## МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

## «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра Вычислительной техники**

## ОТЧЕТ

**по лабораторной работе №6**

## по дисциплине «Организация процессов и программирование в среде Linux»

**Тема: Организация периодических процессов**

|  |  |
| --- | --- |
| Студент гр. 8308 | Петров Г.А. |
| Преподаватель | Разумовский Г.В. |

Санкт-Петербург 2021

# Цель работы

Целью лабораторной работы является использование сервиса cron, механизма сигналов и интервальных таймеров для организации периодических процессов.

# Задание

Написать периодическую программу, в которой период запуска и количество запусков должны задаваться в качестве ее параметров. При каждом очередном запуске программа должна порождать новый процесс, который выводить на экран свой идентификатор, дату и время старта. Программа и ее дочерний процесс должны быть заблокированы от завершения при нажатии клавиши Ctrl/z. После завершения дочернего процесса программа должна вывести на экран информацию о времени своей работы и дочернего процесса.

# Примеры выполнения программы

Программа была разработана и откомпилирована. После чего программа была запушена на трех вариантах входных данных, а также с попыткой остановки процесса при нажатии клавиши Ctrl/z. Результаты работы программы приведены на рисунках 1-4.

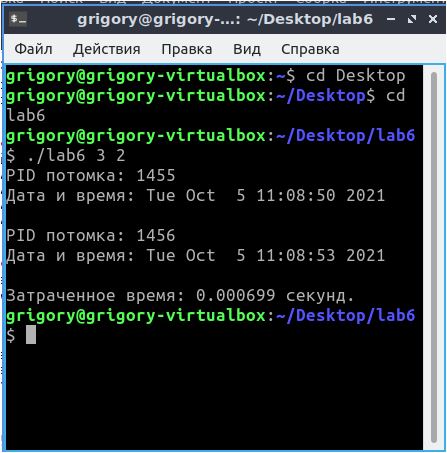


Рисунок 1

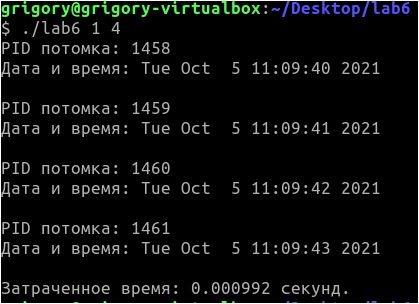


Рисунок 2

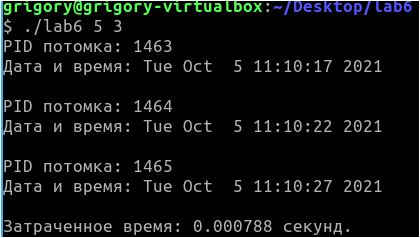


Рисунок 3

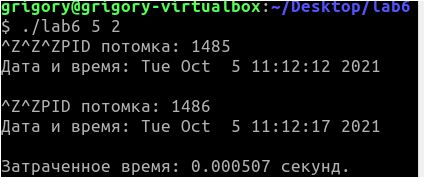


Рисунок 4

# Исходный код программ

## Lab6.cpp

#include <iostream>

#include <ctime>

#include <signal.h>

#include <unistd.h>

#include <sys/wait.h>

#include <sys/time.h>

#include <chrono>

int i=0;

//ф-ия обработки сигнала таймера

void periodicFunction(int sig)

{

pid\_t pid = fork();//создать потомка

if(pid==0)

{//код потомка

time\_t s = time(NULL);

tm\* timeinfo = localtime(&s);

std::cout<<"PID потомка: "<<getpid()<<std::endl;

std::cout<<"Дата и время: "<<asctime(timeinfo)<<std::endl;

exit(0);

}

wait(&pid);

++i;

}

//1 - период | 2 - количество

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

{

unsigned int start\_time = clock();

unsigned int end\_time;

//переопределение обработки сигнала

sigset\_t mask;

sigemptyset(&mask);

sigaddset(&mask,SIGTSTP);//сигнала приостановки

sigprocmask(SIG\_SETMASK,&mask,NULL);

signal(SIGALRM,periodicFunction);//сигнал таймера

int period,count\_period;

period=atoi(argv[1]);

count\_period=atoi(argv[2]);

//инициализация структуры параметров таймера

itimerval value, ovalue;

value.it\_interval.tv\_sec = period;

value.it\_interval.tv\_usec = 0;

value.it\_value.tv\_sec = period;

value.it\_value.tv\_usec = 0;

//работа таймера

setitimer(ITIMER\_REAL,&value,&ovalue);

while(i<count\_period)

pause();

end\_time = clock();

std::cout<<"Затраченное время: "<<(float)(end\_time-start\_time)/CLOCKS\_PER\_SEC<<" секунд."<<std::endl;

return 0;

}

# Вывод

При выполнении лабораторной работы изучены и использован сервис cron, механизм сигналов и интервальных таймеров для организации периодических процессов. Программа разработанная в соответствии с заданием, работает корректно.