

## Practical 5

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

**Program Code :**

```
#include<sys/types.h>
#include<unistd.h>
#include <sys/wait.h>
#include<stdio.h>

int main(){
    pid_t c_pid, pid;
    int status;
    c_pid = fork(); // Parent process will receive process id of child process in c_pid
    while child process
        // will have value 0 for c_pid
        if( c_pid == 0 ){
            //child
            pid = getpid();
            printf("Child: %d: I'm the child: %d\n", pid, c_pid);
            printf("Child: sleeping for 2-seconds, then exiting with status 12\n");
            sleep(2);
        }

        else if (c_pid > 0){
            //parent
            //waiting for child to terminate
            pid = wait(&status);
            if ( WIFEXITED(status) ){
                printf("Parent: Child exited with status: %d\n",
WEXITSTATUS(status));
            }
        }

        else{
            //error: The return of fork() is negative
            perror("fork failed");
            //exit(2);
        }
    return 0; //success
}
```

```
1 #include<sys/types.h>
2 #include<unistd.h>
3 #include <sys/wait.h>
4 #include<stdio.h>
5
6 int main(){
7     pid_t c_pid, pid;
8     int status;
9     c_pid = fork(); // Parent process will receive process id of child process in c_pid while child process
10    // will have value 0 for c_pid
11    if( c_pid == 0 ){
12        //child
13        pid = getpid();
14        printf("Child: %d: I'm the child: %d\n", pid, c_pid);
15        printf("Child: sleeping for 2-seconds, then exiting with status 12\n");
16        sleep(2);
17    }
18
19    else if (c_pid > 0){
20        //parent
21        //waiting for child to terminate
22        pid = wait(&status);
23        if ( WIFEXITED(status) ){
24            printf("Parent: Child exited with status: %d\n", WEXITSTATUS(status));
25        }
26    }
27
28    else{
29        //error: The return of fork() is negative
30        perror("fork failed");
31        //exit(2);
32    }
33    return 0; //success
34 }
```

### Output :

```
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gedit prac5.c
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gcc prac5.c
prac5.c: In function 'main':
prac5.c:14:10: warning: too many arguments for format [-Wformat-extra-args]
 14 |     printf("Child: %d: I'm the child\n", pid, c_pid);
    |
    |
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gedit prac5.c
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ gcc prac5.c
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$ ./a.out
Child: 23457: I'm the child: 0
Child: sleeping for 2-seconds, then exiting with status 12
Parent: Child exited with status: 0
nishit@nishit-Inspiron:~/Desktop/ios practicals/prac 5$
```