

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

Лабораторная работа №4
По дисциплине: «Операционные системы и системное программирование»
Тема: «Процесс»

Подготовил:
Студент 2 курса
Группы ПО-3
Новикович А.А.
Проверил:
Давидюк Ю.И.

Брест 2020

Лабораторная работа №4

Цель: изучить работу с процессами и компилятором GCC

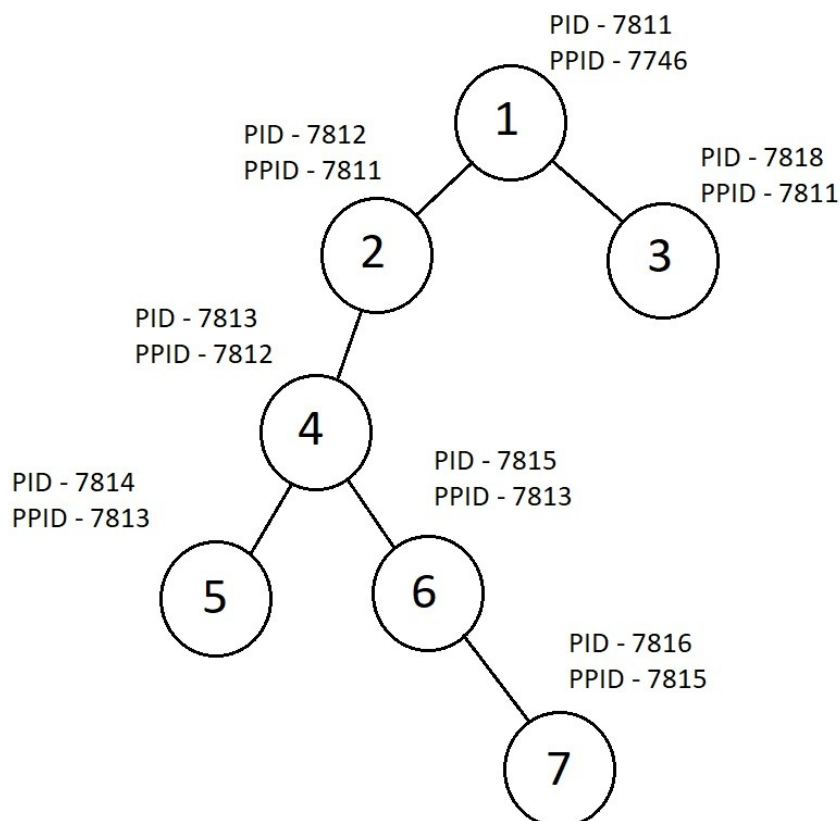
Написать программу, которая будет реализовывать следующие функции:

- 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.

Вариант 18.

18	0 1 1 2 4 4 6	4	pwd
----	---------------	---	-----



Код программы:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
int main(){
    printf("Процесс 1:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());
    pid_t pid;
    if((pid = fork()) == -1){
        printf("Ошибка! \n");
    }
    else if(pid == 0){
        printf("Порождение процесса 2: \n\tPID - %d, \n\tPPID - %d\n", getpid(),
getppid());
        if((pid = fork()) == -1){
            printf("Ошибка!\n");
        }
        else if(pid == 0){
            printf("Порождение процесса 4:\n\tPID = %d,\n\tPPID = %d\n", getpid(),
getppid());
                if((pid = fork()) == -1){
                    printf("Ошибка\n");
                }
            else if(pid == 0){
                printf("Порождение процесса 5:\n\tPID - %d,\n\tPPID - %d\n", getpid(),
getppid());
                    printf("Завершение процесса 5.\n");
                    exit(0);
            }
                else sleep(2);
                if((pid = fork()) == -1){
                    printf("Ошибка\n");
                }
            else if(pid == 0){
                printf("Порождение процесса 6:\n\tPID - %d,\n\tPPID - %d\n", getpid(),
getppid());
                    if((pid = fork()) == -1){
                        printf("Ошибка\n");
                    }
                else if(pid == 0){
                    printf("Порождение процесса 7:\n\tPID - %d,\n\tPPID - %d\n",
getpid(), getppid());
                        printf("Завершение процесса 7.\n");
                        exit(0);
                    }
                else sleep(1);
                printf("Завершение процесса 6.\n");
                exit(0);
            }
                else sleep(4);
                printf("Завершение процесса 4.\n");
            }
    }
```

```

        execl("/bin/pwd", "pwd", NULL);
        exit(0);
    }
    else sleep(8);
    printf("Завершение процесса 2.\n");
    exit(0);
}
else sleep(10);
if((pid = fork()) == -1){
printf("Ошибка! \n");
}
else if(pid == 0){
    printf("Порождение процесса 3:\n\tPID - %d,\n\tPPID - %d\n", getpid(), getppid());
    printf("Завершение процесса 3.\n");
    exit(0);
}
else sleep(10);
printf("Завершение процесса 1.\n");
exit(0);
return 0;
}

```

PPID последующего процесса совпадает с родительским PID.

```

artyom@AN:~/Загрузки$ gcc laba4.c
artyom@AN:~/Загрузки$ ./a.out
Процесс 1:
    PID - 7811,
    PPID - 7746
Порождение процесса 2:
    PID - 7812,
    PPID - 7811
Порождение процесса 4:
    PID = 7813,
    PPID = 7812
Порождение процесса 5:
    PID - 7814,
    PPID - 7813
Завершение процесса 5.
Порождение процесса 6:
    PID - 7815,
    PPID - 7813
Порождение процесса 7:
    PID - 7816,
    PPID - 7815
Завершение процесса 7.
Завершение процесса 6.
Завершение процесса 4.
/home/artyom/Загрузки
Завершение процесса 2.
Порождение процесса 3:
    PID - 7818,
    PPID - 7811
Завершение процесса 3.
Завершение процесса 1.

```