**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ» (МАИ)**

Институт №3 «Системы управления, информатика и электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

**Лабораторная работа №3**

по дисциплине

**«Операционные системы»**

На тему:

**«Управление процессами»**

Выполнили студенты группы М30-325Бк-18

Назарчук Г. В.,

Черноморец М. С.

Проверил:

доцент кафедры 304, к.т.н. Охотников Д. А.

Москва 2021

**Задание**

1. Написать программу «терминал», которая анализирует входную строку и при обнаружении ключевых слов “ls” и “cat” запускает соответствующие процессы.
2. Программа получает сигнал нажатия комбинации Ctrl + C и останавливает запущенную программу.
3. Заменить системный bash на собственный терминал.

**Описание программы**

Программа принимает команду и аргументы к ней как параметры при вызове. Если команда оказывается ls или cat, выводится сообщение об этом, и она выполняется функцией exec в дочернем процессе. Если команда другая, то выводится сообщение об этом и команда передаётся в функцию exec в дочернем процессе (это сделано для того, чтобы не парализовать работу через терминал при замене терминала на эту программу). При нажатии комбинации клавиш Ctrl + C сигнал перехватывается и передаётся дочернему процессу.

Код программы доступен в репозитории по адресу: https://github.com/CherMiTT/lab3\_terminal

**Код программы**

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <signal.h>

void onCtrlC(int sig);

pid\_t pid;

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

{

printf("Started programm\n");

char input[1024];

char\* ptr = NULL;

char\* v\_args[32];

char tmp[32][128];

while(1)

{

if(fgets(input,1023, stdin) == NULL)

{

printf("Some error has occured\n");

return 0;

}

input[1023] = '\0';

if(!strncmp("exit", input, 4)) //if first 4 characters is "exit"; all other words are irrelevant

{

printf("Exiting application\n");

return 0;

}

ptr = strtok(input, " \n\0");

strcpy(tmp[0], ptr);

v\_args[0] = tmp[0];

printf("First word = %s\n", tmp[0]);

int i = 1;

while ((ptr = strtok(NULL, " \n")))

{

strcpy(tmp[i], ptr);

v\_args[i] = tmp[i];

i++;

}

if(!strcmp("cat", tmp[0]))

{

printf("Processing cat command\n");

}

else if(!strcmp("ls", tmp[0]))

{

printf("Processing ls command\n");

}

else

{

printf("Couldn't discern input: it's not ls and not cat!\n");

}

v\_args[i] = NULL;

pid = fork();

switch (pid)

{

case -1: //if error

{

printf("Fork failed!\n");

exit(-1);

}

case 0: //if child process

{

execvp(v\_args[0], v\_args);

break;

}

default: //if parent process

{

(void)signal(SIGINT, onCtrlC);

int stat;

wait(&stat);

printf("Child process successfully ended with code %d.\n", WEXITSTATUS(stat));

break;

}

}

}

return 0;

}

void onCtrlC(int sig) {

printf("Got signal to kill the child %d\n", sig);

if(kill(pid, SIGINT) == 0)

printf("Child killed!\n");

else

{

printf("Some error has occured while killing the child!\n");

return;

}

}

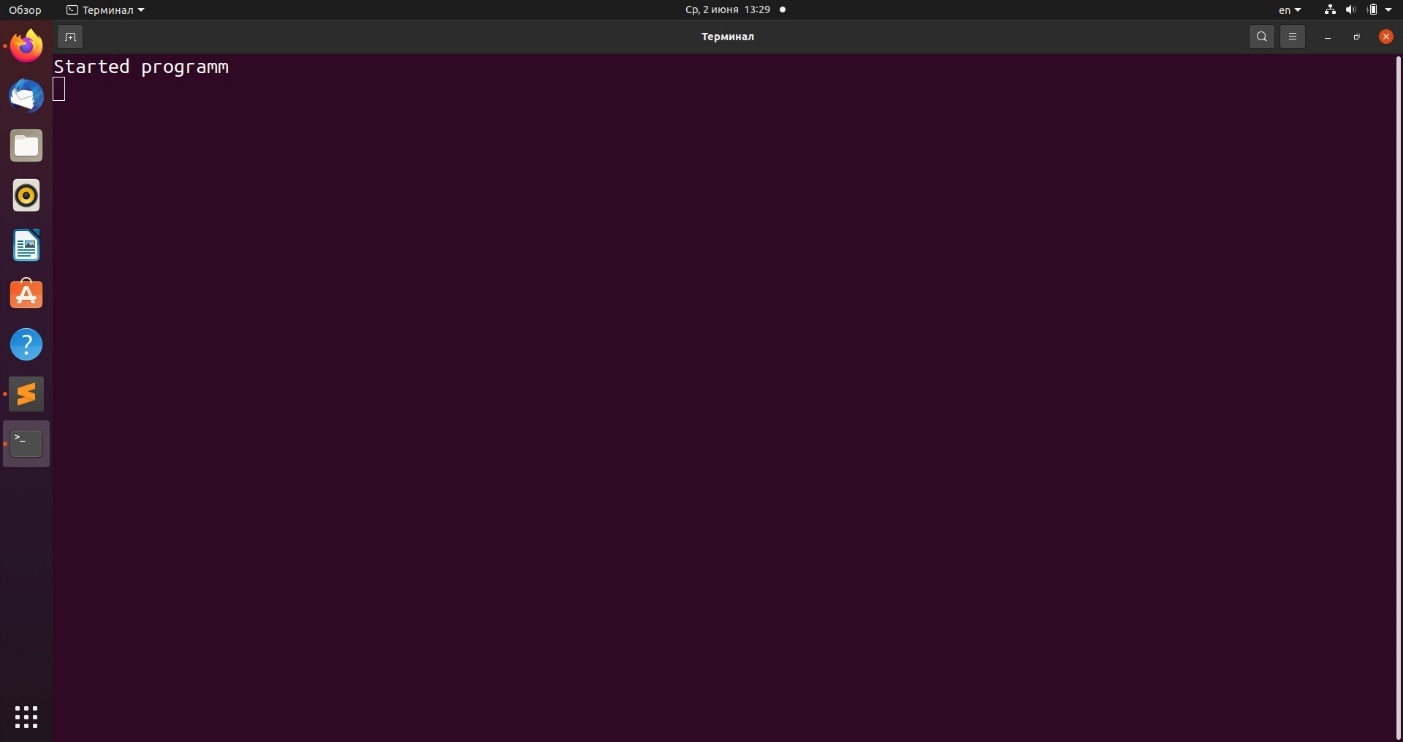
**Использованный makefile**

my\_shell:

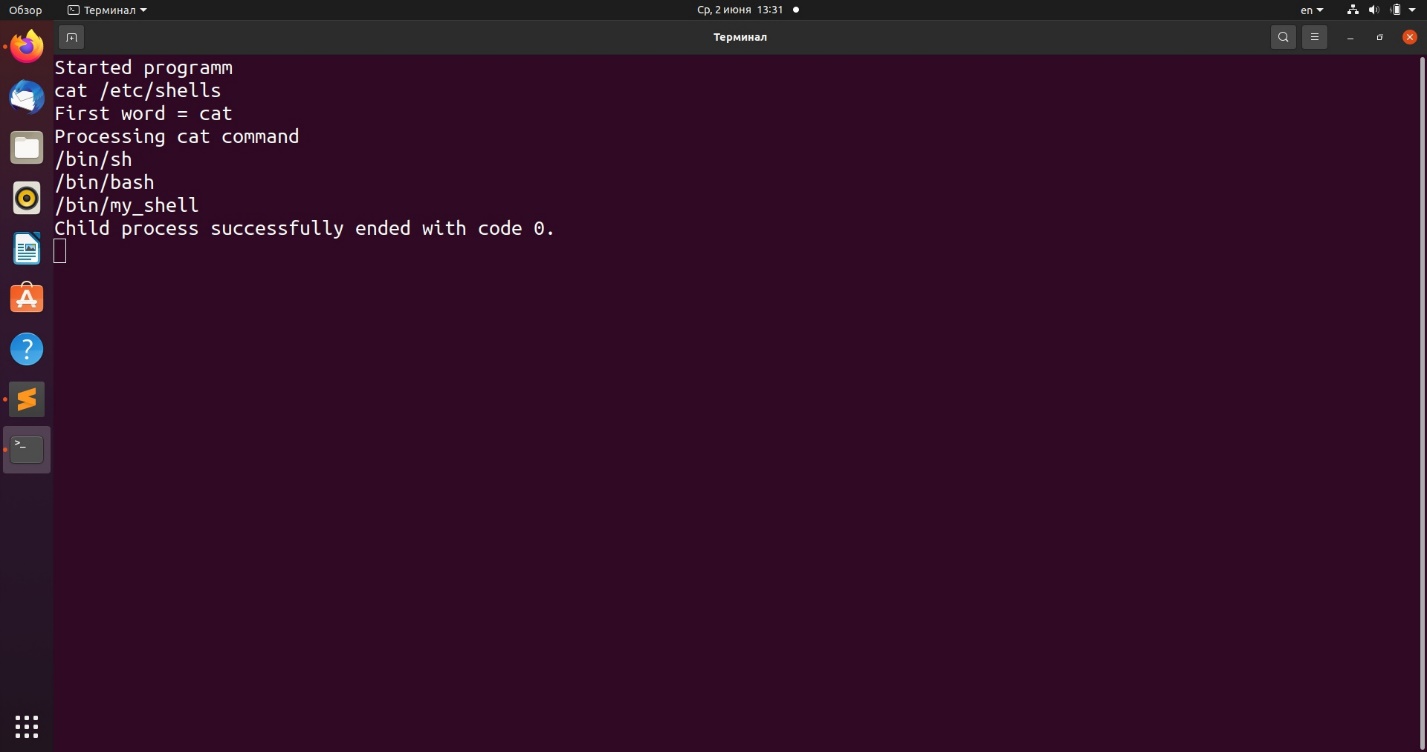
gcc main.c -o my\_shell -Wall -fsanitize=address

**Пример работы программы**

Файл my\_shell скопирован в /bin/, в /etc/shells добавили my\_shell и командой chsh /bin/my\_shell установили my\_shell стандартной оболочкой терминала. Теперь при запуске терминала запускается my\_shell:



Проверка команды cat:



Для проверки обработки Ctrl + C была написана программа, которая с перерывом в секунду бесконечно выводит на экран строку:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

int main() {

while(1) {

printf("Hello World!\n");

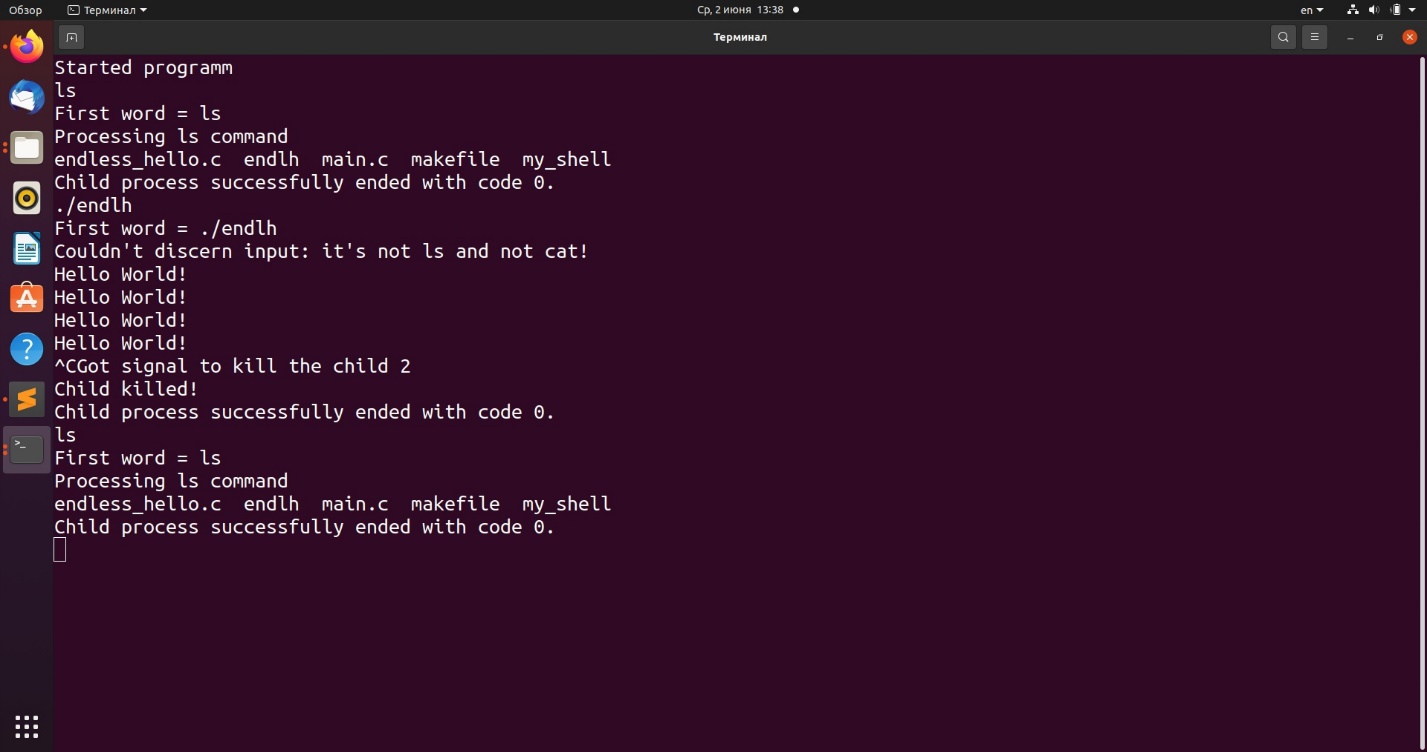
sleep(1);

}

return 0;

}

Пример вызова и остановки этой программы:



**Список использованной литературы**

1. Основы программирования в Linux: Пер. с англ. / Н. Мэтью, Р. Стоунс. – 4-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2009. – 896 с.: ил.