# PRACTICAL – 5

**Aim:** **Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, stat, readdir, opendir.**

**A. Write a program to execute fork () and find out the process id by getpid() system call.**

**B. Write a program to execute following system call fork (), execl(), getpid(), exit(), wait() for a process.**

**C. Write a program to find out status of named file (program of working stat () system cal**

**A:** **Write a program to execute fork () and find out the process id by getpid() system call.**

**PROGRAM:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

int main(void)

{

pid\_t pid = 0;

pid = fork();

if (pid == 0)

{

printf("I am the child.\n");

}

if (pid > 0)

{

printf("I am the parent, the child is %d.\n", pid);

}

if (pid < 0)

{

perror("In fork():");

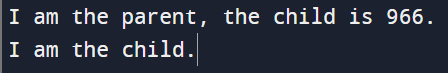
}

exit(0);

printf("\n20DCE019-Yatharth Chauhan\n");

}

**OUTPUT:**

****

**B. Write a program to execute following system call fork (), execl(), getpid(), exit(), wait() for a process.**

**PROGRAM:**

#include <stdio.h>

#include <unistd.h>

#include <stdlib.h>

#include <sys/types.h>

#include <sys/wait.h>

int main(void)

{

pid\_t pid = 0;

int status;

pid = fork();

if (pid == 0)

{

printf("I am the child.");

execl("/bin/ls", "ls", "-l", "/", (char \*)0);

perror("In exec(): ");

}

if (pid > 0)

{

printf("I am the parent, and the child is %d.\n", pid);

pid = wait(&status);

printf("End of process %d: ", pid);

if (WIFEXITED(status))

{

printf("The process ended with exit(%d).\n", WEXITSTATUS(status));

}

if (WIFSIGNALED(status))

{

printf("The process ended with kill -%d.\n", WTERMSIG(status));

}

}

if (pid < 0)

{

perror("In fork():");

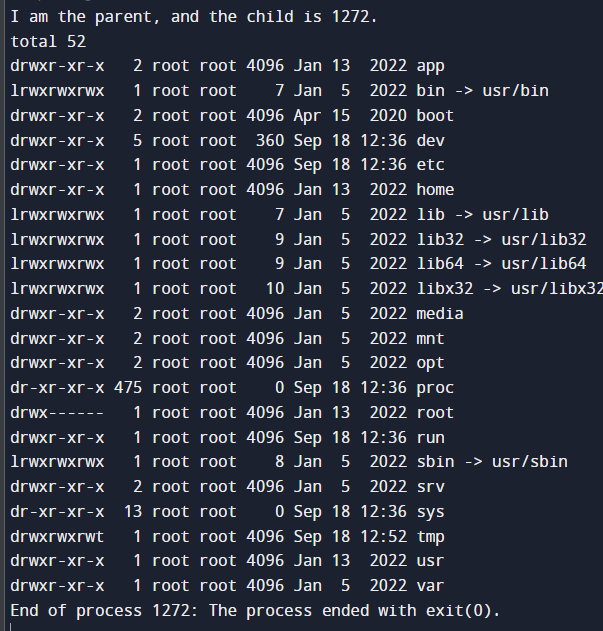
}

exit(0);

printf("\n20DCE019-Yatharth Chauhan\n");

}

**OUTPUT:**

****

**C:** **Write a program to find out status of named file (program of working stat () system call.**

**PROGRAM:**

#include <stdio.h>

#include <sys/stat.h>

int main()

{

struct stat sfile;

stat("stat.c", &sfile);

printf("st\_mode = %o", sfile.st\_mode);

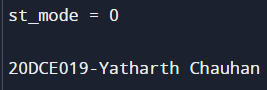
printf("\n");

printf("\n20DCE019-Yatharth Chauhan\n");

return 0;

}

**OUTPUT:**



**CONCLUSION:** In this practical we learnt program maintenance using make utility. We created a program spread over 4 files and maintained it using make utility.