PRACTICAL : 4

Aim :

[A] Write a C program that will print parent process id and child process id. Mention error checking if the child process is not created.

Code :

pr4.c

```c
#include<stdio.h>
#include<unistd.h>
int main()
{

        pid_t pid,ppid;
        pid=getpid();//To get child process id
        ppid=getppid();//to get parent process id
        printf("Child pid : %d\n",pid);
        printf("Parent pid : %d\n",ppid);

        return 0;
}
```

```
jay@jay-VirtualBox:~$ gcc pr4.c -o pr4
jay@jay-VirtualBox:~$ ./pr4
Child pid : 2029
Parent pid : 1880
jay@jay-VirtualBox:~$
```

TASK (With Assignment):

pr4task.c

```c
#include<stdio.h>
#include<string.h>
#include<sys/types.h>
#define MAX_COUNT 200
#define BUF 100

void main(void)
{
        pid_t pid;
        int i;
        char buf[BUF];
        fork();
        pid=getpid();
        for(i=0;i<=MAX_COUNT;i++){
                sprintf(buf,"from pid : %d , value : %d\n",pid,i);
                write(1,buf,strlen(buf));
        }
}
```

```
jay@jay-VirtualBox:~$ gcc pr4task.c -o pr4task
jay@jay-VirtualBox:~$ ./pr4task
from pid : 2086 , value : 0
from pid : 2086 , value : 1
from pid : 2086 , value : 2
from pid : 2086 , value : 3
from pid : 2086 , value : 4
from pid : 2086 , value : 5
from pid : 2086 , value : 6
from pid : 2086 , value : 7
from pid : 2086 , value : 8
from pid : 2086 , value : 9
from pid : 2086 , value : 10
from pid : 2086 , value : 11
from pid : 2086 , value : 12
from pid : 2086 , value : 13
from pid : 2086 , value : 14
from pid : 2086 , value : 15
from pid : 2086 , value : 16
from pid : 2086 , value : 17
from pid : 2086 , value : 18
from pid : 2086 , value : 19
from pid : 2086 , value : 20
from pid : 2086 , value : 21
from pid : 2086 , value : 22
from pid : 2086 , value : 23
from pid : 2086 , value : 24
from pid : 2086 , value : 25
from pid : 2086 , value : 26
from pid : 2086 , value : 27
from pid : 2086 , value : 28
```

```
from pid : 2086 , value : 172
from pid : 2086 , value : 173
from pid : 2086 , value : 174
from pid : 2086 , value : 175
from pid : 2086 , value : 176
from pid : 2086 , value : 177
from pid : 2086 , value : 178
from pid : 2086 , value : 179
from pid : 2086 , value : 180
from pid : 2086 , value : 181
from pid : 2086 , value : 182
from pid : 2086 , value : 183
from pid : 2086 , value : 184
from pid : 2086 , value : 185
from pid : 2086 , value : 186
from pid : 2086 , value : 187
from pid : 2086 , value : 188
from pid : 2086 , value : 189
from pid : 2086 , value : 190
from pid : 2086 , value : 191
from pid : 2086 , value : 192
from pid : 2086 , value : 193
from pid : 2086 , value : 194
from pid : 2086 , value : 195
from pid : 2086 , value : 196
from pid : 2086 , value : 197
from pid : 2086 , value : 198
from pid : 2086 , value : 199
from pid : 2086 , value : 200
```

PRACTICAL : 5

Aim :

[B] In continuation of part (a), write a C program where the parent process waits for the child process to terminate.

Code (With assignment code):

```c
#include<sys/wait.h>
#include<stdio.h>
#include<sys/types.h>
int main()
{
        pid_t pid1,pid2,pid3,pid;
        int s;
        pid1=fork();

        if(pid1==0)
        {
                pid=getpid();
                printf("Child : %d\n",pid);
                printf("Child : now Sleep for some time..\n");
                sleep(2);
        }
        else if(pid1>0)
        {
                //Parent waiting for child to terminate
                pid=wait(&s);
                if(WIFEXITED(s))
                {
                printf("Parent : Child exited status : %d\n",WIFEXITED(s));
                }
                else
                {
                perror("Fork failed\n");
                }
        }
return 0;
}
```

Output :

```
jay@jay-VirtualBox:~$ gcc pr5.c -o pr5
jay@jay-VirtualBox:~$ ./pr5
Child : 3051
Child : now Sleep for some time..
Parent : Child exited status : 1
jay@jay-VirtualBox:~$
```

PRACTICAL : 6

Aim :

[C] Write a C program using execvp() system call which will count the characters from file ‘wc’, using program ‘p.c’.

Code :

*pr6.c

```c
#include<stdio.h>
int main()
{
        FILE* f;
        int count =0 ;
        char filename[200];
        char ch;

        printf("Enter file name :");
        scanf("%s",filename);
        char *const cmd[]={"filename", NULL};
        execvp(cmd[0],cmd);

        f=fopen(filename,"r");
        if(f==NULL)
        {
                printf("Can't open this file %s\n", filename);
                return 0;
        }
        for (ch=getc(f);ch!=EOF;ch++)
        {
                count+=1;
        }
        fclose(f);
        printf("Total number of Characters in file is : %d\n",count);
        return 0;
}
```

Output :

```
jay@jay-VirtualBox:~$ gcc pr6.c -o pr6
jay@jay-VirtualBox:~$ ./pr6
Enter file name :fil1.txt
Total number of Characters in file is : 149
jay@jay-VirtualBox:~$
```

TASK :

pr6task.c

```c
#include<stdio.h>
#include<stdlib.h>
int main()
{
        char *const cmd[]={"ls","-l",NULL};
        execvp(cmd[0],cmd);
        perror("Return from execvp()");
        exit(EXIT_FAILURE);
        return 0;
}
```

```
jay@jay-VirtualBox:~$ gedit pr6task.c
jay@jay-VirtualBox:~$ gcc pr6task.c -o pr6task
jay@jay-VirtualBox:~$ ./pr6task
total 300
-rw-rw-r-- 1 jay jay     4 Jan  7  2020 1.txt
-rw-rw-r-- 1 jay jay    10 Feb 16 19:29 2
-rw-rw-r-- 1 jay jay    10 Feb 16 19:29 3
-rw-r--r-- 1 jay jay     0 Jan  7  2020 7701
-rwxrwxr-x 1 jay jay  7196 Mar 10  2020 a.out
-rw-rw-r-- 1 jay jay   369 Feb  8 14:36 copy_dir.c
-rw-rw-r-- 1 jay jay   369 Feb  8 14:35 copy_dir.c~
-rwxrwxr-x 1 jay jay  7427 Jan 30 11:46 copy_file
-rw-rw-r-- 1 jay jay   849 Jan 30 11:59 copy_file.c
-rw-rw-r-- 1 jay jay   849 Jan 30 11:59 copy_file.c~
-rwxrwxr-x 1 jay jay  7420 Feb 16 19:05 cp
-rw-rw-r-- 1 jay jay  1709 Feb 16 19:38 cp.c
-rw-rw-r-- 1 jay jay  1701 Feb 16 19:37 cp.c~
-rwxrwxr-x 1 jay jay 11910 Feb 18 10:57 cpn
-rw-rw-r-- 1 jay jay  3918 Feb 18 10:52 cpn.c
-rw-rw-r-- 1 jay jay  3918 Feb 18 10:38 cpn.c~
drwxrwxr-x 4 jay jay  4096 Feb 18 09:19 d1
drwxr-xr-x 2 jay jay  4096 Feb 17 11:03 Desktop
drwxrwxr-x 3 jay jay  4096 Feb 17 13:01 dir1
-rw---Sr-x 1 jay jay     0 Feb 17 10:47 dirinside3
drwxr-xr-x 2 jay jay  4096 Dec 10  2019 Documents
drwxr-xr-x 2 jay jay  4096 Feb 17 15:58 Downloads
-rw-r--r-- 1 jay jay  8445 Dec 10  2019 examples.desktop
-rw-rw-r-- 1 jay jay    24 Feb 18 10:55 fil1.txt
-rw-rw-r-- 1 jay jay    24 Feb 18 10:56 fil2.txt
-rwxrwxr-x 1 jay jay    96 Mar  2  2020 fork.c
drwxrwxr-x 4 jay jay  4096 Feb 18 10:59 jay5
```

```
drwxrwxr-x 2 jay jay  4096 Feb 17 15:20 p`?
drwxr-xr-x 2 jay jay  4096 Dec 10  2019 Pictures
-rwxrwxr-x 1 jay jay  7318 Feb 15 14:30 pid.c
-rw-rw-r-- 1 jay jay   299 Feb 15 14:29 pid.c~
-rwxrwxr-x 1 jay jay  7236 Mar 19 11:19 pr4
-rw-rw-r-- 1 jay jay   233 Mar 19 11:19 pr4.c
-rw-rw-r-- 1 jay jay   233 Mar 19 11:18 pr4.c~
-rwxrwxr-x 1 jay jay  7323 Mar 19 11:26 pr4task
-rw-rw-r-- 1 jay jay   291 Mar 19 11:26 pr4task.c
-rw-rw-r-- 1 jay jay   290 Mar 19 11:26 pr4task.c~
-rwxrwxr-x 1 jay jay  7305 Mar 19 11:54 pr5
-rw-rw-r-- 1 jay jay   503 Mar 19 11:57 pr5.c
-rw-rw-r-- 1 jay jay   505 Mar 19 11:53 pr5.c~
-rwxrwxr-x 1 jay jay  7406 Mar 19 12:14 pr6
-rw-rw-r-- 1 jay jay   445 Mar 19 12:16 pr6.c
-rw-rw-r-- 1 jay jay   446 Mar 19 12:14 pr6.c~
-rwxrwxr-x 1 jay jay  7237 Mar 19 12:19 pr6task
-rw-rw-r-- 1 jay jay   175 Mar 19 12:19 pr6task.c
-rw-rw-r-- 1 jay jay   164 Mar 19 12:19 pr6task.c~
drwxr-xr-x 2 jay jay  4096 Dec 10  2019 Public
-rwxrwxr-x 1 jay jay   172 Jan  7  2020 task1.sh
-rwxrwxr-x 1 jay jay   172 Jan  7  2020 task1.sh~
-rwxrwxr-x 1 jay jay   163 Jan  7  2020 task2.sh
-rwxrwxr-x 1 jay jay   167 Jan  7  2020 task2.sh~
drwxr-xr-x 2 jay jay  4096 Dec 10  2019 Templates
-rw-rw-r-- 1 jay jay    10 Feb 17 15:33 test1.txt
-rw-rw-r-- 1 jay jay    10 Feb 17 15:33 test.txt
-rw-rw-r-- 1 jay jay   185 Jan  7  2020 Untitled Document
-rw-rw-r-- 1 jay jay     0 Jan  7  2020 Untitled Document~
drwxr-xr-x 2 jay jay  4096 Dec 10  2019 Videos
jay@jay-VirtualBox:~$
```