:



**Заключение**

Мы научились проецировать данные в память комппьютера.

# Код программы

# Server:

#include <iostream>

#include <stdlib.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/mman.h>

#include <string.h>

#define FILENAME "shared\_mem.bin"

#define FILESIZE 1024

int fd = -1;

char \*ptr = NULL;

void create\_mapping(){

fd = open(FILENAME, O\_RDWR | O\_CREAT | O\_TRUNC, S\_IRUSR | S\_IWUSR);

if (fd == -1){

perror("Ошибка при создании файла");

exit(EXIT\_FAILURE);

}

// Установка размеров файла

if (ftruncate(fd, FILESIZE) == -1){

perror("Ошибка установки размеров файла");

exit(EXIT\_FAILURE);

}

// Проецирование файла в память

ptr = (char\*)mmap(NULL, FILESIZE, PROT\_READ | PROT\_WRITE, MAP\_SHARED, fd, 0);

if (ptr == MAP\_FAILED){

perror("Ошибка проецирования файла");

close(fd);

exit(EXIT\_FAILURE);

}

printf("Файл успешно спроецирован в память\n");

}

void write\_data(){

if (!ptr){

printf("Сначала выполните проецирование!");

return;

}

//Запись данных

sprintf(ptr, "Hello, shared memory!");

printf("Данные записаны в общую память\n");

}

void cleanup(){

if (ptr != NULL){

munmap(ptr, FILESIZE);

ptr = NULL;

}

if (fd == -1){

close(fd);

unlink(FILENAME);

fd = -1;

}

printf("Ресурсы освобождены. Файл удалён.\n");

}

void menu(){

printf("----Menu----\n");

printf("1 - Создать проекцию\n");

printf("2 - Записать данные\n");

printf("3 - Завершить работу\n");

}

int main(){

int choice = 0;

do{

menu();

scanf("%d", &choice);

switch(choice){

case 1:

create\_mapping();

break;

case 2:

write\_data();

break;

case 3:

cleanup();

break;

default:

printf("Попробуйте ещё раз\n");

break;

}

}while (choice != 3);

return 0;

}

**Client:**

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <unistd.h>

#include <sys/mman.h>

#include <sys/select.h>

#define FILENAME "shared\_mem.bin"

#define FILESIZE 1024

int fd = -1;

char \*ptr = NULL;

void open\_mapping(){

fd = open(FILENAME, O\_RDONLY);

if (fd == -1){

perror("Ошибка открытия файла");

exit(EXIT\_FAILURE);

}

// Проецирование файла

ptr = (char\*)mmap(NULL, FILESIZE, PROT\_READ, MAP\_SHARED, fd, 0);

if (ptr == MAP\_FAILED){

perror("Ошибка проецйирования файла");

close(fd);

exit(EXIT\_FAILURE);

}

printf("Файл успешно спроецирован\n");

}

void read\_data(){

if (!ptr){

printf("Сначала выполните проецирование!\n");

return;

}

//Ожидание доступности данных

fd\_set read\_fds;

FD\_ZERO(&read\_fds);

FD\_SET(fd, &read\_fds);

struct timeval timeout = {5, 0}; // Таймаут 5 секунд

int ready = select(fd + 1, &read\_fds, NULL, NULL, &timeout);

if (ready == -1){

perror("Ошибка select");

exit(EXIT\_FAILURE);

} else if (ready == 0){

printf("Данные недоступны (таймаут)\n");

return;

}

//Чтение данных

printf("Полученные данные: %s\n", ptr);

}

void cleanup(){

if (ptr != NULL){

munmap(ptr, FILESIZE);

ptr = NULL;

}

if (fd == -1){

close(fd);

fd = -1;

}

printf("Ресурсы клиента освобождены\n");

}

void menu(){

printf("---Menu----\n");

printf("1 - Открыть проекцию\n");

printf("2 - Прочитать данные\n");

printf("3 - Выход\n");

}

int main(){

int choice = 0;

do{

menu();

scanf("%d", &choice);

switch(choice){

case 1:

open\_mapping();

break;

case 2:

read\_data();

break;

case 3:

cleanup();

break;

default:

printf("Попробуйте ещё раз!");

break;

}

}while (choice != 3);

return 0;

}