

**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 ПЕРИОДЕЧЕСКИХ ПРОЦЕССОВ»**

Студент гр. 8308

\_\_\_\_\_

Тайсумов И.И.

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.

### **Обработка результатов эксперимента.**

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

```

taismov@taismov-TM1703:~/CLionProjects/CPP_lab6$ ./main 4 3
Child's PID: 30172
Child's time: Tue Jan 25 01:41:21 2022

Child's PID: 30173
Child's time: Tue Jan 25 01:41:24 2022

Child's PID: 30174
Child's time: Tue Jan 25 01:41:27 2022

Child's PID: 30182
Child's time: Tue Jan 25 01:41:30 2022

-----
Parent's time: Tue Jan 25 01:41:30 2022

#1 child's time: Tue Jan 25 01:41:21 2022

#2 child's time: Tue Jan 25 01:41:24 2022

#3 child's time: Tue Jan 25 01:41:27 2022

#4 child's time: Tue Jan 25 01:41:30 2022

taismov@taismov-TM1703:~/CLionProjects/CPP_lab6$

```

*Рисунок 1. Работа программы при первом запуске*

```

taismov@taismov-TM1703:~/CLionProjects/CPP_lab6$ ./main 2 1
Child's PID: 30382
Child's time: Tue Jan 25 01:43:04 2022

Child's PID: 30383
Child's time: Tue Jan 25 01:43:05 2022

-----
Parent's time: Tue Jan 25 01:43:05 2022

#1 child's time: Tue Jan 25 01:43:04 2022

#2 child's time: Tue Jan 25 01:43:05 2022

taismov@taismov-TM1703:~/CLionProjects/CPP_lab6$ █

```

*Рисунок 2. Работа программы при запуске с другими входными данными*

Текст программы приведен в приложении.

## **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.

## ПРИЛОЖЕНИЕ

### main.cpp:

```
/**
 * Ислам Тайсумов, группа 8308
 *
 * Компиляция программы:
 * 1. g++ -o main main.cpp
 * 2. ./main x y (x процессов, задержка - y секунд)
 *
 */

#include <iostream>
#include <sys/time.h>
#include <signal.h>
#include <sys/types.h>
#include <unistd.h>
#include <ctime>
#include <vector>

using namespace std;

//сюда записываем время работы каждого процесса
std::vector<string> timeOfChlds;

//функция, вызываемая порождающим процессом
void printTimeChild(int signal_num){

    //ограничиваем от комбинации CTRL+Z
    signal(SIGTSTP, SIG_IGN);

    //получаем время работы дочернего процесса
    time_t secondsChild = time(NULL);
    //переводим его в привычный формат
    tm* timeChild = localtime(&secondsChild);
    //добавляем в глобальный массив
    timeOfChlds.push_back(asctime(timeChild));

    pid_t pid = fork();
    if (!pid){ //если процесс создан, то:
        cout << "Child's PID: " << getpid() << endl;
        cout << "Child's time: " << asctime(timeChild) << endl;
        exit(0);
    }
}

int main(int argc, char *argv[]){
    int numOfChlds = atoi(argv[1]);
    int timeToDelay = atoi(argv[2]);
```

```

//ограничиваем от комбинации CTRL+Z
signal(SIGTSTP,SIG_IGN);
//при окончании таймера запускаем функцию
signal(SIGALRM,printTimeChild);

// struct itimerval {
//     struct timeval it_interval; /* следующее значение */
//     struct timeval it_value;   /* текущее значение */
// };
// struct timeval {
//     long tv_sec;      /* секунды */
//     long tv_usec;    /* микросекунды */
// };

const struct itimerval timer{timeToDelay, 0, timeToDelay, 0};
struct itimerval current_timer;

for (int i = 0; i < numOfChlds; i++){
    if (setitimer(ITIMER_REAL, &timer, &current_timer) == -1){
        //ITIMER_REAL - уменьшается постоянно (в режиме реального времени)
        //и подает сигнал SIGALRM, когда значение таймера становится равным 0.
        cout << "Error!" << endl;
    }
    pause();
}

time_t secondsParent = time(NULL);
tm* timeParent = localtime(&secondsParent);
usleep(500);
cout << "-----" << endl;
cout << "Parent's time: " << asctime(timeParent) << endl;

for(int i = 0; i < numOfChlds; i++) {
    cout << "#" << i+1 << " child's time: " << timeOfChlds[i] << endl;
}

return 0;
}

```