

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

**Лабораторная работа №4 по курсу
«Операционные системы»**

ФАЙЛОВЫЕ СИСТЕМЫ

Студент: Армишев Кирилл Константинович

Группа: М8О–208Б–22

Вариант: 12

Преподаватель: Соколов Андрей Алексеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2022.

Постановка задачи

Цель работы

Приобретение практических навыков в:

- Освоение принципов работы с файловыми системами
- Обеспечение обмена данными между процессами посредством технологии «File mapping»

Задание

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов.

Взаимодействие между процессами осуществляется через системные сигналы/события и/или через отображаемые файлы (memory-mapped files).

Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

Общие сведения о программе

Программа состоит из одного файла main.c. В ходе работы программы создается 2 дочерних процесса, первый переводит все символы в верхний регистр, второй заменяет двойные пробелы на одинарные. Межпроцессорное взаимодействие осуществляется благодаря сигналам.

Системные вызовы:

1. **kill()** - посылает сигнал процессу
2. **waitpid()** - дождаться завершения процесса
3. **mmap()** - отразить файл в память

Общий метод и алгоритм решения.

Так как большинство функционала для работы с лабораторной работой было реализовано еще во второй лабораторной работе, то нужно лишь перестроить способ межпроцессорного взаимодействия и использовать mmap

Для реализации поставленной задачи необходимо:

1. Изучить принцип работы mmap
2. Познакомиться с принципом работы сигналов
3. Организовать межпроцессорное взаимодействие с использованием сигналов
4. Провести тестирование

Основные файлы программы

main.c:

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <ctype.h>
#include <sys/wait.h>

void *spaces(char* str, int size) {
    int i, x;
    for(i=x=0; i< size; ++i)
        if(!isspace(str[i]) || (i > 0 && !isspace(str[i-1])))
            str[x++] = str[i];
    str[x] = '\0';
    return str;
}

int main(){
    int FILE = open("./IN.txt",O_CREAT | O_RDWR,S_IRUSR | S_IWUSR);
    truncate("./IN.txt", 0);
    struct stat sb;
    char c[100];
    while(fgets(c,sizeof(c),stdin)) {
        int n = strlen(c);
        write(FILE, &c, n);
    }
    if(fstat(FILE,&sb)==-1){
        perror("couldn't get file size.\n");
    }
    char *file_in_memory = mmap(NULL, sb.st_size,PROT_READ | PROT_WRITE,
```

```

MAP_SHARED, FILE,0);
int id = fork();
if (id == -1)
{
    perror("fork error");
    return 0;
}else if(id == 0) {
    for(int i=0; i<sb.st_size; i++){
        file_in_memory[i]=toupper(file_in_memory[i]);
    }
    return 0;
}
int id2;
if(id > 0) {
    id2 = fork();
    if (id2 == -1)
    {
        perror("fork2 error");
        return 0;
    }else if(id2 == 0) {
        spaces(file_in_memory,sb.st_size);
    }
}
if(id != 0 && id2 != 0) {
    wait(NULL);
    wait(NULL);
    for(int i=0; file_in_memory[i]; ++i){
        printf("%c", file_in_memory[i]);
    }
    printf("\n");
    munmap(file_in_memory, sb.st_size);
}

return 0;
}

```

Пример работы

-----test.txt-----

```

oijooij jnojoDJKDKojo nojoio
fkrkfkfkk fkrkfkfdll kdfkfkfkf
llllf frfrfff rkrkrkrlllc
vfkfk ddf

```

```
kirillarmishev@2 Lab4 % gcc main.c -o lab4
```

```
kirillarmishev@2 Lab4 % ./lab4 < test.txt
```

```
OIJOOIJ JNOJODJKDKOJO NOJOIO
```

```
FKRKFKFKK FKFKFKKFDLL KDFKFKKFKF
```

```
LLLLF FRFRFFF RKRKRKRLLLC
```

```
VFKFK DFDF
```

Вывод

В данной лабораторной я познакомился с механизмами работы с файловыми системы, в частности с системным вызовом `mmap`. Также я реализовал межпроцессорное взаимодействие с помощью сигналов. Благодаря данной лабораторной работе я понял, что существует несколько способов организации общения процессов, выбор которого зависит от конкретной задачи.