OS LAB 4

Karan Vaidya 130050019
Sivaprasad S 130050085

Exercises

1.
The pid of the bash shell is 6488
Command used : echo $BASHPID

The entire process tree is in tree.txt
That corresponding to bash shell is :
init(1) -> lightdm(1364) -> lightdm(1466) -> init(1727) -> gnome-terminal(6480) -> bash(6488) ;
Command used : pstree -p

2.
Built in the code : history, cd
Exec'ed : ls, ps

The executables corresponding to ls and ps are in /bin/ directory and not for others.
"exec" replaces the current process image with a new process image.
On running "exec ls" and "exec ps" shell's process image is replaced and the current shell is no longer accessible after doing this. They are system programs, that needs to be exec'ed.
Where as for "exec history" and "exec cd" the commands have to be built into the shell, as the executable was not found and shell was able to continue to run after running them.

3.
The pid of the process is 11678 (found from top)

lrwx------ 1 karan karan 64 Jan 30 14:18 fd/0 -> /dev/pts/7
l-wx------ 1 karan karan 64 Jan 30 14:18 fd/1 -> /tmp/tmp.txt
lrwx------ 1 karan karan 64 Jan 30 14:18 fd/2 -> /dev/pts/7

fd 0 and fd points to the corresponding pseudo terminal (in memory).
fd 1 points to the file where it's redirected to.
fd info is available in proc/fd proc/fdinfo proc/mountinfo

I/O redirection is implemented in the shell by manipulating of the file descriptor table.
To implement output redirection from a command, the shell forks a child process to implement the command, opens the file to write the output to, and “copies” the file descriptor information of the opened file into the file descriptor entry of the child’s standard output. Similarly for input redirection.

4.
The bash spawns two children : cpu1print - 11929 and grep - 11930

cpu1print :
lrwx------ 1 karan karan 64 Jan 30 14:32 fd/0 -> /dev/pts/7
l-wx------ 1 karan karan 64 Jan 30 14:22 fd/1 -> pipe:[231542]

grep :
lr-x------ 1 karan karan 64 Jan 30 14:22 ../11930/fd/0 -> pipe:[231542]
lrwx------ 1 karan karan 64 Jan 30 14:22 ../11930/fd/1 -> /dev/pts/7

bash:
lrwx------ 1 karan karan 64 Jan 30 14:21 ../11933/fd/0 -> /dev/pts/12
lrwx------ 1 karan karan 64 Jan 30 14:21 ../11933/fd/1 -> /dev/pts/12

From above data, we infer the following :-
a) The standard output of cpu1print is redirected to a buffer that corresponds to the pipe.
b) The standard input to grep is also redirected from the same file.
b) The stdin and stdout of bash, stdin of cpu1print , stdout of grep, all point to the corresponding pseudoterminals

Pipe is a half-duplex connection between two processes. A pair of fds are bound using the pipe. The parent process creates a pipe and hands over the read end to grep and write end to cpu1print. bash first creates a pipe that redirects its fds 0,1 to the different buffers (provided by pipe), then it forks two processes and redirects the fds back to the corresponding pseudoterminals. cpu1print redirects it's stdin and grep redirects it's stdout back. The two buffer are maintaned by the OS. Thus the output of cpu1print is redirected to the input of grep.