Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Институт «Электронных и информационных систем»

Кафедра «Информационных систем и технологий»

**ПРОЦЕССЫ В ОС UNIX**

Лабораторная работа №4 по учебной дисциплине «Операционные системы»

По направлению 09.03.01 «Информатика и вычислительная техника»

Отчёт

Принял преподаватель:

\_\_\_\_\_\_\_\_\_ Ананьев В. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Выполнил студент группы 8091:

\_\_\_\_\_\_\_ Бочкарев Б. А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Великий Новгород

2020

**Цель работы:** научиться работать с процессами в UNIX-like системах.

**Задание:**

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).

**Реализация:**

#include <stdio.h>

#include <sys/wait.h>

#include <sys/shm.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

const int chisl = 20;

int func(const void\* x, const void\* y)

{

return (\*((int\*)x) - \*((int\*)y));

}

void sort (int\* arr)

{

qsort(arr, chisl, sizeof(int), func);

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

{

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

}

}

int main()

{

srand(time(NULL));

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

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

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

{

arr[i] = rand() % 51;

}

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

{

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

}

printf("\n");

int child\_id = fork();

if (child\_id == 0)

{

sort(arr);

}

else

{

waitpid(child\_id, NULL, 0);

}

shmdt(arr);

return 0;

}

**Результат выполнения программы:**



**Вывод:** В процессе выполнения лабораторной работы я научился работать с процессами.