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:

\_\_\_\_\_\_\_ Кудряшов И. С.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Великий Новгород

2020

**Цель работы:** познакомиться с механизмами разделённой памяти и многопоточности.

**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).

**Содержание файла processmaker.c:**

#include <stdio.h>

#include <sys/wait.h>

#include <sys/shm.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

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

{

return \*a - \*b;

}

void sortAndPrint(int \*mem)

{

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

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

{

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

}

printf("\n");

}

int main()

{

srand(time(NULL));

int memId = shmget(IPC\_PRIVATE, sizeof(int) \* 20, 0600 | IPC\_CREAT | IPC\_EXCL);

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

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

{

numbers[i] = rand() % 10000;

}

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

{

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

}

printf("\n");

int child\_id = fork();

if (child\_id == 0)

{

sortAndPrint(numbers);

}

else

{

waitpid(child\_id, NULL, 0);

}

shmdt(numbers);

shmctl(memId, 0, IPC\_RMID);

return 0;

}

**Результат, выведенный на экран:**

make

gcc processmaker.c -o processmaker

processmaker.c: In function ‘sortAndPrint’:

processmaker.c:15:33: warning: passing argument 4 of ‘qsort’ from incompatible pointer type [-Wincompatible-pointer-types]

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

^~~~

In file included from processmaker.c:4:0:

/usr/include/stdlib.h:827:13: note: expected ‘\_\_compar\_fn\_t {aka int (\*)(const void \*, const void \*)}’ but argument is of type ‘int (\*)(const int \*, const int \*)’

extern void qsort (void \*\_\_base, size\_t \_\_nmemb, size\_t \_\_size,

^~~~~

./processmaker

6668 5866 5177 1160 2920 4875 4250 2952 9024 2566 5581 8695 8124 5463 9176 9573 4869 6707 7973 9732

1160 2566 2920 2952 4250 4869 4875 5177 5463 5581 5866 6668 6707 7973 8124 8695 9024 9176 9573 9732

**Вывод:** В процессе выполнения лабораторной работы я на практике познакомился с механизмами запуска новых процессов в UNIX.