

Федеральное государственное автономное
образовательное учреждение
высшего профессионального образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Вариант 2

Тема работы

Порождение и замещение процессов

Преподаватель
Шаров В.В.

подпись, дата

Студент гр. КИ–15 08 Б
Войченко В.В.

подпись, дата

Красноярск 2017

Выполнение работы:

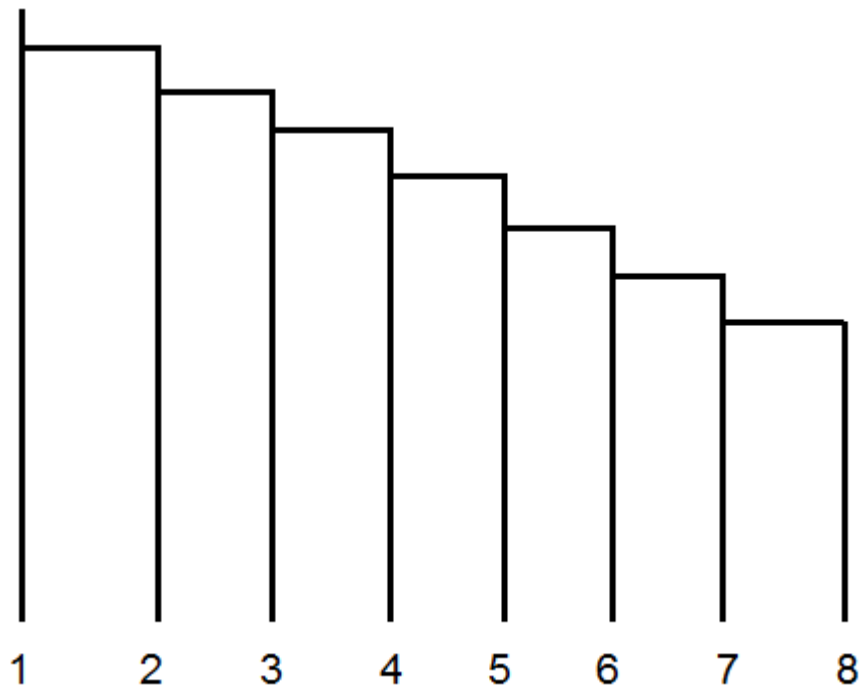
Задание:

1. Написать отчет по приведенной в варианте программе: составить схему процессов, указать программный код, выполняемый каждым процессом, указать действия выполняемые процессами. Сколько процессов создается во время работы программы? Все ли процессы завершаются?
2. Составить программу на языке C – перед пользователем появляется меню:
 - 1 – Выполнить введенную команду
 - 2 – Запустить на исполнение файл со сценарием (вариант в работе 1)
 - 3 – Выполнить действие, указанное в варианте
 - 4 - Выход

Для выполнения пунктов 1-3 использовать порождение дочернего процесса. Для третьего пункта предполагается реализация посредством команд shell, сформированных в строки аргументов функций `execp` или `system`, но если число команд shell велико тогда можно создать командный файл и запустить его с помощью `execp` или `system`.

Теория

Программа	<pre>for (int i=0; i<4; i++) fork();</pre>
-----------	---



Создаётся 8 процессов. Все процессы завершаются.

Выполняемый код

Прц 1	Прц 2	Прц 3	Прц 4	Прц 5	Прц 6	Прц 7	Прц 8
fork();	fork();	fork();	fork();	fork();	fork();	fork();	Не выполн. условие (4 < 4)

Словесное описание

	Прц 1	Прц 2	Прц 3	Прц 4	Прц 5	Прц 6	Прц 7	Прц 8
Порожд. Доч. Прцсс	+	+	+	+	+	+	+	Не выполн. условие (4 < 4)
Вып. Ветвлен.	+	+	+	+	+	+	+	-, конец итератора в счетчике

Содержимое файла script.sh

```
#!/bin/bash
ls -lr -R | grep "rwxr-xr-x"
```

Содержимое файла lab2.cpp

```
#include <sys/types.h>
#include <iostream>
#include <cstdlib>
#include <string>
#include <unistd.h>

using namespace std;

void displayInterface ()
{
    cout << "*****\n";
    cout << "1 - Выполнить введенную команду\n";
    cout << "2 - Запустить на исполнение файл со сценарием\n";
    cout << "3 - Пометить все файлы как исполняемые\n";
    cout << "4 - Выход\n";
    cout << "*****\n";
}

void runCommand ()
{
    string command;
    cout << "Ваша команда:\n";
    cin >> command;
    system (command.c_str());
}

void runShellcript ()
{
    system("chmod +x script.sh");
    cout << "Результат работы скрипта:\n";
    system("./script.sh");
}

void runLab2Command ()
{
    cout << "Все файлы в данной директории помечены как исполняемые\n";
    system ("chmod +x *.*");
}
```

```
int main()
{
    bool exit = false;
    int choice = 0;
    system("clear");
    while(!exit)
    {
        displayInterface();
        cin >> choice;
        switch (choice)
        {
            case 1: {system ("clear"); runCommand(); break;}
            case 2: {system ("clear"); runShellcript(); break;}
            case 3: {system ("clear"); runLab2Command(); break;}
            case 4: {exit = true; break;}
            default: cout << "Введите число от 1 до 4!\n";
        }
    }
    return 0;
}
```