

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

ОТЧЕТ
по лабораторной работе №4
Дисциплина «ОСиСП»

Выполнил:

студ. гр.ПО-3

Аниськин К.Д.

Проверил:

Давидюк.Ю.И.

Брест 2020

Лабораторная работа №4.

Цель работы: изучить GCC. процессы.

Задание для выполнения

Написать программу, которая будет реализовывать следующие функции:

- сразу после запуска получает и сообщает свой ID и ID родительского процесса;
- перед каждым выводом сообщения об ID процесса и родительского процесса эта информация получается заново;
- порождает процессы, формируя генеалогическое дерево согласно варианту, сообщая, что "процесс с ID таким-то породил процесс с таким-то ID";
- перед завершением процесса сообщить, что "процесс с таким-то ID и таким-то ID родителя завершает работу";
- один из процессов должен вместо себя запустить программу, указанную в варианте задания.

На основании выходной информации программы предыдущего пункта изобразить генеалогическое дерево процессов (с указанием идентификаторов процессов). Объяснить каждое выведенное сообщение и их порядок в предыдущем пункте.

Варианты индивидуальных заданий

В столбце `exec` указан номер процесса, выполняющего вызов `exec`, команды для которого указаны в последнем столбце. Запускайте команду обязательно с какими-либо параметрами.

№	fork	exec	
1	0 1 1 1 3 3 5	1	ls

```
#include <stdio.h>
```

```
#include <sys/types.h>
```

```
#include <unistd.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
int id1 = fork();
```

```
if (id1 == 0) {
```

```
printf("proc1 pid(%d), ppid (%d) \n", getpid(), getppid());
```

```
int id2 = fork();
```

```
if (id2 == 0) {
```

```
printf("proc2 pid(%d), ppid (%d) \n", getpid(), getppid());
```

```
exit(0);
```

```
}
```

```
int id3 = fork();
```

```
if (id3 == 0) {
```

```
printf("proc3 pid(%d), ppid (%d) \n", getpid(), getppid());

int id5 = fork();

if (id5 == 0) {

printf("proc5 pid(%d), ppid (%d) \n", getpid(), getppid());

int id7 = fork();

if (id7 == 0) {

printf("proc7 pid(%d), ppid(%d)\n", getpid(), getppid());

exit(0);

}

exit(0);

}

int id6 = fork();

if (id6 == 0) {

printf(" proc6 pid(%d), ppid (%d) \n", getpid(), getppid());

execl("/bin/ls", "ls", "--version", NULL);

exit(0);

}

exit(0);

}

int id4 = fork();

if (id4 == 0) {

printf("proc4 pid(%d), ppid (%d) \n", getpid(), getppid());

exit(0);

}

}

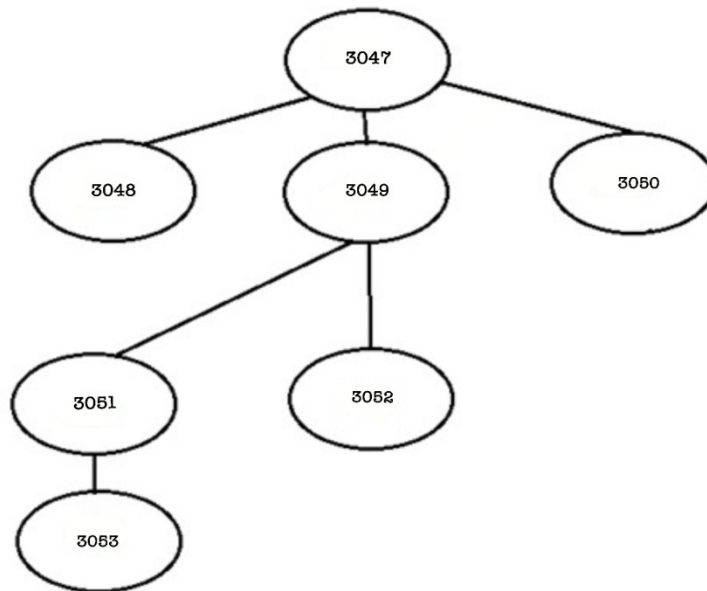
return 0;

}
```

Результат:

```
kirill@kirill-VirtualBox:~$ gcc kirill.c
kirill@kirill-VirtualBox:~$ ./a.out
proc1 pid(3047), ppid (840)
kirill@kirill-VirtualBox:~$ proc2 pid(3048), ppid (3047)
proc4 pid(3050), ppid (840)
proc3 pid(3049), ppid (840)
  proc6 pid(3052), ppid (840)
proc5 pid(3051), ppid (840)
proc7 pid(3053), ppid(840)
ls (GNU coreutils) 8.28
Copyright (C) 2017 Free Software Foundation, Inc.
Лицензия GPLv3+: GNU GPL версии 3 или новее <http://gnu.org/licenses/gpl.html>
Это свободное ПО: вы можете изменять и распространять его.
Нет НИКАКИХ ГАРАНТИЙ до степени, разрешённой законом.

Авторы программы -- Richard M. Stallman и David MacKenzie.
```



Вывод: изучил GCC. процессы.