Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Институт «Электронных и информационных систем»

Кафедра «Информационных систем и технологий»

**ОРГАНИЗАЦИЯ FS В UNIX. РАБОТА С ФАЙЛАМИ. ПОНЯТИЕ О MPF**

Лабораторная работа №8 по учебной дисциплине «Операционные системы»

По направлению 09.03.01 «Информатика и вычислительная техника»

Отчёт

Принял преподаватель:

\_\_\_\_\_\_\_\_\_ Ананьев В. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Выполнил студент группы 8091:

\_\_\_\_\_\_\_ Григорьев Д. И.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Великий Новгород

2020

**Цель работы:** научиться работать с файловой системой в UNIX-like системах.

**Задание:**

Написать программу для работы с файлами, проецируемыми в память.

В параметрах командной строки программа должна получить имена двух файлов.

Первый файл должен содержать входные данные - короткий текст на английском языке. Достаточно нескольких предложений. (Например, их можно скопировать со страницы справки man mmap, или с любой страницы из интернета.)

Второй файл (выходной) требуется открыть, если надо, то создать, и спроецировать в память текущего процесса.

Программа должна прочитать входной файл, размещая получаемые данные в памяти, в которую спроецирован выходной файл.

В конце работы требуется освободить все выделенные ресурсы и закрыть оба файла.

В отчете привести исходный код программы, а также содержимое и размеры обоих файлов после окончания работы программы.

(Для просмотра содержимого бинарных файлов можно использовать утилиту hexdump с параметром -C).

**Реализация:**

#include <stdio.h>

#include <sys/mman.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

*int* main(*int* *argc*, *char* \**argv*[])

{

    const *char*\* input\_file =  argv[1];

    const *char*\* output\_file = argv[2];

*int* fmapi = open(input\_file, O\_RDWR, 0600);

*int* fmapo = open(output\_file, O\_CREAT | O\_RDWR, 0600);

    if (fmapi < 0 || fmapo < 0)

    {

        perror("error with open of map file");

        return -1;

    }

*struct* stat\* t\_stat;

    fstat(fmapi, t\_stat);

*int* dataSize = t\_stat->st\_size;

*char*\* data = (*char* \*)mmap(NULL, dataSize, PROT\_WRITE | PROT\_READ, MAP\_SHARED, fmapi, 0);

    if (MAP\_FAILED == data)

    {

        perror("error with mmap");

        return -2;

    }

    lseek(fmapo, 0, SEEK\_SET);

    write(fmapo, data, dataSize);

    ftruncate(fmapo, dataSize);

    close(fmapi);

    close(fmapo);

}

**Содержимое input-файла:**

NAME

mmap, munmap - map or unmap files or devices into memory

SYNOPSIS

#include <sys/mman.h>

void \*mmap(void \*addr, size\_t length, int prot, int flags,

int fd, off\_t offset);

int munmap(void \*addr, size\_t length);

See NOTES for information on feature test macro requirements.

**Содержимое output-файла:**

NAME

mmap, munmap - map or unmap files or devices into memory

SYNOPSIS

#include <sys/mman.h>

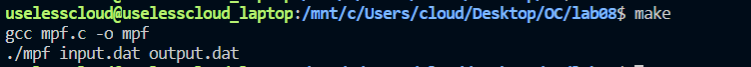
void \*mmap(void \*addr, size\_t length, int prot, int flags,

int fd, off\_t offset);

int munmap(void \*addr, size\_t length);

See NOTES for information on feature test macro requirements.

**Результат выполнения программы:**

****

**Вывод:** В процессе выполнения лабораторной работы я научился работать файловой системой в UNIX-like системах.