LAB 4

Amit Malav :130050032
Lokit Kumar Paras : 130050047

---

1)
PID of bash: 1271
command used: <echo $$> or <ps aux | grep bash>

Process Tree:
init───lightdm───lightdm───init───gnome-terminal───bash
command used: pstree -s <bash PID>

---

2)

cd:	Implemented by bash
ls:	system program; exec'ed
history:	Implemented by bash
ps:	system program; exec'ed

command used: type -a <command>

---

3)
pid of the new process: 2269
/proc/2311/fd/0: symbolic link to `/dev/pts/11'
/proc/2311/fd/1: symbolic link to `/tmp/tmp.txt'
/proc/2311/fd/2: symbolic link to `/dev/pts/11'

command used: file /proc/<pid>/fd/<fd-id>

Description:
We can see that the file-descriptor(1) which was earlier pointing to stdout is now pointing to /tmp/tmp.txt file in the list of opened file descriptors. These manipulations are done by bash while forking the child process(cpu1print) which makes the child process print the output in /tmp/tmp.txt file instead of default stdout.

---

4)
./cpu1print and grep are spawned by bash.
fd info for ./cpu1print process:

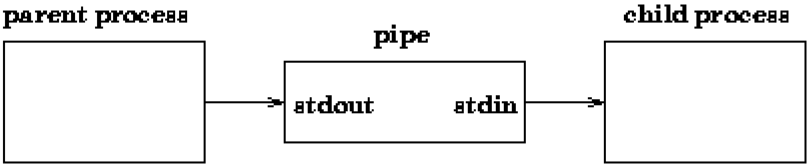
```
/proc/2878/fd/0: symbolic link to `/dev/pts/11'
/proc/2878/fd/1: broken symbolic link to `pipe:[691666]'
/proc/2878/fd/2: symbolic link to `/dev/pts/11'
```

fd info for grep process:

```
/proc/2879/fd/0: broken symbolic link to `pipe:[691666]'
/proc/2879/fd/1: symbolic link to `/dev/pts/11'
/proc/2879/fd/2: symbolic link to `/dev/pts/11'
```

Working of pipes:

When a pipe is used, the cpu1print is assumed to be writing to stdout and the grep is assumed to be reading from stdin. So, the pipe write device descriptor is assigned to stdout in the first process and the pipe read device descriptor to stdin in the second process.


parent process
pipe
stdout
stdin
child process