Practical – 1

**Aim:-** Write a program for creating a child process using fork() system call. Print the process ID of the child and parent process. Implement the program in UNIX/Linux.

**Theory:**-

The fork() function is used to create a new process by duplicating the existing process. The new process is referred to as the child process, while the parent process. After the fork() call, both processes continue executing from the same point.

The fork() function returns different values for parent and child processes. In the parent process, the return value is the process ID (PID) of the child process. The fork() returns different values for the parent and child processes : a non-negative process ID (PID) for the parent and 0 for the child process. If the creation of the child class fails, fork returns a negative process ID.

**Code:**- C Language #include<stdio.h> #include<sys/types.h> #include<unistd.h> void childProcess(); void parentProcess(); void main(){

pid\_id pid = fork(); if(pid == 0){

childProcess();

}

else{

ParentProcess();

}

}

void childProcess(){

printf(“ child id %d”, getpid());

}

void parentProcess(){

printf(“ parent id %d”, getppid());

}

**Output:-**

