

LAB 4 REPORT

-Shouvanik Chakrabarti (130050072)

-Krishna Harsha(130050076)

1)

We start a bash process by opening the gnome terminal. We can find the pid of this process by using `ps -aux | grep bash`.

We run `pstree -s <pid>` to get the process tree.

We get:

init——lightdm——lightdm——init——gnome-terminal——bash
as the process tree from init to the bash shell.

2)

We find executables for `ls` and `ps` in the `/bin` folder. Thus these are separate program which are exec'ed by the shell. There are no executables for `cd` and `history`. These are shell builtins.

Running the command

`type cd ps ls history`

will give us the same information.

3)

We run the `cpu1print` process and then go to the directory `/proc/<pid>/fd`.

Running `ls -l` here gives us the output:

lrwx----- 1 shouvanik ug13 64 Feb 2 19:16 0 -> /dev/pts/15

l-wx----- 1 shouvanik ug13 64 Feb 2 19:16 1 -> /tmp/tmp.txt

lrwx----- 1 shouvanik ug13 64 Feb 2 19:15 2 -> /dev/pts/15

Thus we see that the 0 and 2 (stdin and stderr) are mapped to the terminal from which the process was started. However 1 (stdout) is mapped to `/tmp/tmp.txt`. Thus all the standard output from the file is written into the file `/tmp/tmp.txt`.

4)

We run `ls -l` on the `/proc/<pid>/fd` folder for `./cpu1print` process and get

lrwx----- 1 shouvanik shouvanik 64 Feb 2 19:35 0 -> /dev/pts/22

l-wx----- 1 shouvanik shouvanik 64 Feb 2 19:35 1 -> pipe:[479130]

lrwx----- 1 shouvanik shouvanik 64 Feb 2 19:34 2 -> /dev/pts/22

We run `ls -l` on the `/proc/<pid>/fd` folder for `grep hello` process and get

lr-x----- 1 shouvanik shouvanik 64 Feb 2 19:36 0 -> pipe:[479130]

lrwx----- 1 shouvanik shouvanik 64 Feb 2 19:36 1 -> /dev/pts/22

lrwx----- 1 shouvanik shouvanik 64 Feb 2 19:34 2 -> /dev/pts/22

Thus we see that the stdout(1) for `cpu1print` and the stdin(0) for `grep` are mapped to the same file descriptor which is the `pipe:[479130]`. Thus the output from `cpu1print` becomes the input for `grep`, implementing the pipe.