

## Practical - 1

### Fork () system call

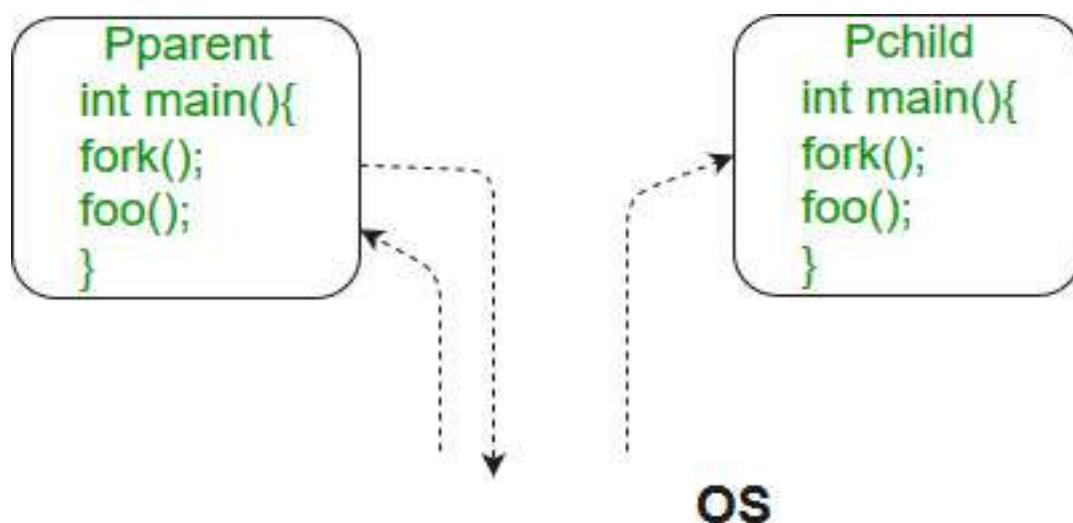
Fork system call is used for creating a new process, which is called child process, which runs concurrently with the process that makes the fork() call (parent process). After a new child process is created, both processes will execute the next instruction following the fork() system call. A child process uses the same pc(program counter), same CPU registers, same open files which use in the parent process.

It takes no parameters and returns an integer value. Below are different values returned by fork().

Negative Value: creation of a child process was unsuccessful.

Zero: Returned to the newly created child process.

Positive value: Returned to parent or caller. The value contains process ID of newly created child process.



**Write a program for creating Child process using fork() system call. Print the process ID of child and parent process. :-**

**Source Code: -**

```
#include<stdio.h>

#include<unistd.h>

int main()

{

int p;

p=fork();

if(p<0)

{

printf("cannot be executed");

}

else if(p==0)

{

printf("child process id %d \n",getpid());

}

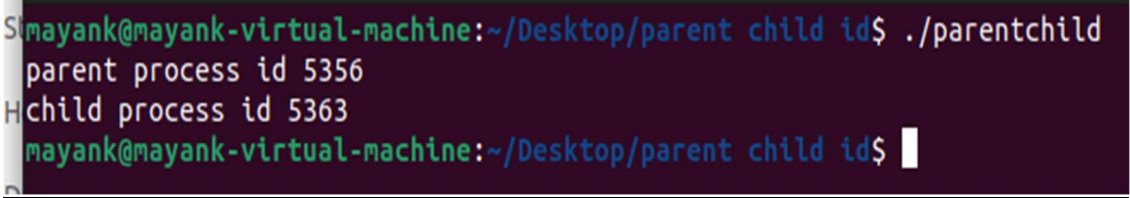
else{

printf("parent process id %d \n",getppid());

}
```

```
return 0;  
}
```

## Output: -



```
mayank@mayank-virtual-machine:~/Desktop/parent child id$ ./parentchild  
parent process id 5356  
child process id 5363  
mayank@mayank-virtual-machine:~/Desktop/parent child id$
```