

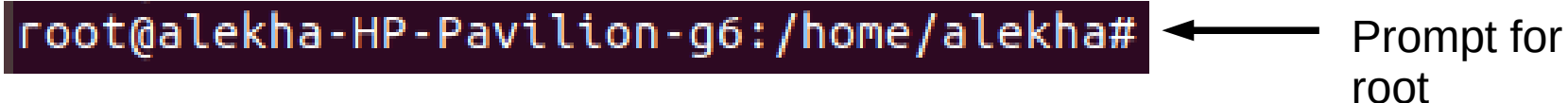
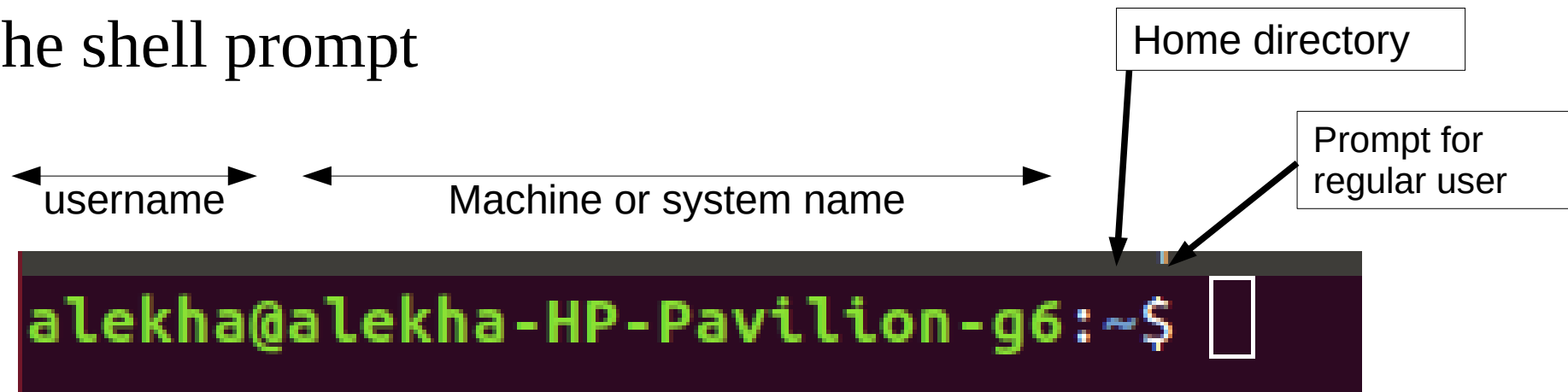
Unix & Shell Programming

Shell

- This is one way to talk to computer
- It is a command line interface
- It allow to execute commands
- In UNIX, shell is separated from other units of operating system.
- Shell is just another program in UNIX whose task is to execute other programs behalf of a user
- Most popular/latest shell in UNIX is bourne shell or bash
- Other shells include C shell, korn shell etc.

bash shell

- It is a command line interpreter that performs the following tasks
 - Launches programs
 - Takes command
 - Shell scripting (sequence of actions)
- The shell prompt



bash shell

- Bash shell keeps the list of directory to search for command in an environmental variable (builtin) called PATH
- We can use echo command to display the PATH variable

```
alekha@alekha-HP-Pavilion-g6:~$ echo $PATH
/home/alekha/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/java-8-oracle/jre/bin
alekha@alekha-HP-Pavilion-g6:~$
```

UNIX Essential Commands

- **clear**
 - Clear the screen
- **who**
 - Prompts user name with login time currently logged to this system
- **whoami**
 - Prompt the name of the current user
- **ls**
 - Lists all the files and directories.
 - Accepts the path to the listing directory as argument(s).
 - Also accept patterns (RE) for selecting files and directories.

Some options for ls

```
$ ls chap* ↵
```

- list all files and directories with “chap”

```
$ ls -l ↵
```

- detailed listing of files and directories

```
$ ls -a ↵
```

- include hidden files (starts with ‘.’)

```
ls -d */ ↵
```

- Listing only directories

```
ls -lh ↵
```

- Shows size in human readable format

```
ls -r ↵
```

- Show files in reverse order

```
ls -R ↵
```

- list recursively directory tree

UNIX Essential Commands(2)

- **cat**
 - Display the content of a file
\$ cat filename ↵
- > directs the output of a command to an output file. It overwrites the output file if already exists
\$ ls -l > list.txt ↵
\$ cat list.txt ↵
- >> directs and append the output of a command to an existing file. (Does not overwrites)
\$ who >> list.txt ↵
\$ cat list ↵

UNIX Essential Commands(3)

- **wc**

- Displays number of lines, number of words, and number of characters in a given file

```
$ wc filename ↵
```

- Output format : #oflines #noofwords #noofchars filename

- **wc options**

```
$ wc -l filename ↵ (only number of lines)
```

```
$ wc -w filename ↵ (only number of words)
```

```
$ wc -c filename ↵ (only number of characters)
```

- **| ('the pipe')**

- Feeds output of one command to another command

```
$ ls -l | wc -l ↵
```


UNIX Essential Commands(4)

- **echo**

- Display strings, variables, constants etc.

```
$ echo 'hello' ↵
```

```
$ echo hello ↵
```

```
$ x = 5 ↵
```

```
$ echo $x ↵
```

- **type**

- It tells the location of a unix command/program

```
$ type ls ↵
```

```
ls is /bin/ls
```

- **uname**

- Prompts the version of the linux/unix (kernel)

UNIX Essential Commands(5)

- **man**

- Help manual for command
- \$ man commandname ↵
- Alternate way : \$ command - - help ↵

- **less**

- Read the content of a file one page at a time. (allow scrolling)
- \$ less myfile.txt ↵
- \$ man ls | less ↵

- **more**

- Read the content of a file one page at a time (does not allow scrolling of pages)
- \$ more myfile.txt ↵
- \$ man ls | more ↵

UNIX Essential Commands(6)

- date
 - Displays data and time
- Date printing format options
 - Options always prefixed with +
%
 - d : day of the month
 - y : last two digit of the year
 - m : month in number
 - h : month in name
 - H, M, S : hour, minute, and second

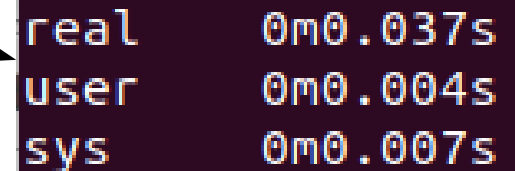
\$ date +%m ↵

\$ date +%h ↵

\$ date +%H:%M:%S ↵

UNIX Essential Commands(7)

- **passwd**
 - Change the current password of the user
- **bc**
 - The command line calculator
- **time**
 - Shows time taken by a program execution at various resources
 - `$ time sort -o netlist invoice.lst ↵`
 - real : total time elapsed
 - user : time spent executing itself
 - sys : time spent by linux kernel system



A terminal window showing the output of the 'time' command. The output consists of three lines: 'real 0m0.037s', 'user 0m0.004s', and 'sys 0m0.007s'. An arrow points from the command line in the previous block to this output.

real	0m0.037s
user	0m0.004s
sys	0m0.007s

UNIX Essential Commands(8)

- **pwd**
 - Current working location or present working directory
- **cd**
 - Change directory by either providing sub-directory name or a director path
 - \$ cd Documents ↵
 - \$ cd /home/alekha/Downloads ↵
- Special notations for directories
 - ./ (dot slash)- current directory
 - .. (double dots)- parent directory
 - / (slash) - root directory

UNIX Essential Commands(9)

- **mkdir**
 - Create a new directory with the name or the directory path provide as argument
- **rm**
 - Remove a directory either by providing sub-directory name or the directory path
- **cp**
 - Copy a file from the source location (path) to destination location (path)
 - Syntax : `$ cp sourcepath destinationpath ↵`
- **mv**
 - Move or rename a file
 - Syntax : `$ mv oldfilepath newfilepath ↵`
- Some necessary options for **mv**
 - i : interactive mode
 - r : recursive mode (when a directory contains subdirectories)
 - f : forece mode (opposite to interactive mode)

UNIX Essential Commands(10)

- **cmp**

- Compares the content of two given files. It shows the difference in the file if exists. However, shows no message if two files are identical.

```
$ cmp file1 file2 ↵
```

- **comm**

- Finds common in two files

```
$ comm file1 file2 ↵
```

- **diff**

- Displays file differences

```
$ diff file1 file2 ↵
```

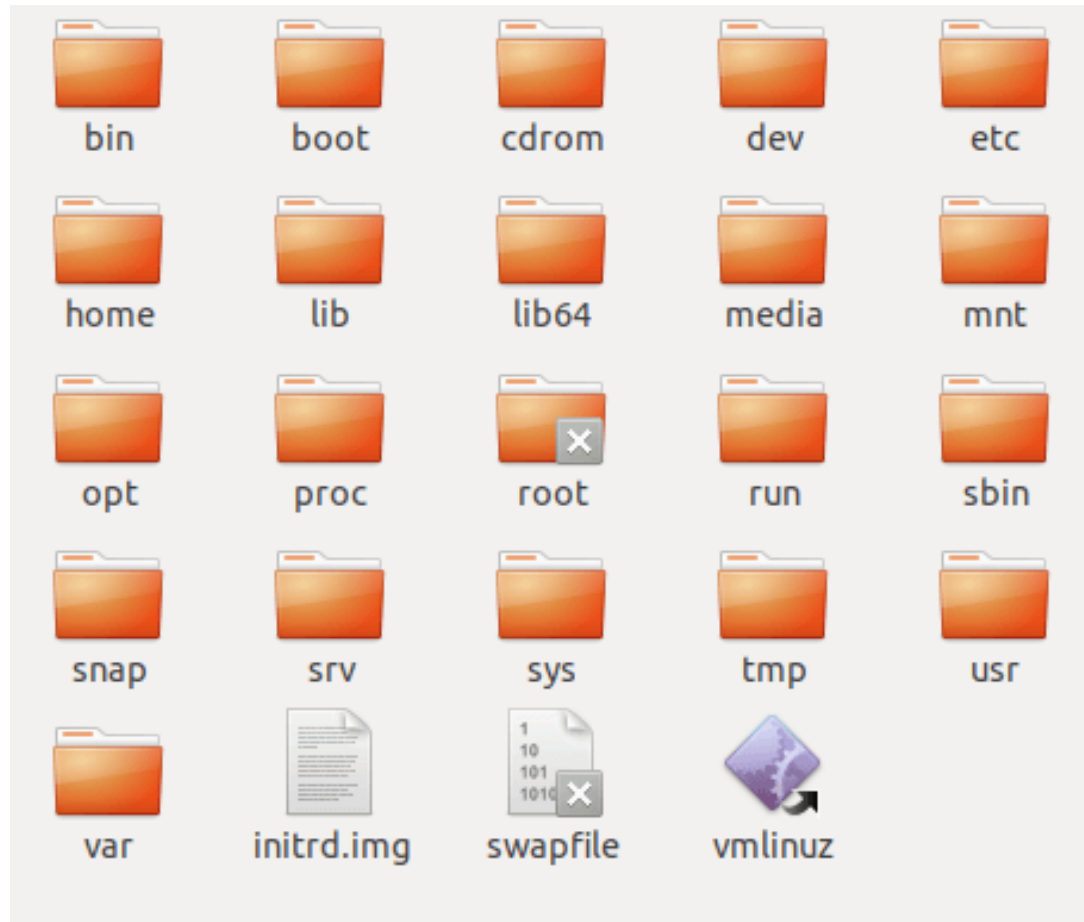
chmod command

- It is used to change the permission of a file
- Types of permission
 - Read (r)
 - Write (w)
 - Execute (x)
- Types of user
 - User (u)
 - Group (g)
 - Other (o)
 - For All (a)
- Example
 - \$ chmod u+x filename ↵
 - \$ chmod ugo+x filename ↵
 - \$ chmod a+x filename ↵
- \$ chmod go-r, u-x filename ↵
- Numeric representation of permission
 - 9 bits
 - r w x r w x r w x
 - user group others
- Example
 - 1 1 1 1 0 0 1 0 0
 - 744
 - \$ chmod 744 filename ↵
- \$ chmod 777 filename ↵ (all permission to all)
- \$ chmod 444 filename ↵ (read permission to all)
- \$ chmod 222 filename ↵ (write permission to all)
- \$ chmod 111 filename ↵ (execute permission to all)

UNIX Essential Commands(11)

- Using semicolon (;) we can execute multiple commands in a single line
 - `$chmod 666 filename ; ls -l filename ; mkdir myfiles ↵`
- **sort**
 - Ordering of a file
 - `$ sort filename ↵`
 - `$ sort -r filename ↵`
- **uniq**
 - Locate repeated lines in a file
 - `$ uniq filename ↵`

Linux file system structure



Linux file system structure(2)

- The root (/) contains the entire operating system
- **/bin** – binary files such as linux terminal commands
- **/boot** – necessary files form system boot
- **/dev** – physical devices that are mounted here such as USB, External HD etc. Other partitions are also visible here.
- **/etc** – stores all configuration files. Typically the configurations stored here affect all the users.
- **/home** – home(local) folder of all users.
- **/lib** – stores all library files, additional libraries, or each package installed.
- **/media** – Alternate for external storage and drives

Linux file system structure(3)

- **/mnt** – a placeholder folder used for mounting other folders and drives.
Typically for network locations
- **/opt** – optional softwares for the system that are not managed by distro package manager
- **/proc** – processes folder, where a lot of system information is represented as file
- **/root** – the home folder for the root user
- **/sbin** – similar to /bin, except that it is dedicated to certain commands that can only be run by the root user/super user
- **/tmp** – stores temporary files
- **/usr** – contains files and utilities that are shared by user
- **/var** – the variable data are kept (such as system logs)

Processes in UNIX

- **ps**

- This command tells about all the processes that are currently running in os.
- `$ ps ↵` : shows only the process of the current user
- `$ ps -a ↵` : shows processes of all users
- `$ ps -f ↵` : shows full listing of processes

- **'&'**

- This character is used to run a process in the background
- `$ sort emp.txt > empsort.txt & ↵`
22310 (returns the pid of the process)

- **kill**

- Kill or terminates a process
- `$ kill pid ↵`
- `$ kill -9 pid ↵` : forcibly kills a current running process with given pid.

grep command and regular expression

- The grep command scans for occurrence of a pattern and can display the selected text matching the pattern
 - Syntax : `$ grep options pattern filename(s)`
 - Here, pattern accept text and regular expression
- Example
 - `$ grep sales emp.txt ↵`
 - This command displays all the lines in the file ‘emp.txt’ containing the term ‘sales’
- Grep is one of the UNIX command that silently returns the prompt in case no pattern match found in the file.

grep command(2)

- Grep can use a series of string in the argument, where first argument as the pattern, and rest as filenames.
 - In this case, the output shall be displayed along with filenames.
 - \$ grep director emp1.txt emp2.txt ←

```
alekha@alekha-OptiPlex-3060:~/shellprog$ grep SALESMAN emp1.txt emp2.txt
emp1.txt:7499 ALLEN SALESMAN 7698 20-FEB-1981 1600 30
emp1.txt:7521 WARD SALESMAN 7698 22-FEB-1981 1250 30
emp1.txt:7654 MARTIN SALESMAN 7698 28-SEP-1981 1250 30
emp1.txt:7844 TURNER SALESMAN 7698 8-SEP-1981 1500 30
emp2.txt:7499 JOHN SALESMAN 7698 20-FEB-1981 1600 30
emp2.txt:7521 STEVE SALESMAN 7698 22-FEB-1981 1250 30
emp2.txt:7654 SUNNY SALESMAN 7698 28-SEP-1981 1250 30
alekha@alekha-OptiPlex-3060:~/shellprog$
```

grep command(3)

- If the pattern contains whitespace, then it is mandatory to quote the pattern.
- Otherwise, the first word of the pattern shall be treated as actual pattern and rest of the words as filenames
 - `$ grep 'SCOTT ANALYST' emp1.txt ←`

grep command options

- -C
 - It is used to count the occurrence of the pattern
 - `$ grep -c 'MANAGER' emp1.txt ↵`
- -n
 - Displays the line numbers of the line of pattern match
- -v
 - Option to select all but lines containing the pattern
- -l
 - Displays only filename containing the pattern
 - `$ grep -l 'MANAGER' *.txt`
- -i
 - Ignore case in pattern

Grep with RE

- []
- *
- .
- ^
- \$
- +
- ?