Федеральное агентство по образованию Российской Федерации

Новгородский государственный университет имени Ярослава Мудрого

Кафедра [Кафедра информационных технологий и систем](http://www.novsu.ru/dept/1238)

**Процессы в ОС UNIX**

**Отчет по лабораторной работе #4**

Выполнил

студент гр. 8091

Васильев Иван Владимирович

«» 2020 г.

Принял

[Владислав Валерьевич Ананьев](http://people.novsu.ru/profiles/html/profileView.do?key=709f5d58-6b4b-463c-b2a3-81ef40ff7202)

«\_\_»\_\_\_\_\_ 2020 г.

Великий Новгород

2020

## Задание на лабораторную работу

Объединить программы первой части предыдущей лабораторной в одну программу.

Вначале программа должна сгенерировать 20 случайных чисел, разместить их в разделяемой памяти и вывести на экран.

Затем требуется породить новый процесс, который выполнит сортировку самостоятельно, без запуска других программ (в отличие от предыдущей лабораторной).

Второй процесс должен отсортировать числа и также вывести их на экран.

Первый процесс должен дождаться завершения работы второго, после чего самостоятельно выполнить освобождение выделенной разделяемой памяти и всех прочих выделенных ресурсов (при наличии таковых).

**Код программы:**

#include <stdio.h>

#include <stdlib.h>

#include <sys/shm.h>

#include <time.h>

#include <sys/wait.h>

#include <unistd.h>

int comp(const int \*a, const int \*b);

void FunctionSort(int \*mem);

void CreatAndPrintArray(int \*mem);

int main()

{

srand(time(NULL));

int memId = shmget(IPC\_PRIVATE, sizeof(int), 0600 | IPC\_CREAT | IPC\_EXCL);

int \*mem = (int \*)shmat(memId, 0, 0);

CreatAndPrintArray(mem);

printf("Sorted array:\n");

int sonId;

sonId = fork();

if (sonId == 0)

{

FunctionSort(mem);

}

else

{

waitpid(sonId, NULL, 0);

}

shmdt(mem);

shmctl(memId, 0, IPC\_RMID);

return 0;

}

void CreatAndPrintArray(int \*mem)

{

printf("Original array:\n");

for (size\_t i = 0; i < 20; i++)

{

mem[i] = rand() % 100;

}

for (size\_t i = 0; i < 20; i++)

{

printf("%d ", mem[i]);

}

printf("\n");

}

int comp(const int \*a, const int \*b)

{

return \*a - \*b;

}

void FunctionSort(int \*mem)

{

qsort(mem, 20, sizeof(int), comp);

for (size\_t i = 0; i < 20; i++)

{

printf("%i ", mem[i]);

}

printf("\n");

}

**Выполнение программы:**

