Федеральное агентство связи

Сибирский Государственный Университет Телекоммуникаций и Информатики

## Кафедра ТС и ВС

**Лабораторная работа № 8**

**По дисциплине: Операционные системы**

**Выполнил**: Зарубин Максим Евгеньевич

**Группа**: ИА-831

## Вариант: 5

**Проверила**: Моренкова Ольга Ильинична

Новосибирск, 2020 г

**Программирование в ОС Linux. Взаимодействие «родственных» процессов.**

**Задание на лабораторную работу.**

Написать программу на языке Си, выполняющую следующие действия:

- создание программного канала и порождение дочернего процесса;

- основной процесс считывает содержимое файла, созданного в лабораторной работе №1, и передаёт его в канал;

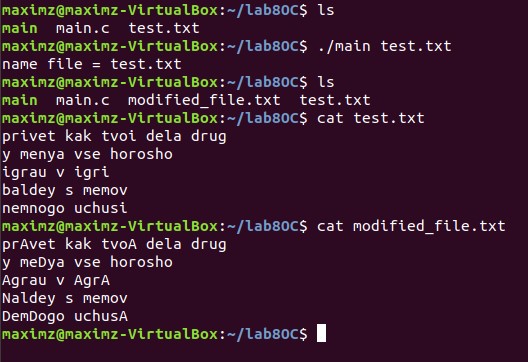
- дочерний процесс получает информацию через канал и выводит эту информацию в файл с новым именем

- передача завершается при поступлении в канал символа с кодом 26.

- при получении информации дочерний процесс производит замену символов по схеме: b N, i A, n D.

**Результаты работы программы.**

Для работы программы я использовал содержимое нового файла, а не из лабораторной работы №1, потому что там у меня всё было на русском языке. С помощью ls показал содержимое папки до работы программы и после. С помощью cat показал содержимое файлов до и после работы программы.



**Листинг программы.**

**Main.c:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

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

FILE \*a, \*b;

char ch;

pid\_t pid;

int potok[2];

printf("name file = %s\n",argv[1]);

if(pipe(potok) < 0){

printf("Pipe failed\n");

exit(-1);

}

pid = fork();

if(pid == -1){

perror("fork");

exit(1);

}

else if(pid == 0){

close(potok[1]);

b = fopen("modified\_file.txt","w");

while(!feof(b) && ch!=EOF){

read(potok[0], &ch, sizeof(char));

if(ch!=EOF){

if(ch == 'b') ch='N';

if(ch == 'i') ch='A';

if(ch == 'n') ch='D';

putc(ch, b);

}

}

close(potok[0]);

fclose(b);

sleep(1);

}

else{

close(potok[0]);

a = fopen(argv[1],"r");

while(!feof(a) && ch!=EOF) {

ch=fgetc(a);

write(potok[1], &ch, sizeof(char));

}

close(potok[1]);

fclose(a);

sleep(1);

}

return (0);

}