**Processing Environment**

**Subject - Unix Operating System**

**Name – Rachit Bhat**

**PRN – 22610007 Class – TYIT**

**Assignment No - 1e**

**Title-** 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.

**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.

**Theory-**

1. **Syntax-**

#include<stdlib.h>

int system(const char \*command);

1. **Description:**

system() executes a command specified in command by calling /bin/sh -c command,

and returns after the command has been completed. During execution of the

command, SIGCHLD will be blocked, and SIGINT and SIGQUIT will be ignored.

1. **Return Value:**

The value returned is -1 on error (e.g., fork(2) failed), and the return status of the

command otherwise. This latter return status is in the format specified in wait(2).

Thus, the exit code of the command will be WEXITSTATUS(status). In case /bin/sh

could not be executed, the exit status will be that of a command that does exit(127). If the value of command is NULL, system() returns nonzero if the shell is available, and zero if not. system() does not affect the wait status of any other children

**Program:**

#include<stdio.h>

#include<stdlib.h>

int main()

{

char str[256], buf[256];

printf("Enter command ");

scanf("%s",str);

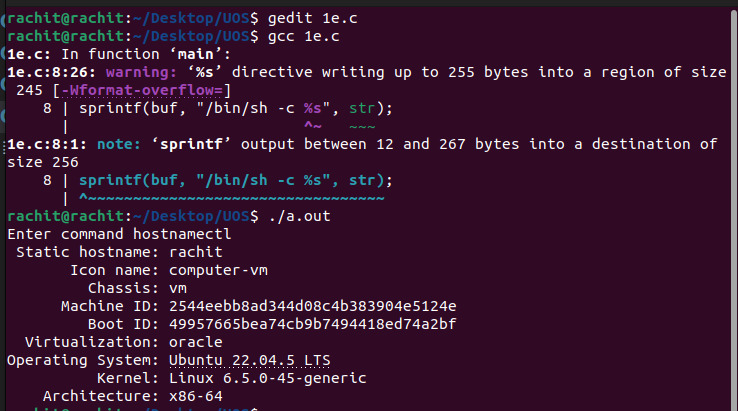
sprintf(buf, "/bin/sh -c %s", str);

system(buf);

return 0;

}

**Output:**

****

**Conclusion:**

system() can be used to perform various shell commands when the commands areread from standard input. The output of shell is printed.