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

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

Выполнил:
Студент 2 курса
Группы ПО-3
Новикевич А.А.
Проверил:
Давидюк Ю.И.

Брест 2020

Лабораторная работа №6

Вариант 18. Разделяемая память. Родитель передаёт потомку число, потомок возвращает следующее за ним число Фибоначчи.

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/shm.h>
#include <sys/mman.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
#include <signal.h>
#define SHMNAME "my_shm"

void dochernij();
int main() {
int rod = getpid();
printf("PID процесса: %d\n", rod);
int SIZE = 2048;
int fd;
void *memory;
fd = shm_open(SHMNAME, O_CREAT | O_RDWR, 0777);
if(fd < 0)
{
printf("cant open file\n");
return -1;
}
ftruncate(fd, SIZE);
memory = mmap(0, SIZE, PROT_WRITE, MAP_SHARED, fd, 0);
int pid = fork();
if (pid == 0) {
signal(SIGUSR1, dochernij);
pause();
} else {
char chislo[100];
printf("Введите число: ");
scanf("%s", chislo);
strcat(memory, chislo);
printf("Родителлль отправил: %s\n", (char *)memory);
kill(pid, SIGUSR1);
waitpid(pid, NULL, 0);
printf("Родитель принял: %s\n", (char *)memory);
shm_unlink(SHMNAME);
}
return 0;
}
```

```c
void dochernij() {
int doch = getpid();
printf("PID процесса: %d\n", doch);
int SIZE = 1000;
int fd;
void *memory;
fd = shm_open(SHMNAME, O_CREAT | O_RDWR, 0666);
ftruncate(fd, SIZE);
memory = mmap(0, SIZE, PROT_WRITE, MAP_SHARED, fd, 0);
printf("Ребёнок принял: %s\n", (char *)memory);
int zadannoe = atoi(memory);
// printf("Оно целое число???????%lu\n", sizeof(zadannoe));
int first = 1;
int second = 0;
int temp = 0;
int rezult;
for(int i = 0; i<1000; i++) // Нахождение последующего числа Фибоначчи
{
if(zadannoe < second){
printf("Следующее число Фибоначчи %d\n", second);
rezult = second;
break;
}
first +=second;
second = first - second;
// printf(" %d ",second);
}
char rez[100];
sprintf(rez, "%d", rezult);
strcpy(memory,rez);
printf("Процесс %d отправил число: %d\n", getpid(), rezult);
printf("Процесс %d завершается\n", getpid());
}
```

Результат выполнения работы:

```
artyom@AN:~/OS/lab6$ gcc laba6.c -o 1.out -lrt
artyom@AN:~/OS/lab6$ ./1.out
PID процесса: 5641
Введите число: 3
Родителль отправил: 3
PID процесса: 5642
Ребёнок принял: 3
Следующее число Фибоначчи 5
Процесс 5642 отправил число: 5
Процесс 5642 завершается
Родитель принял: 5
```

```
artyom@AN:~/OS/lab6$ gcc laba6.c -o 1.out -lrt
artyom@AN:~/OS/lab6$ ./1.out
PID процесса: 5794
Введите число: 65
Родителль отправил: 65
PID процесса: 5795
Ребёнок принял: 65
Следующее число Фибоначчи 89
Процесс 5795 отправил число: 89
Процесс 5795 завершается
Родитель принял: 89
```

Вывод: изучил средства межпроцессного взаимодействия.