

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №6
за 4 семестр
По дисциплине: «ОСиСП»
Тема: «Средства межпроцессного взаимодействия»

Выполнила:
Студентка 2 курса
Группы ПО-3(2)
Овсейчук В.И.
Проверил:
Давидюк Ю.И.

Лабораторная работа №6

Вариант 19

14	Общие файлы	Родитель передает потомку число, потомок возвращает его факториал
----	-------------	---

Содержимое lab6.c

```
#include <stdio.h>
#include <sys/mman.h>
#include <unistd.h>
#include <fcntl.h>
#include <signal.h>
#include <stdlib.h>
int factorial(int chislo)
{
    if (chislo < 0)
        return 0;
    if (chislo == 0)
        return 1;
    else return chislo * factorial(chislo - 1);
}

void handler(int i)
{
    printf("parent sent signal to the child process \n");
}

int main(void) {
    signal(SIGUSR1, handler);
    int fd; size_t pagesize = getpagesize();
    int *region;
    int chislo;
    printf("Input value: ");

    scanf("%i", &chislo);
    //create file for shared access
    if ((fd = open("myfile", O_RDWR | O_CREAT | O_TRUNC, 0777)) < 0) {
        printf("Can't open file\n");
        return -1;
    }
    //empty file is not mapping
    write(fd, "/0", sizeof(char));
    region = (int*)mmap(
        NULL, // Map from the start of the null page
        pagesize, // for one page length
        PROT_READ | PROT_WRITE, // to a private block of
        // hardware memory
        MAP_SHARED,
        fd, 0);
    if (region == MAP_FAILED) {
        perror("Could not mmap");
        return -1;
    }

    //create process
    int result = fork();
    if (result < 0) {
        printf("Can't fork child\n");
        return -1;
    }
}
```

```

}
else if (result > 0) {
    //parent process

    region[0] = chislo;
    printf("Shared file (in parent): ");
    printf("%i \n", region[0]);

    //parent process is stopped, jumping into child process
    kill(result, SIGUSR1);
    sleep(1);

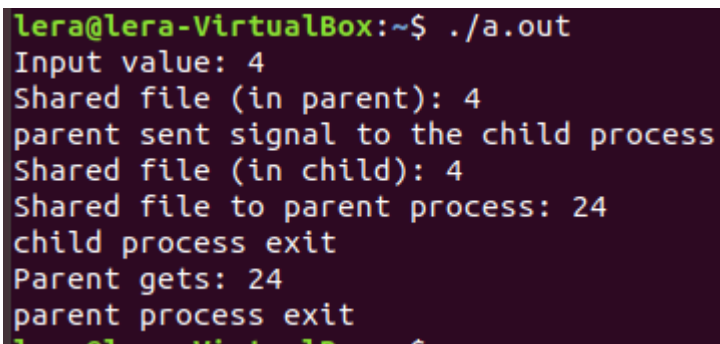
    printf("Parent gets: ");
    printf("%i\n", region[0]);
    if (munmap(region, pagesize) < 0)
    {
        perror("Could not munmap");
        return -1;
    }
    printf("parent process exit\n");
}
else {
    //child process

    printf("Shared file (in child): ");
    printf("%i \n", region[0]);
    region[0] = factorial(region[0]); //finding factorial
    printf("Shared file to parent process: %i \n", region[0]);

    printf("child process exit\n");
    exit(0);
}
return 0;
}

```

Вывод программы:



```

lera@lera-VirtualBox:~$ ./a.out
Input value: 4
Shared file (in parent): 4
parent sent signal to the child process
Shared file (in child): 4
Shared file to parent process: 24
child process exit
Parent gets: 24
parent process exit
lera@lera-VirtualBox:~$

```

Вывод: Изучила средства межпроцессорного взаимодействия.