МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра вычислительной техники

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Организация процессов и программирование в среде Linux»

Темы: Создание и идентификация процессов

Студент гр. 8306 _____ Пеунов В.В.

Преподаватель _____ Разумовский Г.В.

Санкт-Петербург,

2021

Цель работы

Изучение и использование системных функций обеспечивающих порождение и идентификацию процессов.

Задание на лабораторную работу

1. Разработать программу, которая порождает 2 потомка. Первый потомок порождается с помощью fork, второй – с помощью vfork с последующей заменой на другую программу. Все 3 процесса должны вывести в один файл свои атрибуты с предварительным указанием имени процесса (например: Предок, Потомок1, Потомок2). Имя выходного файла задается при запуске программы. Порядок вывода атрибутов в файл должен определяться задержками процессов, которые задаются в качестве параметров программы и выводятся в начало файла.

2. Откомпилировать программу и запустить ее 3 раза с различными сочетаниями задержек

Описание работы

После написания программы было проведено 3 эксперимента. С разными значениями задержек.

	Главный процесс	Первый ребенок	Второй ребенок
Эксперимент 1	1	3	5
Эксперимент 2	1	5	3
Эксперимент 3	10	5	3

Эксперимент 1

Родитель - идентификатор процесса: 129707

Родитель - идентификатор предка: 129706

Родитель - идентификатор сессии процесса: 3708

Родитель - идентификатор группы процессов: 129705

Родитель - реальный идентификатор пользователя: 1000

Родитель - эффективный идентификатор пользователя: 1000

Родитель - реальный групповой идентификатор: 1000

Родитель - эффективный групповой идентификатор: 1000

Потомок 1 - идентификатор процесса: 129705

Потомок 1 - идентификатор предка: 114571

Потомок 1 - идентификатор сессии процесса: 3708

Потомок 1 - идентификатор группы процессов: 129705

Потомок 1 - реальный идентификатор пользователя: 1000

Потомок 1 - эффективный идентификатор пользователя: 1000

Потомок 1 - реальный групповой идентификатор: 1000

Потомок 1 - эффективный групповой идентификатор: 1000

Потомок 2 - идентификатор процесса: 129706

Потомок 2 - идентификатор предка: 1

Потомок 2 - идентификатор сессии процесса: 3708

Потомок 2 - идентификатор группы процессов: 129705

Потомок 2 - реальный идентификатор пользователя: 1000

Потомок 2 - эффективный идентификатор пользователя: 1000

Потомок 2 - реальный групповой идентификатор: 1000

Потомок 2 - эффективный групповой идентификатор: 1000

Процесс из другого файла - идентификатор процесса: 129706

Процесс из другого файла - идентификатор предка: 1

Процесс из другого файла - идентификатор сессии процесса: 3708

Процесс из другого файла - идентификатор группы процессов: 129705

Процесс из другого файла - реальный идентификатор пользователя: 1000

Процесс из другого файла - эффективный идентификатор пользователя: 1000

Процесс из другого файла - реальный групповой идентификатор: 1000

Процесс из другого файла - эффективный групповой идентификатор: 1000

Эксперимент 2

Родитель - идентификатор процесса: 128817

Родитель - идентификатор предка: 128816

Родитель - идентификатор сессии процесса: 3708

Родитель - идентификатор группы процессов: 128815

Родитель - реальный идентификатор пользователя: 1000

Родитель - эффективный идентификатор пользователя: 1000

Родитель - реальный групповой идентификатор: 1000

Родитель - эффективный групповой идентификатор: 1000

Потомок 2 - идентификатор процесса: 128816

Потомок 2 - идентификатор предка: 128815

Потомок 2 - идентификатор сессии процесса: 3708

Потомок 2 - идентификатор группы процессов: 128815

Потомок 2 - реальный идентификатор пользователя: 1000

Потомок 2 - эффективный идентификатор пользователя: 1000

Потомок 2 - реальный групповой идентификатор: 1000

Потомок 2 - эффективный групповой идентификатор: 1000

Процесс из другого файла - идентификатор процесса: 128816

Процесс из другого файла - идентификатор предка: 128815

Процесс из другого файла - идентификатор сессии процесса: 3708

Процесс из другого файла - идентификатор группы процессов: 128815

Процесс из другого файла - реальный идентификатор пользователя: 1000

Процесс из другого файла - эффективный идентификатор пользователя: 1000

Процесс из другого файла - реальный групповой идентификатор: 1000

Процесс из другого файла - эффективный групповой идентификатор: 1000

Потомок 1 - идентификатор процесса: 128815

Потомок 1 - идентификатор предка: 114571

Потомок 1 - идентификатор сессии процесса: 3708

Потомок 1 - идентификатор группы процессов: 128815

Потомок 1 - реальный идентификатор пользователя: 1000

Потомок 1 - эффективный идентификатор пользователя: 1000

Потомок 1 - реальный групповой идентификатор: 1000

Потомок 1 - эффективный групповой идентификатор: 1000

Эксперимент 3

Потомок 1 - идентификатор процесса: 127485

Потомок 1 - идентификатор предка: 114571

Потомок 1 - идентификатор сессии процесса: 3708

Потомок 1 - идентификатор группы процессов: 127485

Потомок 1 - реальный идентификатор пользователя: 1000

Потомок 1 - эффективный идентификатор пользователя: 1000

Потомок 1 - реальный групповой идентификатор: 1000

Потомок 1 - эффективный групповой идентификатор: 1000

Родитель - идентификатор процесса: 127489

Родитель - идентификатор предка: 127488

Родитель - идентификатор сессии процесса: 3708

Родитель - идентификатор группы процессов: 127485

Родитель - реальный идентификатор пользователя: 1000

Родитель - эффективный идентификатор пользователя: 1000

Родитель - реальный групповой идентификатор: 1000

Родитель - эффективный групповой идентификатор: 1000

Потомок 2 - идентификатор процесса: 127488

Потомок 2 - идентификатор предка: 1

Потомок 2 - идентификатор сессии процесса: 3708

Потомок 2 - идентификатор группы процессов: 127485

Потомок 2 - реальный идентификатор пользователя: 1000

Потомок 2 - эффективный идентификатор пользователя: 1000

Потомок 2 - реальный групповой идентификатор: 1000

Потомок 2 - эффективный групповой идентификатор: 1000

Процесс из другого файла - идентификатор процесса: 127488

Процесс из другого файла - идентификатор предка: 1

Процесс из другого файла - идентификатор сессии процесса: 3708

Процесс из другого файла - идентификатор группы процессов: 127485

Процесс из другого файла - реальный идентификатор пользователя: 1000

Процесс из другого файла - эффективный идентификатор пользователя: 1000

Процесс из другого файла - реальный групповой идентификатор: 1000

Процесс из другого файла - эффективный групповой идентификатор: 1000

Вывод: в ходе лабораторной работы была проделана работа по изучению функций fork и vfork и сопутствующих им для порождения процессов.

Приложение А. Основная программа

```
#include <iostream>

#include <fstream>

#include <unistd.h>

#include <sys/types.h>

#include <string>

#include <sys/wait.h>


using namespace std;


void output_in_file(const string& process_name){

    string file_path = "/home/peunov/highschool/opp-linux/output/output.txt";

    ofstream file;

    file.open(file_path, ios::app);


    if (file.is_open()) {

        pid_t process_id = getpid();

        file << process_name << " - идентификатор процесса: " <<  process_id << endl;

        file << process_name << " - идентификатор предка: " << getppid() << endl;

        file << process_name << " - идентификатор сессии процесса: " << getsid(process_id) << endl;

        file << process_name << " - идентификатор группы процессов: " << getpgid(process_id) << endl;

        file << process_name << " - реальный идентификатор пользователя: " << getuid() << endl;
```

```
        file << process_name << " - эффективный идентификатор пользователя: " << geteuid() << endl;
        file << process_name << " - реальный групповой идентификатор: " << getgid() << endl;
        file << process_name << " - эффективный групповой идентификатор: " << getegid() << endl;
    }

    file.close();
}

void main_process(){
    output_in_file("Родитель");
}

void child1_process(){
    output_in_file("Потомок 1");
}

void child2_process(){
        const char *child2_program_path = "/home/peunov/highschool/opp-linux/laba1.2/cmake-build-debug/laba1_2";
    output_in_file("Потомок 2");
    execlp(child2_program_path, NULL);
}

int error(){
```

```
    cout << "При создании процесса произошла ошибка";
    return 1;
}

int spawning_processes(){
    pid_t fork_process_id = fork();

    if(fork_process_id == -1){
        return error();
    }

    if(fork_process_id > 0){
        sleep(5);
        child1_process();
    }

    if(fork_process_id == 0){
        pid_t vfork_process_id = vfork();

        if(vfork_process_id == -1){
            return error();
        }

        if(vfork_process_id > 0){
            sleep(3);
```

```
            child2_process();
        }

        if(vfork_process_id == 0){
            sleep(10);
            int status;
            wait(&status);
            main_process();
        }
    }

    return 0;
}

int main(){
    cout << "Лабораторная работа №3" << endl;
    return spawning_processes();
}
```

Приложение В. Заменяемая программа

```
#include <iostream>

#include <fstream>

#include <unistd.h>

#include <string>

using namespace std;


void output_in_file(const string& process_name){

    string file_path = "/home/peunov/highschool/opp-linux/output/output.txt";

    ofstream file;

    file.open(file_path, ios::app);


    if (file.is_open()) {

        pid_t process_id = getpid();

        file << process_name << " - идентификатор процесса: " <<  process_id << endl;

        file << process_name << " - идентификатор предка: " << getppid() << endl;

        file << process_name << " - идентификатор сессии процесса: " << getsid(process_id) << endl;

        file << process_name << " - идентификатор группы процессов: " << getpgid(process_id) << endl;

        file << process_name << " - реальный идентификатор пользователя: " << getuid() << endl;

        file << process_name << " - эффективный идентификатор пользователя: " << geteuid() << endl;

        file << process_name << " - реальный групповой идентификатор: " << getgid() << endl;
```

```
        file << process_name << " - эффективный групповой идентификатор: " << getegid() <<
endl;
    }

    file.close();
    exit(EXIT_SUCCESS);
}

int main(){
    output_in_file("Процесс из другого файла");
    return 0;
}
```