**BCS303 Program 1**

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

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

int main() {

pid\_t pid, child\_pid;

int status;

pid = fork();

if (pid < 0) {

perror("fork failed");

exit(EXIT\_FAILURE);

}

if (pid == 0) {

printf("Child process: PID = %d, PPID = %d\n", getpid(), getppid());

execlp("ls", "ls", NULL);

perror("execlp failed");

exit(EXIT\_FAILURE);

} else {

printf("Parent process: PID = %d, Child PID = %d\n", getpid(), pid);

child\_pid = wait(&status);

if (child\_pid == -1) {

perror("wait failed");

exit(EXIT\_FAILURE);

}

if (WIFEXITED(status)) {

printf("Child process exited with status %d\n", WEXITSTATUS(status));

} else {

printf("Child process did not exit normally\n");

}

}

return 0;

}

**OUTPUT:**

acharya@acharya:~/Desktop/program$ gcc prg1.c -o prg1

acharya@acharya:~/Desktop/program$ ./prg1

Parent process: PID = 4203, Child PID = 4204

Child process: PID = 4204, PPID = 4203

prg1 prg1.c

Child process exited *with* status 0

**Theory:**

[fork()](https://www.geeksforgeeks.org/c/fork-system-call/) system call is used to create a process generally known as child process and the process that created it is known as parent process.