**Пометка к заданию №10**

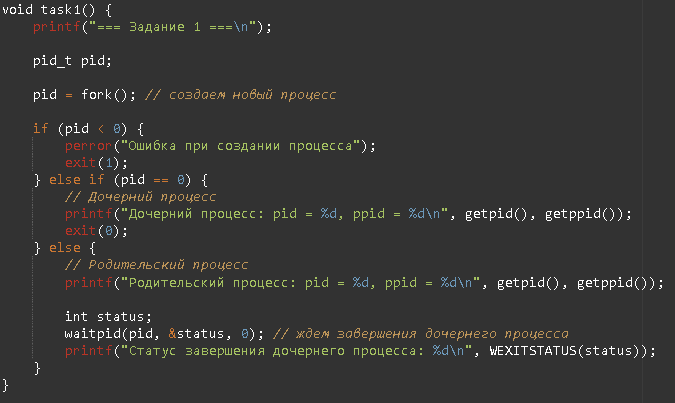
***Задание 1:***

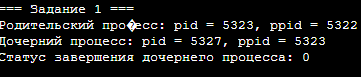
Создается один дочерний процесс с помощью fork().

Если fork() завершился неудачно, выводится сообщение об ошибке и программа завершает работу.

Если это дочерний процесс, выводится его pid (идентификатор процесса) и ppid (идентификатор родительского процесса), затем процесс завершается.

Если это родительский процесс, выводятся его pid и ppid, после чего родитель ждет завершения дочернего процесса с помощью waitpid(). После завершения дочернего процесса, выводится его статус завершения.



Пример выполнения задания 1:  


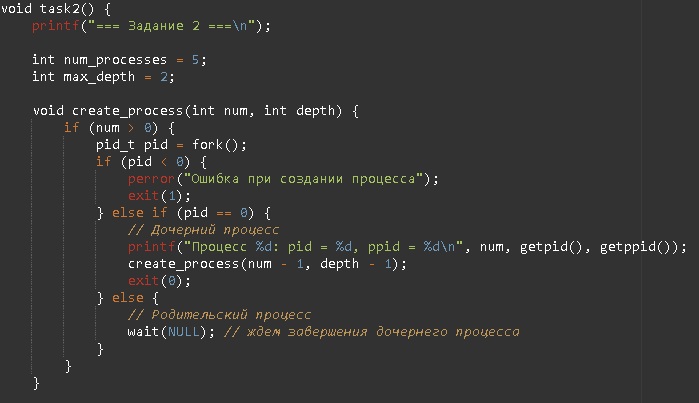
***Задание 2****:*

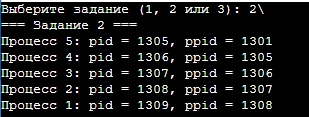
Создается цепочка из 5 процессов, где каждый процесс (кроме последнего) создает еще один дочерний процесс.

Процессы создаются рекурсивно с помощью функции create\_process, которая вызывает сама себя до достижения заданной глубины.

Каждый процесс выводит свой pid, ppid и номер процесса.

Родительский процесс ожидает завершения дочернего процесса перед завершением своей работы.



Пример выполнения задания 2:  


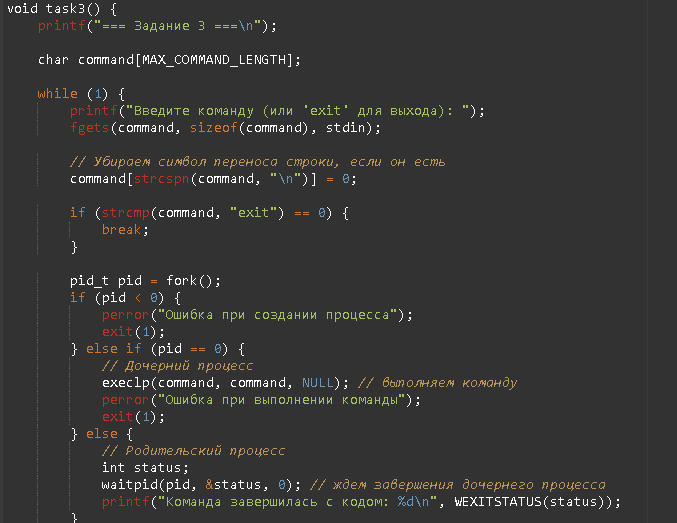
***Задание 3****:*

Программа ждет ввода команды от пользователя в цикле.

Если пользователь вводит exit, программа завершает работу.

Иначе создается новый дочерний процесс, который пытается выполнить введенную команду с помощью execlp().

Родительский процесс ожидает завершения дочернего и выводит код завершения команды.



Пример выполнения задания 3, используя команду ls:

