

Program 1. Develop a C program to implement the Process system calls (fork (), exec(), wait(), create process, terminate process).

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

int main()
{
    pid_t pid;
    int status;

    // Fork a child process
    pid = fork();

    if (pid < 0)
    {
        // Error occurred
        fprintf(stderr, "Fork Failed\n");
        return 1;
    }
    else if (pid == 0)
    {
        // Child process
        printf("This is the child process with pid = %d\n", getpid());

        // Execute /bin/ls
        execl("/bin/ls", "ls", NULL);

        // If there is an error, print it and exit
        perror("execl failed");
        _exit(1);
    }
    else
    {
        // Wait for the child to complete
        printf("Parent process, PID = %u\n", getpid());
        waitpid(pid, &status, 0);
        printf("Child completed with pid = %d\n", pid);
    }
    return 0;
}
```

Output:

```
krishna@ubuntu:~/Pictures$ cc 1.c
krishna@ubuntu:~/Pictures$ ./a.out
Parent process, PID = 3621
This is the child process with pid = 3622
1.c  a.out
Child completed with pid = 3622
```