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»

**АППАРАТ ПРЕРЫВАНИЙ. СИГНАЛЫ В ОС UNIX**

Лабораторная работа №10 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:

\_\_\_\_\_\_\_ Григорьев Д. И.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 г.

Великий Новгород

2020

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

**Задание:**

Написать программу для обработки сигналов. Основную часть времени она должна находиться в бесконечном цикле и раз в секунду выводить на экран текущее время.

В программе должны быть реализованы обработчики сигналов SIGUSR1, SIGUSR2, SIGINT. Одновременно с выводом времени в основном цикле так же требуется выводить количество сигналов каждого вида, которые были получены.

До тех пор, пока количество полученных сигналов SIGINT меньше 5, программа должна продолжать работу. Когда будет накоплено больше 5 таких сигналов, требуется вывести на экран сообщение о количестве полученных сигналов каждого вида и прекратить работу.

**Реализация:**

#include <stdio.h>

#include <signal.h>

#include <unistd.h>

#include <time.h>

*int* sig1 = 0, sig2 = 0, sig3 = 0;

*void* handler(*int* *nsig*)

{

    if (nsig == SIGINT)

    {

        sig1++;

    }

    else if (nsig == SIGUSR1)

    {

        sig2++;

    }

    else if(nsig == SIGUSR2)

    {

        sig3++;

    }

}

*int* main()

{

    signal(SIGINT, handler);

    signal(SIGUSR1, handler);

    signal(SIGUSR2, handler);

    while(1)

    {

        if (sig1 < 5)

        {

*time\_t* curTime = time(NULL);

            printf("Current time %s\n", ctime(&curTime));

            printf("SIGINT: [%d]\n", sig1);

            printf("SIGUSR1: [%d]\n", sig2);

            printf("SIGUSR2: [%d]\n", sig3);

        }

       else

       {

            printf("RESULT:\n");

            printf("SIGINT: [%d]\n", sig1);

            printf("SIGUSR1: [%d]\n", sig2);

            printf("SIGUSR2: [%d]\n", sig3);

            break;

       }

        sleep(1);

    }

    return 0;

}

**Результат выполнения программы:**

uselesscloud@uselesscloud\_laptop:/mnt/c/Users/cloud/Desktop/ОС/lab10$ make

gcc signals.c -o signals

./signals

Current time Thu May 14 23:42:19 2020

SIGINT: [0]

SIGUSR1: [0]

SIGUSR2: [0]

^CCurrent time Thu May 14 23:42:19 2020

SIGINT: [1]

SIGUSR1: [0]

SIGUSR2: [0]

Current time Thu May 14 23:42:20 2020

SIGINT: [1]

SIGUSR1: [0]

SIGUSR2: [0]

^CCurrent time Thu May 14 23:42:20 2020

SIGINT: [2]

SIGUSR1: [0]

SIGUSR2: [0]

^CCurrent time Thu May 14 23:42:20 2020

SIGINT: [3]

SIGUSR1: [0]

SIGUSR2: [0]

Current time Thu May 14 23:42:21 2020

SIGINT: [3]

SIGUSR1: [0]

SIGUSR2: [0]

^CCurrent time Thu May 14 23:42:22 2020

SIGINT: [4]

SIGUSR1: [0]

SIGUSR2: [0]

^CRESULT:

SIGINT: [5]

SIGUSR1: [0]

SIGUSR2: [0]

**Вывод:** В процессе выполнения лабораторной работы я научился работать с сигналами в UNIX-like системах.