TATA CONSULTANCY SERVICES

TATA

Understanding UNIX and Shell Scripting

Mayank Gupta (#764606)

TATA CONSULTANCY SERVICES

Session 1

Experience certainty.

IT Services
Business Solutions
Outsourcing

Introduction to UNIX

UNIX is an operating system

An operating system is the program that controls all the other parts of a computer system, both the hardware and the software. It allocates the computer's resources and schedules tasks. It allows you to make use of the facilities provided by the system. Every computer requires an operating system.

It is a **C**ommand **U**ser **I**nterface Operating System.

Developed by Ken Thompson and Ritchie

Features of UNIX

Performance: In load condition, Unix performs better than WOS.

Security: Every inch of Unix is secured. It is ultimate OS.

Introduction to UNIX

Price: Free available.

Portability: UNIX is a machine independent operating system

Stability: Unix is a hands down winner over all other OS. High-up time and high availability.

UNIX is a **multi-user, multi-tasking operating system.**

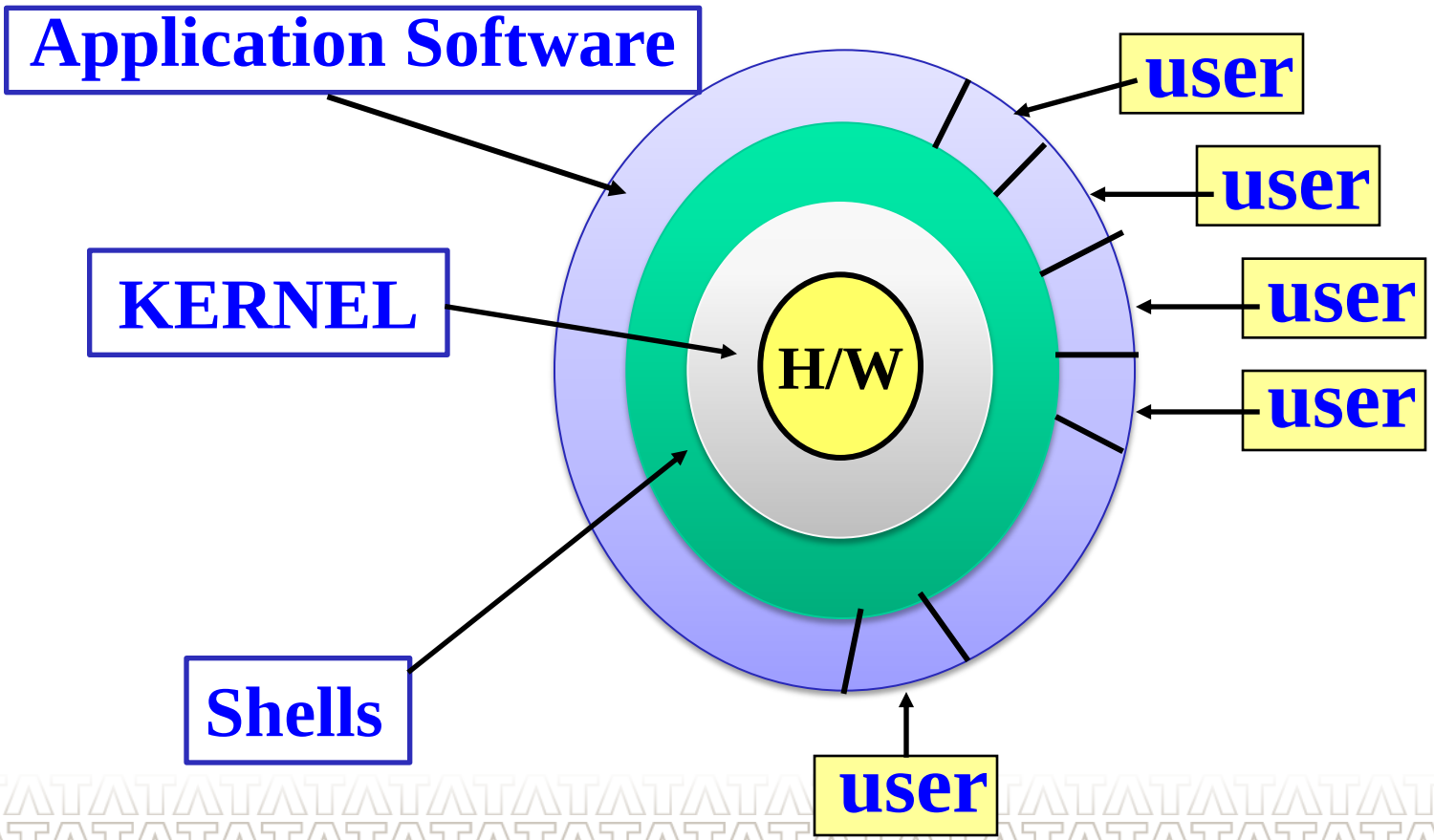
UNIX Flavors

There are many versions of Unix that are used by users,

- SysV (from AT&T)
- BSD (from Berkeley)
- Solaris (Sun)
- IRIX (SGI)
- AIX (IBM)
- LINUX (free software)

UNIX System Architecture

Shell and kernel together make UNIX system work.



UNIX: File System

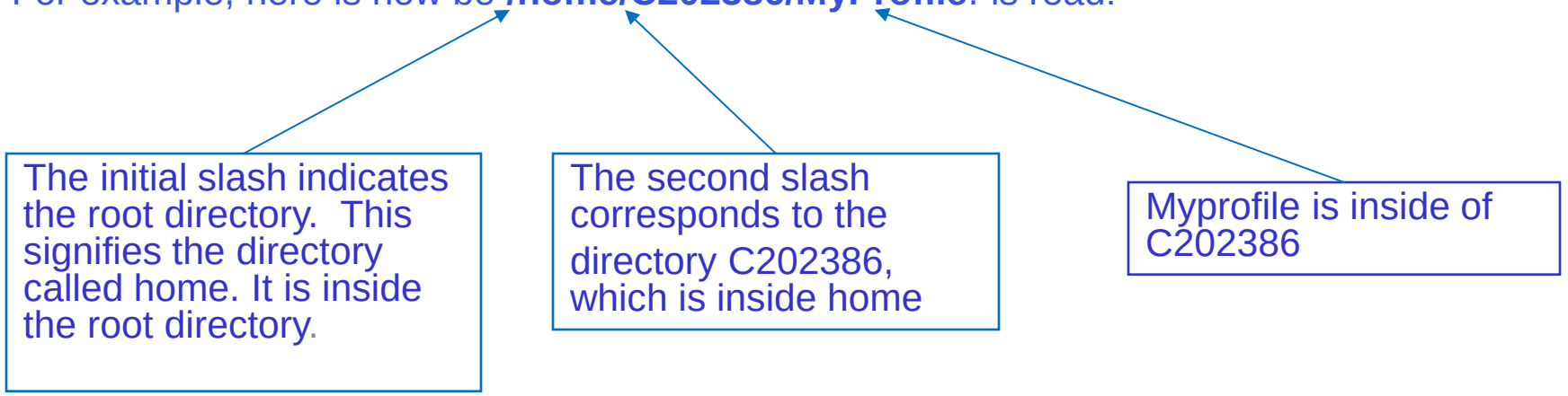
In unix, the collection of directories and files is called **the file system.**

A directory is like a folder: it contains files.

A large folder can even hold other folders-directories can be inside directories

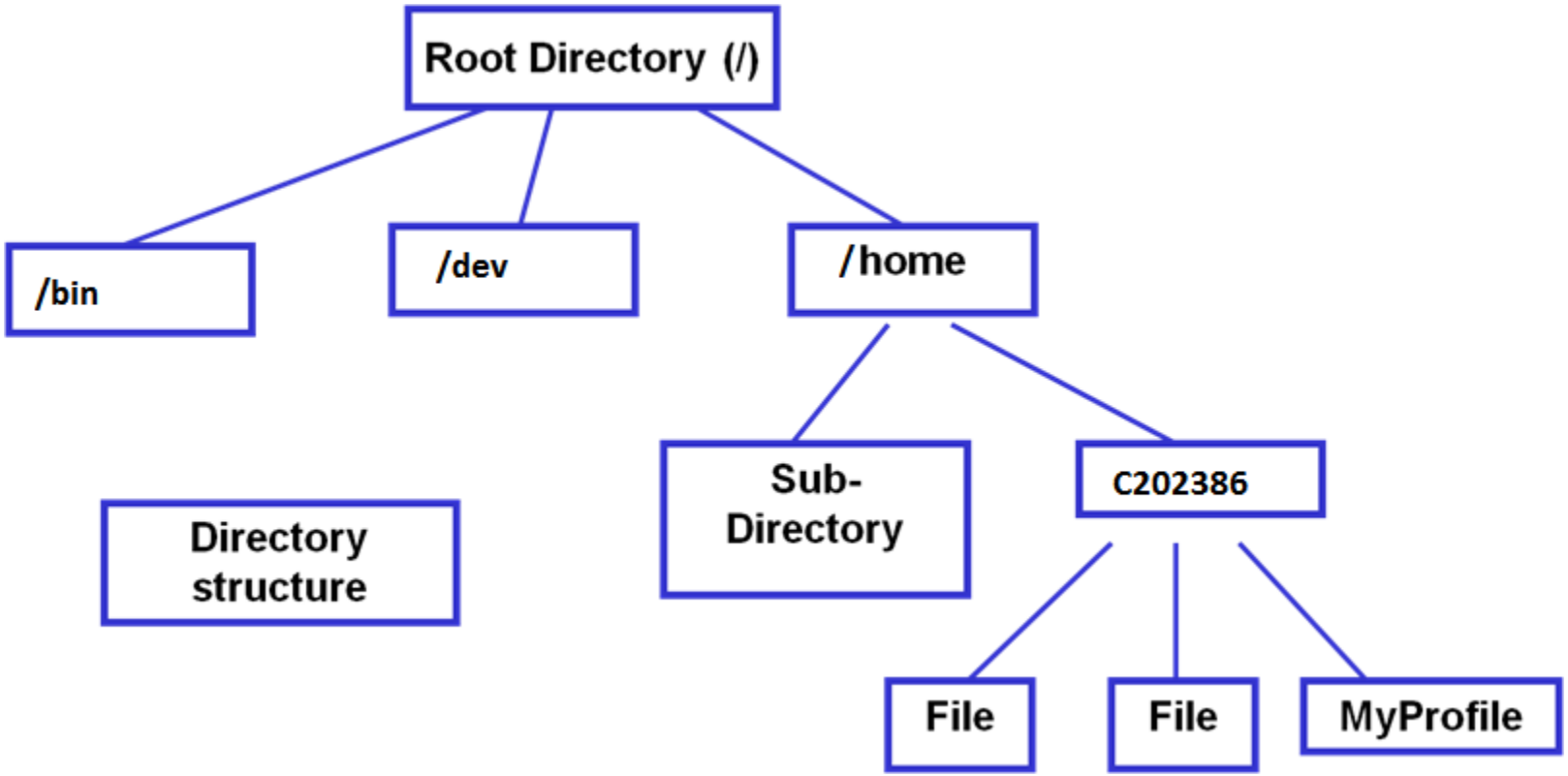
The path can be divide into a sequence of directories.

For example, here is how be **/home/C202386/MyProfile**. is read:



UNIX Directory Structure

A path could refer to either a directory or a filename, so MyProfile could be either.
All the items before the short name must be directories.



UNIX Commands

Command Format

Format: command name and 0 or more arguments:

$ commandname [-][option] [arg1] ... [argN]

Arguments can be operands, basically the data to work with (actual data, or a file name)

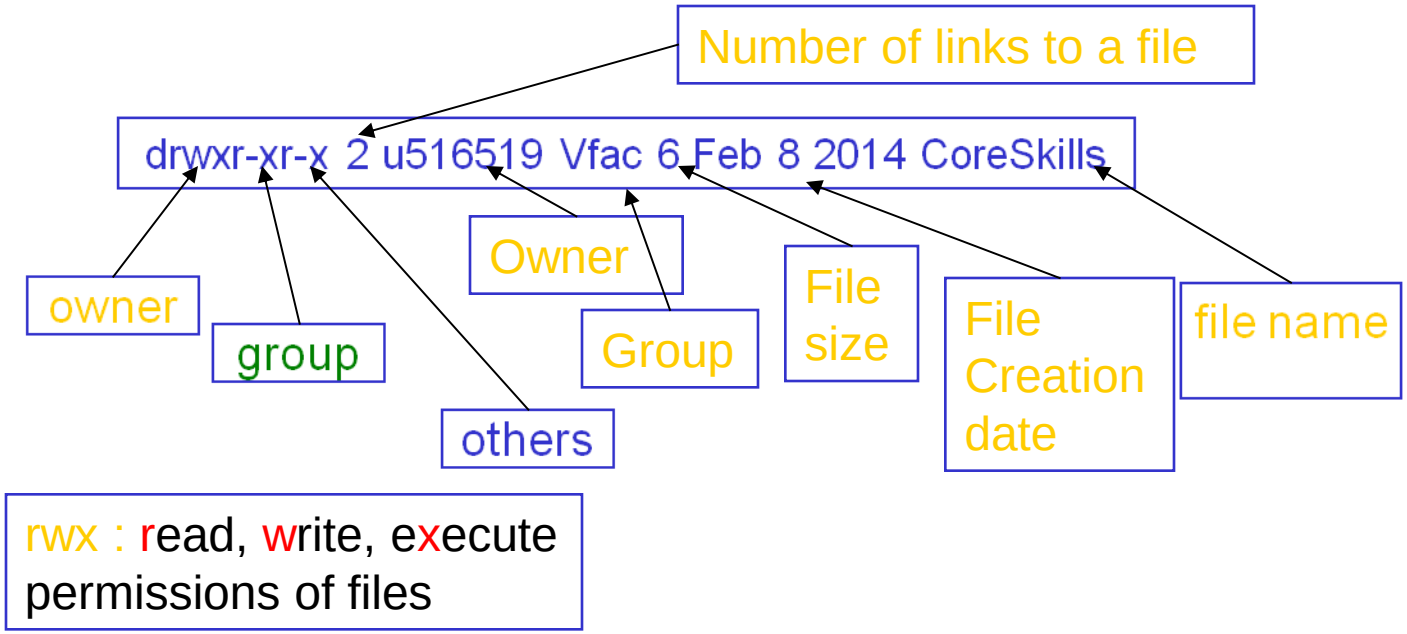
An option is a special kind of parameter that starts with a dash “-”

An option modifies how the program runs, but not what the program runs on.

UNIX Commands

ls

ls command is use to list files and directories inside all directories.



Type of file:
- – plain file
d – directory
s – symbolic link
c - Device file

Command : pwd

pwd (present working directory) tells you your current directory. Most commands act, by default, on the current directory. For instance, ls without any parameters displays the contents of the current directory.

$ pwd

Command : cd

cd is used to change directories.
The format of this command :
cd new-directory (where new-directory is the name of the new directory you want).

$ cd /home/

Command : mkdir

mkdir (make directory) is used to create a new directory,
It can take more than one parameter, interpreting each parameter as another directory to create.
$ mkdir “dirname”
$ mkdir Cprograms

Introduction to UNIX

Command : rmdir

rmdir (remove directory) is used to remove a directory,
rmdir will refuse to remove a non-existant directory, as well as a directory that has anything in it.

The primary commands for manipulating files under unix are cp, mv, and rm. They stand for copy, move, and remove, respectively.

Command : cp

cp is used to copy contents of file1 to file2
cp file1 file2 (contents of file1 is copied to file2 in the same directory)
cp folder1/file1 folder2 (contents of file1 is copied to file1 in the inside of folder2 directory)

Command : rm

rm is used to remove a file.
rm filename ---> removes a file named filename

Command : mv

mv is used to move a file.
mv will rename a file if the second parameter is a file. If the second parameter is a directory, mv will move the file to the new directory, keeping it's shortname the same.
mv file1 file2
mv file1 ./test/file1

Command : cat

cat command is used to concatenate or displays the contents of a file.
To use it, type cat, and then pres

```
$ cat
```

$ cat file1 [file2] ...

Will concatenate all files in one and print them to STDOUT

$ cat > filename

Will take whatever you type from STDIN and will put it into the file filename

Chmod

Chmod (change mode) is used to change the permissions on a file.
(owner) (group) (others)
chmod [number][number][number] file1
Number = (read)4 + (write)2 + (execute)1
Example: Chmod 754 file1
for owner: read, write and execute permissions (4+2+1)
for group: read and execute permissions (4+0+1)
for others: only read permission (4+0+0)

Consider permission for each set of users (user, group, other) as a 3-bit #
r – 4
w – 2
x – 1
A permission (mode) for all 3 classes is a 3-digit octal #
755 – rwxr-xr-x
644 – rw-r—r--
700 – rwx------

df

df (disk filling) summarizes the amount of disk space in use.
For each file system, it shows the total amount of disk space, the amount used, the amount available, and the total capacity of the file system that's used.

Wc

wc (word count) simply counts the number of words, lines, and characters in the file(s).

wc [-clw] [file1 file2 ... fileN]

The three parameters, clw, stand for character, line, and word respectively, and tell wc which of the three to count.

echo

It is a standard output commands use to print string.

Introduction to Environment Variable

Environmental Variables

Some examples of the environment variables are the USER, LOGNAME, HOME, PS1etc.

The **HOME** Variable

It specifies an associated directory with every user in a UNIX system. If the HOME variable for the user Sita contains /usr/sita/stores, every time Sita logs in, she is taken to the directory stores. The variable HOME is referenced the same way,

The **PATH** Variable contains a list of all full path-names (separated by a colon) of directories that are to be searched for an executable program. For example, the command

$PATH=.:/usr/bin:/bin<Enter> specifies directories to be searched for any executable file or a command file (current directory, /usr/bin and /bin, in that order).

The **PS1** Variable

The system prompt may be changed by setting the value of this variable to the desired prompt:

$ PS1="Hello>"<Enter>

Hello> #can be changed only at the UNIX command line, not within a shell script.

TATA CONSULTANCY SERVICES

TATA

Session 2

Experience certainty.

IT Services
Business Solutions
Outsourcing

VI editor

The vi editor is available on almost all Unix systems. vi can be used from any type of terminal because it does not depend on arrow keys and function keys--it uses the standard alphabetic keys for commands.

It displays a window into the file being edited that shows lines of text

This help note explains the basics of vi:

- opening and closing a file
- moving around in a file
- elementary editing

Modes of vi

vi has three modes:

- command mode
- insert mode
- colon mode **:**

Moving the Cursor

To move the cursor to another position, you must be in command mode. If you have just finished typing text, you are still in insert mode. Go back to command mode by pressing <Esc>. If you are not sure which mode you are in, press <Esc> once or twice until you hear a beep. When you hear the beep, you are in command mode.
The cursor is controlled with four keys: h, j, k, l.

Key	Cursor Movement
h	left one space
j	down one line
k	up one line
l	right one space

Basic Editing

Editing commands require that you be command mode. Many of the editing commands have a different function depending on whether they are typed as upper- or lowercase. Often, editing commands can be preceded by a number to indicate a repetition of the command.

Shell Script

Shell Script is a basically collection of commands in a file. When we run the file the shell will execute the command one by one. This is the simple shell script, Example is given below

```
$ vi sample.sh
clear
date
ls
who
ls –l
echo “hello”
```

With above commands save file

Give execute permission to the file by using chmod command

```
$ chmod 755 sample.sh
```

To run the script

```
$ sh sample.sh
```

Or

```
$ ./sample.sh
```

Conditional Statements

if condition

if condition which is used for decision making in shell script, If given condition is true then command1 is executed.

Syntax:

if condition

then

command1 if condition is true or if exit status of condition is 0 (zero) ... ...

fi

Conditional Statements

The case Statement

The case statement is good alternative to Multilevel if-then-else-fi statement. It enable you to match several values against one variable. Its easier to read and write.

Syntax:

```
case  $variable-name  in
    pattern1)   command
                ...
                command;;
    pattern2)   command
                ...
                command;;
    patternN)   command
                ...
                command;;
    *)          command
                ...
                command;;
esac
```

The $variable-name is compared against the patterns until a match is found. The shell then executes all the statements up to the two semicolons that are next to each other. The default is *) and its executed if no match is found. For e.g. write script as follows:

Looping statements

Loops in Shell Scripts

Loop defined as:

"Computer can repeat particular instruction again and again, until particular condition satisfies. A group of instruction that is executed repeatedly is called a loop."

for loop

while loop

untill loop

Note that in each and every loop,

(a) First, the variable used in loop condition must be initialized, then execution of the loop begins.

(b) A test (condition) is made at the beginning of each iteration.

(c) The body of loop ends with a statement that modifies the value of the test (condition) variable.

TATA CONSULTANCY SERVICES

TATA

Any Queries

Experience certainty.

IT Services
Business Solutions
Outsourcing

TATA CONSULTANCY SERVICES

TATA

Thanks!!!

Happy Learning!!!

Experience certainty.

IT Services
Business Solutions
Outsourcing