

Shell

Rupesh Nasre.
rupesh@cse.iitm.ac.in

Shell

- Shell is also an interface between user and the computer system.
 - Permits text-based interaction
 - Enables automation (shell scripts)
- It is also a piece of software.
- A shell can be developed and deployed by users.
 - You may develop one in the OS course.
 - Your system may have multiple shells (sh, bash, csh, ...)

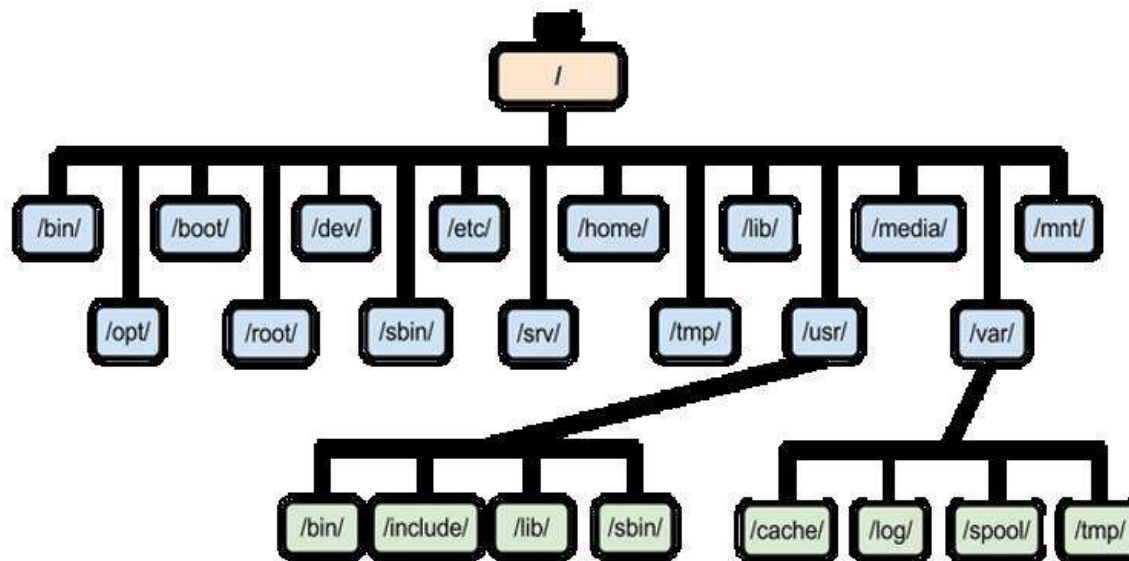


```
#!/bin/bash

~root: env X="() { :; } ; echo shellshock" /bin/sh -c "echo completed"
> shellshock:
> completed
```

Terminal Commands

- Readonly: ls, pwd, cat, cd
- Create: cp, mkdir, vi, emacs
- Remove: rm, rmdir, mv
- Others we will learn later (sed, grep, awk, ...)
- Basic terminal commands



Directories

- Create: `mkdir`
- Remove: `rmdir` (the directory must be empty)
 - Alternatively, `rm -r <directory>` (be careful)
- Navigate: `cd`
- Current directory is `.` (dot)
- Parent directory is `..` (dot-dot)
- Grand-parent directory is `../..`
- Absolute path: `/home/cs25b001/cs1234/L3`
 - Relative path (relative to current directory):
`../cs1234/L3`
- Present working directory: `pwd`
- Go to the home directory: `cd` `#/home/cs25b001`
- Go to the last directory: `cd -` `#toggles`

```
:mkdir L3
```

```
:mkdir L4
```

```
:mkdir L5 L6
```

```
:ls
```

```
L3    L4    L5    L6
```

```
:cd L3
```

```
:cat >1.txt          # alternatively, use vi/emacs
```

```
This is some text.  # press Ctrl-D to end the input
```

```
:cat 1.txt
```

```
This is some text.
```

```
:pwd
```

```
/home/hp/teaching/ssad/jan26/work/L3
```

```
:cd ..
```

```
:rmdir L3 L6
```

```
rmdir: failed to remove 'L3': Directory not empty
```

```
:ls
```

```
L3    L4    L5
```

```
:rm -r L3
```

```
:ls
```

```
L4    L5
```

Files

- **Create:** `vi`, `emacs`, `touch`, `>`
- **Remove:** `rm`
- **Rename:** `mv` (be careful)
- **Copy:** `cp`
- **Read:** `vi`, `emacs`, `cat`, `<`
- **Move:** `mv`

```
:ls
L4      L5
:cd L4
:vi starter.c    #write some code
:ls
starter.c
:touch starter2.c
:ls
starter2.c starter.c
:cat starter.c
#include <stdio.h>
int main() {}
:cat starter2.c
:mv starter.c starter2.c #overwrite
:ls
starter2.c
:cp starter2.c starter.c
```

```
:mv starter2.c ../L5
:ls ../L5
starter2.c
:ls
starter.c
:mv ../L5/starter2.c starter.c #overwrite
:ls ../L5
:ls
starter.c
:rm starter.c
:ls
:
```

Command Prompt

- Shell has a full-fledged programming language.
- PS1 variable
 - Arbitrary text with `\u`, `\h`, `\w`, `\W`, `\n`, `\t`, `\d`
 - Set variable: `export`
 - `export PS1="\[\e[1;42m\]\[\e[5;31m\]\w\\$ \[\e[0m\]"`
 - Print variable: `echo $PS1` # all variables with `env`
 - **Homework:** Find out what PS2 is.
- Navigate through commands: Up/down arrow keys
- Search through previous commands: `Ctrl-r`
- Execute previous command starting with `xyz`: `!xyz`
- Repeat previous command: `!!`
- Copy last argument of the previous command: `Esc.`
- Complete command / filename: `tab`
- To run executable as `a.out` (instead of `./a.out`)
 - `export PATH=$PATH:`

ls

- List directory contents
 - Home directory listing: `ls ~`
 - Arbitrary directory: `ls /home/cs25b001/cs1234/L3`
#absolute path
 - Arbitrary directory: `ls ~/cs1234/L3` #using ~
 - Arbitrary directory: `ls ../L3` #relative path
- Does not show hidden files (names starting with .)
 - show hidden files and directories: `ls -a`
- Long listing with details: `ls -l` # permissions, timestamp, filesize
- Recursively all the subdirectories: `ls -R ~`
- Reverse the order of listing: `ls -r`
- Chronological order: `ls -rt ~/Downloads`

In this course...

Application	Programming	Tools
Puzzle of grass, goat, lion	Pointers, structures	Editor, visualization
Data maintenance	File I/O	I/O redirection, make
Item grouping	Linked Lists	Debugging
Gene sequencing	Recursion, strings	Memory profiling, ulimit
Closest match	Matrices	Compute profiling, plotting
Keep the recent	Arrays	Version control
Cryptography	Strings, bitwise operators	Library (ncurses)
Unit testing	Command-line	Testing
Web-page creation, analysis	HTML, Strings	wget, shell scripting
Game	Graphics	Library (graphics)
Basics	Python	--

Our goal: *to have fun with automation*