**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра вычислительной техники**

отчет

**по лабораторной работе №3**

**по дисциплине «Программирование в среде Linux»**

Тема: «Создание и идентификация процессов»

|  |  |  |
| --- | --- | --- |
| Студент гр. 6307 |  | Лазарев С. О. |
| Преподаватель |  | Разумовский Г. В. |

Санкт-Петербург

2019

# ЦЕЛЬ

Изучение и использование системных функций, обеспечивающих порождение и идентификацию процессов.

# ЗАДАНИЕ

Разработать программу, которая порождает 2 потомка: fork() и vfork(), с последующей заменой на другую программу. Все 3 процесса должны вывести в один файл свои атрибуты с указанием имени процесса. Порядок вывода должен определяться задержками процессов, которые задаются в качестве параметров программы и выводятся в начало файла. Имя файла задается в начале программы.

# РЕЗУЛЬТАТ

# 

# КОД

main.cpp

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include "utils.cpp"

using namespace std;

int main(){

string filename = inputString("File name");

int fork\_sleep = inputInt("Fork sleep");

int vfork\_sleep = inputInt("Vfork sleep");

int exec\_sleep = inputInt("Exec sleep");

char \_param\_exec\_sleep[64];

char \_param\_file\_name[64];

sprintf(\_param\_exec\_sleep,"%d",exec\_sleep);

sprintf(\_param\_file\_name,"%s", filename.c\_str());

char \*argv[] = {

"/home/sergey/Рабочий стол/linux\_labs/lab3/exec",

\_param\_exec\_sleep,

\_param\_file\_name,

0

};

ofstream file(filename);

file << "fork - " << to\_string(fork\_sleep) << " seconds" << endl;

file << "vfork - " << to\_string(vfork\_sleep) << " seconds" << endl;

file << "exec - " << to\_string(exec\_sleep) << " seconds" << endl << endl;

if(fork() == 0){

sleep(fork\_sleep);

writeAttributes("fork",filename);

exit(1);

}

if (vfork() == 0) {

sleep(vfork\_sleep);

writeAttributes("vfork",filename);

execv(argv[0], argv);

perror("exec error");

exit(-1);

}

file.close();

return 0;

}

exec.cpp

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include "utils.cpp"

using namespace std;

int main(int argc, char \*argv[]){

if(argv[1] == nullptr || argv[2] == nullptr){return -1;}

int sleep\_value = atoi(argv[1]);

string filename = argv[2];

sleep(sleep\_value);

writeAttributes("exec",filename);

return 3;

}

utils.cpp

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <iostream>

#include <fstream>

using namespace std;

int inputInt(string message){

int integer=0;

cout << (message + " = ");

cin >> integer;

return integer;

}

string inputString(string message){

string str = "";

cout << (message + " = ");

cin >> str;

cout << endl;

return str;

}

void writeAttributes(const char \*name, string filename){

ofstream cout(filename, ios\_base::app);

cout << endl << name << endl;

pid\_t pid = getpid();

cout << "Идентификатор процесса " << pid << endl;

cout << "Идентификатор предка " << getppid() << endl;

cout << "Идентификатор сессии процесса " << getsid(pid) << endl;

cout << "Идентификатор группы процессов " << getpgid(pid) << endl;

cout << "Реальный дентификатор пользователя " << getuid() << endl;

cout << "Эффективный идентификатор пользователя " << geteuid() << endl;

cout << "Реальный групповой идентификатор " << getgid() << endl;

cout << "Эффективный групповой идентификатор " << getegid() << endl;

cout << endl;

cout.close();

}

# ВЫВОДЫ

Познакомился с функциями создания процессов fork(), vfork(). Научился запускать другие программы, используя функции exec. Получил атрибуты процессов при помощи команд getpid, getppid, getsid и тд.