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»

**Организация взаимодействия процессов через pipe и FIFO в UNIX**

Лабораторная работа №5 по учебной дисциплине «Операционные системы»

Отчёт

Принял преподаватель:

\_\_\_\_\_\_\_\_\_ Ананьев В. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

Выполнил студент группы 9091:

\_\_\_\_\_\_\_ Ковалев А.Д.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

Великий Новгород

2020

**Цель лабораторной работы**

Цель работы: познакомиться с механизмами организации взаимодействия процессов через pipe и FIFO в UNIX.

**Исходный текст программы**

#include <stdio.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <unistd.h>

#include <time.h>

#include <fcntl.h>

const char \*FIFO\_NAME = "FifoFile";

int compare\_ints(const void \*a, const void \*b)

{

return (\*((int \*)b) - \*((int \*)a));

}

int \*RandomNumbers(int n)

{

srand((unsigned)(time(0)));

int \*nums = malloc(sizeof(int) \* n);

for (int i = 0; i < n; i++)

{

nums[i] = rand() % 100000;

}

return nums;

}

void PrintNumbers(int \*nums, int c)

{

for (int i = 0; i < c; i++)

{

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

}

printf("\n");

}

int main(int argv, char \*argc[])

{

int n = atoi(argc[1]);

int \*nums = RandomNumbers(n);

PrintNumbers(nums, n);

mknod(FIFO\_NAME, S\_IFIFO | 0666, 0);

int p[2];

pipe(p);

int child\_id = fork();

if (child\_id == 0)

{

close(p[0]);

int fifo = open(FIFO\_NAME, O\_RDONLY);

int \*received\_nums = malloc(sizeof(int) \* n);

read(fifo, received\_nums, sizeof(int) \* n);

close(fifo);

qsort(received\_nums, n, sizeof(int), compare\_ints);

write(p[1], received\_nums, sizeof(int) \* n);

close(p[1]);

}

else

{

close(p[1]);

int fifo = open(FIFO\_NAME, O\_WRONLY);

write(fifo, nums, sizeof(int) \* n);

close(fifo);

int \*sorted\_nums = malloc(sizeof(int) \* n);

read(p[0], sorted\_nums, sizeof(int) \* n);

PrintNumbers(sorted\_nums, n);

close(p[0]);

unlink(FIFO\_NAME);

}

}

**Результат выполнения программы**

>> make

Gcc Laba5.c -o Laba5

./Laba5 10

24417 83420 43289 85002 55778 97734 10387 34858 99498 98885

99498 98885 97734 85002 83420 55778 43289 34858 24417 10387

**Вывод**

Выполняя лабораторную работу, я познакомился с механизмами организации взаимодействия процессов через pipe и FIFO в UNIX.