Министерство образования и науки Российской Федерации

Югорский государственный университет

Институт цифровой экономики

Отчет о лабораторной работе №3

По дисциплине «Разработка системных приложений»

Выполнил: студент группы 1161б

Стародубов Юрий

Проверил: Шицелов А.В.

Ханты-Мансийск

2019

Цели работы:

* Научиться "защищать" свои программы от сигнала прерывания Control-C;
* использовать сигнал таймера для прекращения работы вашей программы по истечению определенного периода времени;
* использовать сигналы в повседневных рутинных операциях.

Задание 1

На основе листинга 3 напишите программу, которая перехватывает все сигналы. В обработчике выводите номер полученного сигнала и его название – при помощи функции strsignal. Перед возвратом из обработчика не забудьте восстановить обработчик. С помощью команды killall (см. man killall) проверьте работу своей программы:

Листинг 1. Код задания 1

void lab31(){

vector<sighandler\_t> asd;

for(int i=0; i<31; ++i)

asd.push\_back(signal(i, alarmHandler));

pause();

for(int i=0; i<asd.size(); ++i)

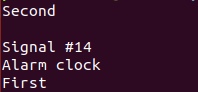
signal(i, asd[i]);

}

void alarmHandler (int s){

cout << endl <<"Signal #"<< s << endl << strsignal(s)<<endl;

}



Скриншот 1 – синхронизация по прерыванию

Задание 2

Модифицируйте ее код так, чтобы вначале дочерний процесс печатал «Second» до того, как родительский процесс напечатает «First». Не используйте вызов sleep() or wait().

Листинг 2. Код задания 2

void lab30(){

pid\_t pid;

pid = fork();

if (pid != 0) {

signal(SIGALRM, alarmHandler);

alarm(1);

pause();

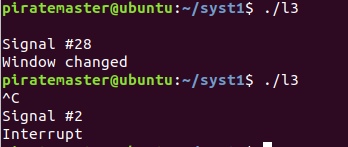
printf("First\n");

}

else {

printf("Second\n");

}



Скриншот 2 – ожидание прерывания

Вывод: В ходе данной лабораторной работы были получены навыки работы с прерываниями