Name : Hrushikesh 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 program to use fork/ vfork system call and assign process to work as a shell. OR Read commands from standard input and execute them. Comment on the feature of this programe.

Code :

```
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <stdlib.h>

int main() {
   char command[100];
   int pid;

   while (1) {
      printf("$ "); // simulate a shell prompt
      scanf("%[^\n]%*c", command); // read command from standard input

      pid = fork(); // create a new process

      if (pid == 0) {
         // Child process
         system(command); // execute command
         exit(0); // exit child process
      } else {
         // Parent process
         wait(NULL); // wait for child process to exit
      }
   }

   return 0;
}
```

Theory:

In this program, the fork system call is used to create a new process for each command entered by the user. The child process uses the system function to execute the command, and the parent process uses the wait function to wait for the child process to exit. This allows the program to simulate a shell, where the user can enter commands and the program will execute them.

This program is a basic shell implementation that allows the user to enter commands and execute them. But it lacks several features that a typical shell would have such as redirecting input/output, handling background processes, handling environment variables, managing job control, and providing command line editing and history.

It's worth noting that, the program doesn't handle the case where the command entered by the user is not a valid command, in this case the program will call system with the entered command and it will return a non-zero status.