

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

**Лабораторная работа №6**  
**«Средства межпроцессного взаимодействия»**  
По дисциплине «ОСиСП»  
За четвертый семестр

**Выполнила:**  
Студентка 2 курса  
Факультета ЭИС  
Группы ПО-3  
Дмитрук В.Г.  
**Проверила:**  
Давидюк Ю.И.

**Брест 2020 г.**

**Цель работы:** изучить средства межпроцессорного взаимодействия.

9	Общие файлы	Родитель передает потомку строку с различными символами, потомок возвращает только цифры, если цифр нет, то возвращает слово «No»
---	-------------	---

**Текст программы:**

```
//Родитель передает потомку строку с различными символами, потомок
возвращает только цифры, если цифр нет, то возвращает слово «No»
#include <stdio.h>
#include <sys/types.h>
#include <sys/msg.h>
#include <sys/ipc.h>
#include <string.h>
#include <unistd.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/mman.h>

void handler(int i)
{
    signal(SIGUSR1, handler);
    sleep(1);
    printf("\nРодительский процесс отправил SIGUSR1 дочернему!\n\n");
}

int main(void){
    size_t pagesize = getpagesize();
    char numbers[10] = "1234567890";
    int fd, result;
    char *region;
    (void)umask(0);
    printf("System page size: %zu bytes\n", pagesize);
    //создание файла для общего доступа
    fd = open("text.txt", O_RDWR|O_CREAT, 0777);
    if(fd < 0)
    {
        printf("Can\'t open\n");
        return -1;
    }

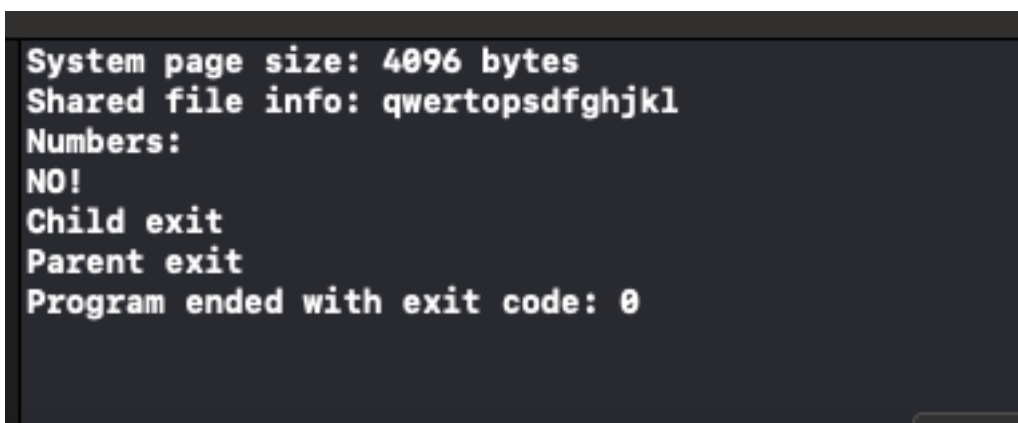
    write (fd, "/0", sizeof(char));
    region=(char*)mmap(NULL,pagesize,PROT_READ|PROT_WRITE, MAP_SHARED,
fd, 0); //сделали общий файл
    //если не получилось сделать map
    if (region == MAP_FAILED)
    {
        perror("Could not mmap");
        return -1;
    }
    char *stroka="qwertyop2sdfghjkl";
    region = stroka;
```

```

result=fork();
if(result<0)
{
    printf("Can't fork child\n");
    return -1;
}
else if (result > 0) //родитель
{
    close(fd);
    kill(result,SIGUSR1);
    sleep(3);
    printf("Parent exit\n");
}
else //дочерний
{
    signal(SIGUSR1, handler); //дочерний процесс получил сигнал от
родительского
    sleep(1);
    char *s = region;
    char str[] = ""; //строка, которую будем выводить
    printf("Shared file info: %s \n",s);
    printf("Numbers: ");
    for(int i=0; i<strlen(s);i++){
        if(strchr(numbers, s[i]) != NULL){ //если символ - цифра
            printf("%c",s[i]);
            strcpy(str, &s[i]); //копируем
        }
    }
    printf("\n");
    if(strcmp(str, "") == 0){
        printf("NO!\n"); //если в строке нет цифр
    }
    printf("Child exit\n");
    exit(0);
}
return 0;
}

```

Результат:



```

System page size: 4096 bytes
Shared file info: qwertopsdfghjkl
Numbers:
NO!
Child exit
Parent exit
Program ended with exit code: 0

```

Примечание: Программа работает корректно, но ,из-за особенностей ОС, не всегда.

Вывод: Изучила средства межпроцессорного взаимодействия.