# 1a: Processing Environment

**Name : Hruhsikesh Vijaykumar Bhosale**
**PRN : 2020BTEIT00047**

**Objectives:**

1.To learn about Processing Environment.
2. To know the difference between fork/vfork and various execs variations.
3. Use of system call to write effective programs.

> **1.Write the application or program to create Childs assign the task to them by variation exec system calls. (B)**

**Code :**

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

int main() {
   pid_t child_pid;

   child_pid = fork();

   if (child_pid == 0) {

      // This is the child process
      char *args[] = {"/bin/ls", "-l", NULL};
      execvp(args[0], args);
      printf("This message will not be printed if execvp is successful\n");

   } else {

      // This is the parent process
      printf("Child process created with pid: %d\n", child_pid);
      wait(NULL);
      printf("Child process has completed\n");

   }
```

**Theory:**

**This program uses the fork system call to create three child processes. Each child process uses a different exec function from the exec family (execvp) to replace its own memory space with a different command. The first child process runs the "ls -l" command, the second child process runs the "grep root /etc/passwd" command, and the third child process runs the "date" command. The parent process then waits for all three child processes to complete before exiting.**
**It's also worth noting that the exit(0) after the execvp calls are to prevent the child from executing the parent code and in case the execvp fails.**