

# Lab-9

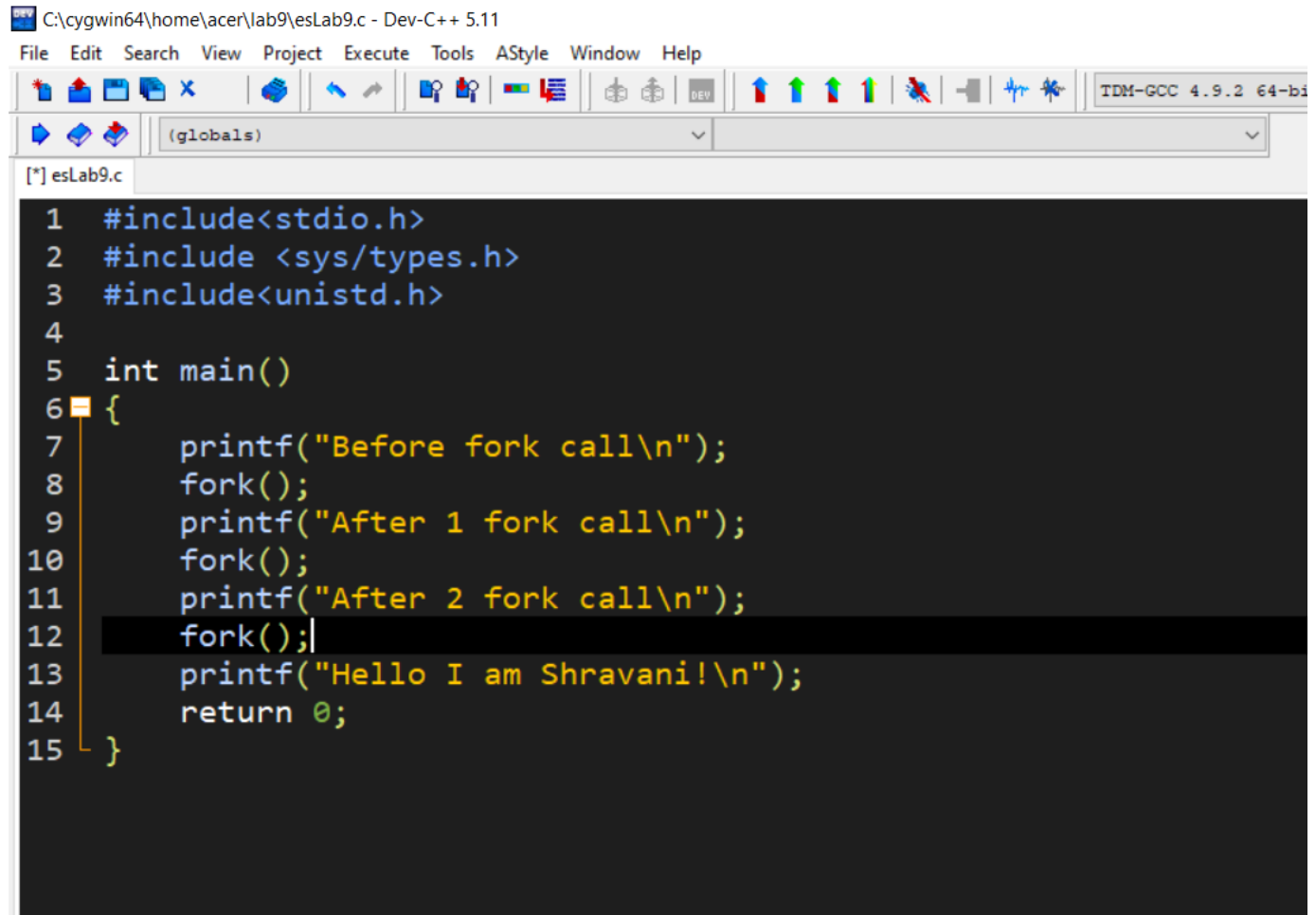
Friday, October 16, 2020 12:00 PM

**Aim:** Program to introduce process and child process: introduction to fork system call, printing parent and child process IDs.

**Date:** 16-10-20

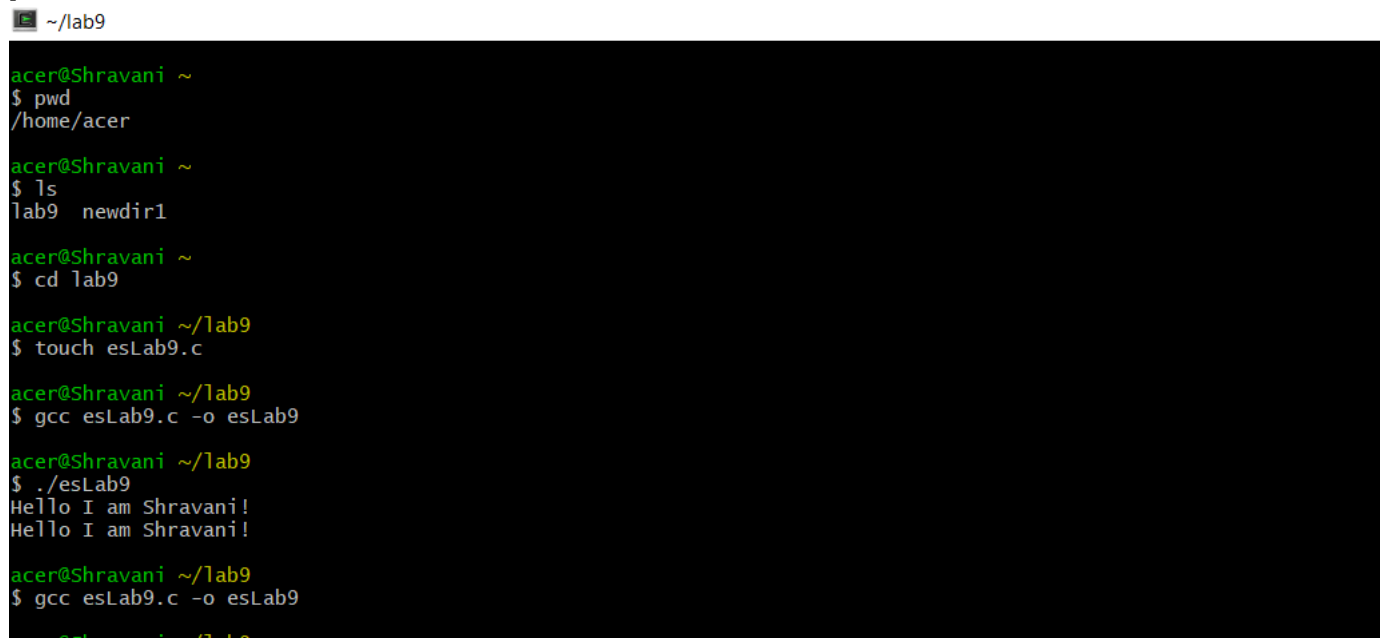
**Source code:**

Understanding Child and parent using fork



```
1 #include<stdio.h>
2 #include <sys/types.h>
3 #include<unistd.h>
4
5 int main()
6 {
7     printf("Before fork call\n");
8     fork();
9     printf("After 1 fork call\n");
10    fork();
11    printf("After 2 fork call\n");
12    fork();
13    printf("Hello I am Shravani!\n");
14    return 0;
15 }
```

There are 6 printf statements printing because there are 3 forks under parent process



```
acer@Shravani ~
$ pwd
/home/acer

acer@Shravani ~
$ ls
lab9  newdir1

acer@Shravani ~
$ cd lab9

acer@Shravani ~/lab9
$ touch esLab9.c

acer@Shravani ~/lab9
$ gcc esLab9.c -o esLab9

acer@Shravani ~/lab9
$ ./esLab9
Hello I am Shravani!
Hello I am Shravani!

acer@Shravani ~/lab9
$ gcc esLab9.c -o esLab9

acer@Shravani ~/lab9
```

```

acer@Shravani ~/lab9
$ ./esLab9
Before fork call
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!

acer@Shravani ~/lab9
$ gcc esLab9.c -o esLab9

acer@Shravani ~/lab9
$ ./esLab9
Before fork call
After 1 fork call
After 1 fork call
After 2 fork call
After 2 fork call
After 2 fork call
After 2 fork call
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!
Hello I am Shravani!

```

Printing PPID and PID

```

C:\cygwin64\home\acer\lab9\esLab9.c - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
TDM-GCC 4.9.2 64-bit

[*] esLab9.c
1 #include<stdio.h>
2 #include <sys/types.h>
3 #include<unistd.h>
4
5 int main()
6 {
7     printf("Before fork call\n");
8     printf("PID= %d PPID= %d\n", getpid(),getppid());
9     return 0;
10 }

```

Always a different PPID and PID gets printed due to the system process

```

acer@Shravani ~/lab9
$ gcc esLab9.c -o esLab9

acer@Shravani ~/lab9
$ ./esLab9
Before fork call
PID =1879 PPID =1834

```

Checking with If else condition for Parent and child process:

```

esLab9.c

```

```

1  #include<stdio.h>
2  #include <sys/types.h>
3  #include<unistd.h>
4
5  int main()
6  {
7      if (fork()==0){
8          printf("Hello from child!\n");
9      }
10     else{
11         printf("Hello from Parent!\n");
12     }
13     return 0;
14 }

```

Compiled output:

```

acer@Shravani ~/lab9
$ gcc esLab9.c -o esLab9

acer@Shravani ~/lab9
$ ./esLab9
Hello from child!
Hello from Parent!

acer@Shravani ~/lab9
$ |

```

Printing PPID and PID for parent and child

```

1  #include<stdio.h>
2  #include <sys/types.h>
3  #include<unistd.h>
4
5  int main()
6  {
7      if (fork()==0){
8          printf("Hello from child!\n PID=%d PPID=%d\n", getpid(),
9      }
10     else{
11         printf("Hello from Parent!\n PID=%d PPID=%d", getpid(),
12     }
13     return 0;
14 }

```

The output is different for PPID and PID of the Parent and Child

```

acer@Shravani ~/lab9
$ gcc esLab9.c -o esLab9

acer@Shravani ~/lab9
$ ./esLab9
Hello from Parent!
PID=1892 PPID=1834Hello from child!
PID=1893 PPID=1892

acer@Shravani ~/lab9

```

```
acer@sin-avant: ~/Tab9  
$ |
```

Inference:

In the above performed experiment we observed the function of parent and child using fork system call, we also printed the Parent IDs and Child Process IDs of them using `getpid()` and `getppid()` functions and obtained desired output.