Unix Shell Programming
Unit-2

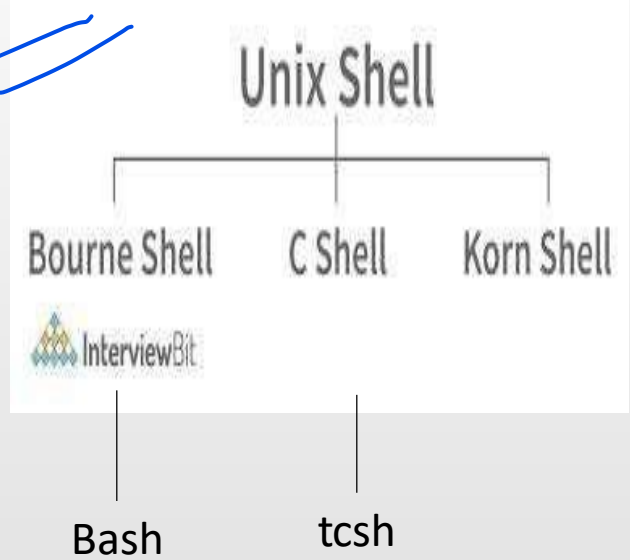
Ability Enhancement Course

AEC310

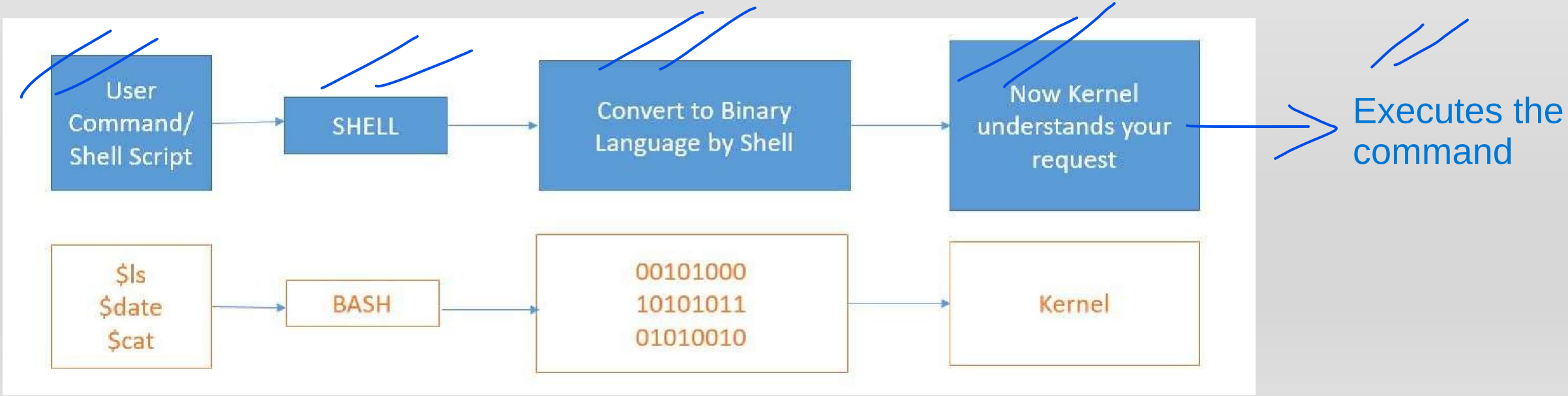
Introduction to Shell

- UNIX Session
- UNIX utilities –
  - process utilities
  - disk utilities
  - networking commands
  - Text processing utilities
  - backup utilities

Unix Shell

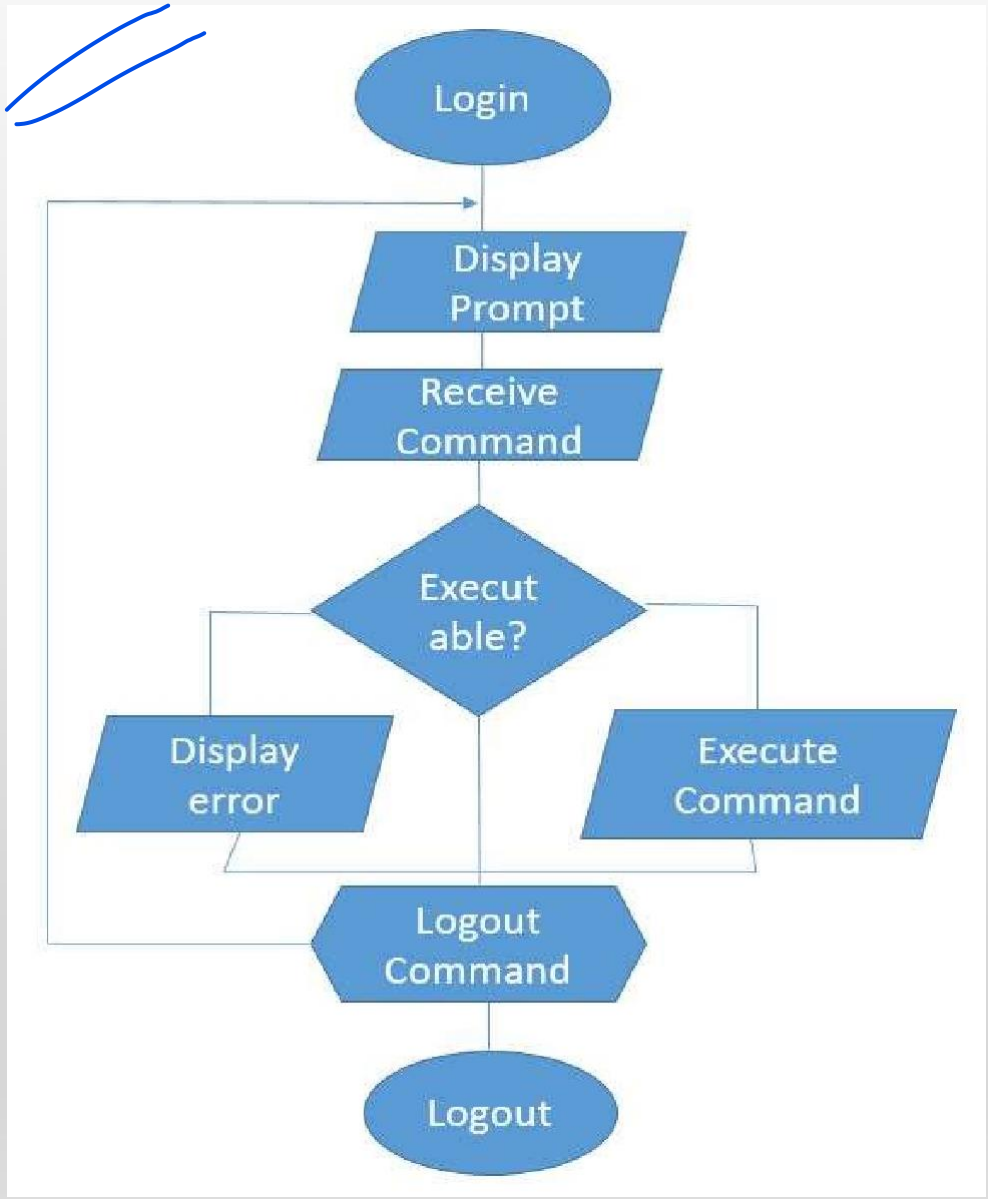


- A **Unix shell** is the part of UNIX that is most visible to the user. It receives and interprets the command entered by the user. If the command requires a utility or application program, the shell requests the kernel to execute it.
- There are major two parts to a shell. 1. Interpreter
  2. Programming Capability
- The interpreter reads your command and works with the kernel to execute them.

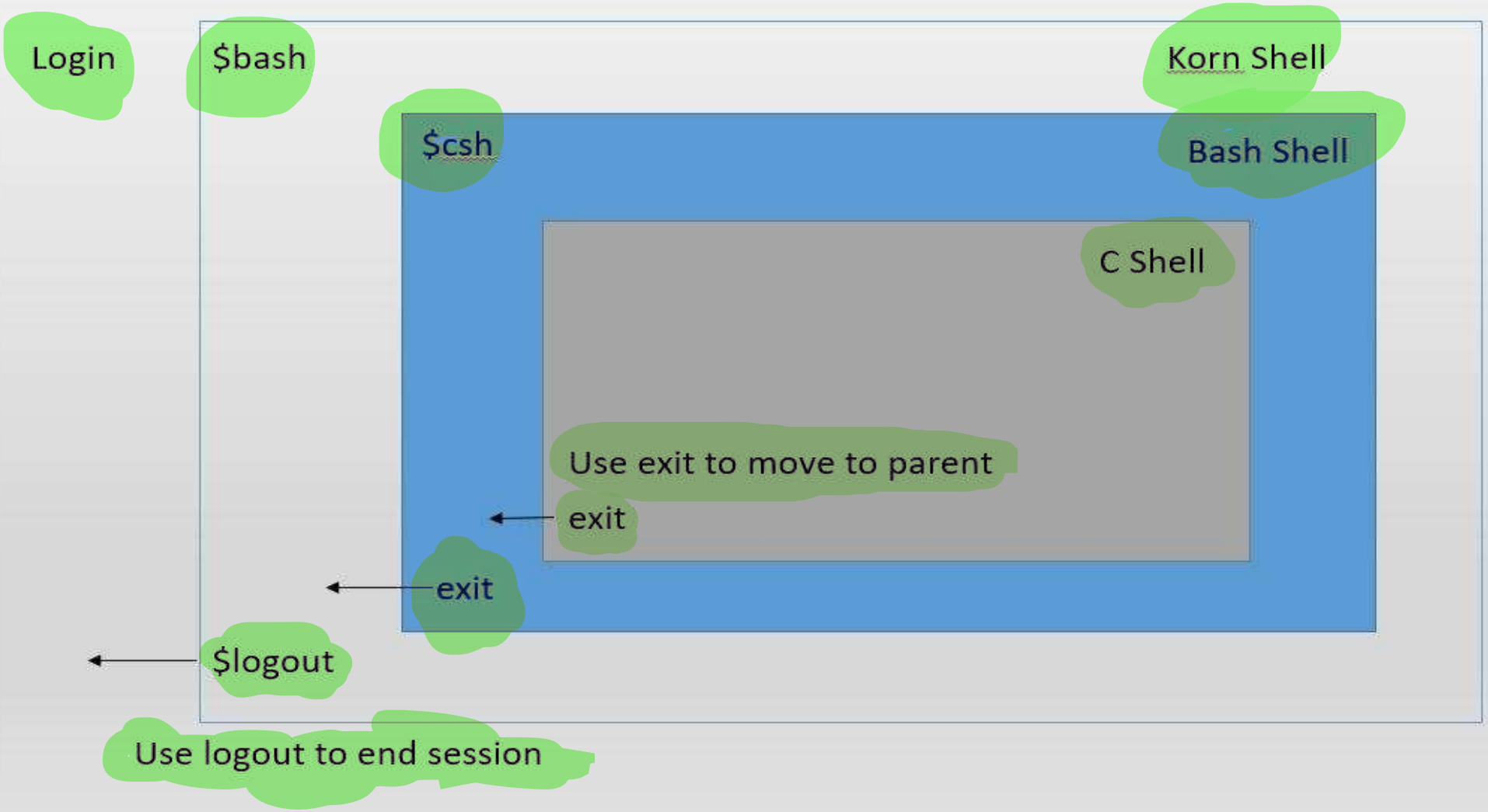


UNIX Session

- Unix session consists of logging into the system and then executing commands to accomplish our work. After the completion of work done, logout of the system.
- When logged in, the user will be in one of five shells. The system administrator determines which shell you start in by an entry in the password file(/etc/passwd).
- $echo $SHELL---Login Shell
- $echo $0-----Prints current shell
- But user can switch to different shell by typing the command:
  - Bash
  - ksh
  - csh



Shell relationship



Unix utilities

- It is a standard UNIX program that provides a support process for users.

TYPES.....

1. Process
2. Disk
3. Network
4. Text
5. Back-Up

Process utilities

- A program/command when executed, a special instance is provided by the system to the process. This instance consists of all the services/resources that may be utilized by the process under execution.
- Whenever a command is issued in Unix/Linux, it creates/starts a new process. For example, pwd when issued which is used to list the current directory location the user is in, a process starts.
- Through a 5 digit ID number Unix/Linux keeps an account of the processes, this number is call process ID or PID. Each process in the system has a unique PID.
- Used up pid's can be used in again for a newer process since all the possible combinations are used.
- At any point of time, no two processes with the same pid exist in the system because it is the pid that Unix uses to track each process.

Initializing a process

- A process can be run in two ways:
  - Method 1: Foreground Process
  - Method 2: Background Process

Method 1: Foreground Process

Every process when started runs in foreground by default, receives input from the keyboard, and sends output to the screen. When issuing pwd command

$ ls pwd

When a command/process is running in the foreground and is taking a lot of time, no other processes can be run or started because the prompt would not be available until the program finishes processing and comes out.

Method 2: Background Process: It runs in the background without keyboard input and waits till keyboard input is required. Thus, other processes can be done in parallel with the process running in the background since they do not have to wait for the previous process to be completed. Adding & along with the command starts it as a background process

$ pwd &

ps (Process status) can be used to see/list all the running processes.

- $ps
- **UID**: User ID that this process belongs to (the person running it)
- **PID**: Process ID
- **PPID**: Parent process ID (the ID of the process that started it)
- **C**: CPU utilization of process
- **STIME**: Process start time
- **TTY**: Terminal type associated with the process
- **TIME**: CPU time is taken by the process
- **CMD**: The command that started this process
  - **-a**: Shows information about all users
  - **-x**: Shows information about processes without terminals
  - **-u**: Shows additional information like -f option
  - **-e**: Displays extended information

- **Stopping a process:**

When running in foreground, hitting Ctrl + c (interrupt character) will exit the command. For processes running in background kill command can be used if it's pid is known.

```
$ ps -f

UID        PID  PPID  C STIME    TTY          TIME CMD
52471       19     1  0 07:20    pts/1    00:00:00       sh
52471       25    19  0 08:04    pts/1    00:00:00       ps -f

$ kill 19
Terminated
```

If a process ignores a regular kill command, you can use kill -9 followed by the process ID.

```
$ kill -9 19
Terminated
```

- bg: A job control command that resumes suspended jobs while keeping them running in the background

Syntax: bg [job]

- fg: It continues a stopped job by running it in the foreground.

Syntax: fg [job_id]

- df: It shows the amount of available disk space being used by file systems

Syntax: df

```
Filesystem      1K-blocks       Used Available Use% Mounted on
/dev/loop0       18761008   15246876   2554440  86% /
none                    4          0         4   0% /sys/fs/cgroup
udev               493812          4    493808   1% /dev
tmpfs              100672       1364     99308   2% /run
none                 5120          0      5120   0% /run/lock
none               503352       1764    501588   1% /run/shm
none               102400         20    102380   1% /run/user
/dev/sda3       174766076  164417964  10348112  95% /host
```

free: It shows the total amount of free and used physical and swap memory in the system, as well as the buffers used by the kernel

Syntax:

```
free
```

Output:

```
             total       used       free     shared    buffers     cached
Mem:       1006708     935872      70836          0     148244     346656
-/+ buffers/cache:     440972     565736
Swap:       262140     130084     132056
```

Types of Process

- **Parent and Child process :** The 2nd and 3rd column of the ps –f command shows process id and parent's process id number. For each user process, there's a parent process in the system, with most of the commands having shell as their parent.

- **Zombie and Orphan process :** After completing its execution a child process is terminated or killed and SIGCHLD updates the parent process about the termination and thus can continue the task assigned to it. But at times when the parent process is killed before the termination of the child process, the child processes become orphan processes, with the parent of all processes "init" process, becomes their new pid. A process which is killed but still shows its entry in the process status or the process table is called a zombie process, they are dead and are not used.

- **Daemon process :** They are system-related background processes that often run with the permissions of root and services requests from other processes, they most of the time run in the background and wait for processes it can work along with.

Disk utilities

- While working on UNIX, there might come a situation when you want to transfer a set of files or the entire directory. In such a case, you might want to know the disk space consumed by that particular directory or set of files. There exists a command line utility for this also which is **du** command that estimates and displays the **disk space** used by files.

- So, in simple words **du** command-line utility helps to find out the disk usage of set of files or a directory.

du [OPTION]... [FILE]...

where **OPTION** refers to the options compatible with du command and **FILE** refers to the filename of which you want to know the disk space occupied.

Examples:

- **Using -h option :** As mentioned above, -h option is used to produce the output in human readable format.

- **Using du to show disk usage of a directory**

```
/*using du to display disk usage
of a directory and its
sub-directories */

$du kartik
4       kartik/thakral
24      kartik
```

- **Using -a option:** This option produces counts as output for all files with the disk usage space.
- **Using – -time option :** This option is used to display the last modification time in the output of du.

Network Utilities

- Variety of network tools are used to perform tasks such as obtaining information about other systems on your network, accessing other systems, and communicating directly with other users.

- Network information can be obtained using utilities such as **ping, finger, traceroute, host, dig, nslookup etc**. These are useful for smaller networks and enable to access remote systems directly to copy files or execute the command.
- ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol network.

ping Domain_name/IP Address

- host

This command is used to obtain network address information about a remote system connected to your network. This information usually consists of system's IP address, domain name address and sometimes mail server also.

Syntax:

$ host www.google.com

- **Traceroute** : This command is used to track the sequence of computer networks. You can track to check the route through which you are connected to a host.

- **Netstat :** This command is used to check the status of ports whether they are open, closed, waiting, and masquerade connections. Network Statistic (netstat) command displays connection information, routing table information, etc.

  Syntax: $ netstat

- **Tracepath :** tracepath performs a very similar function to that of traceroute command. The main difference between this command is that tracepath doesn't take complicated options. This command doesn't require root privileges.

  Syntax: $ tracepath www.google.com

- **Hostname:** This command is used to see the hostname of your computer. You can change hostname permanently in etc/sysconfig/network. After changing the hostname you need to reboot the computer.

  Syntax: $ hostname

- **Route :** The route command is used to display or modify the routing table. To add a gateway use (-n).

  Syntax: $ route -n

- **Nslookup** :You can use nslookup (name server lookup) command to find out DNS related queries or testing and troubleshoot DNS server.

  Syntax: $ nslookup google.com

Text Processing Utilities

- UNIX provides rich set of text processing commands which are comprehensive and time saving.
- **Sort :** sort lines of text files

  sort file.txt : Sorts the file alphabetically.

  sort -n file.txt : Sorts the file numerically

  sort -r file.txt : Sorts the file in the reverse order. Thus, b comes before a.

  sort -u file.txt : Sorts the file, removing duplicate lines, thereby ensuring each output line is unique.

- **Uniq**: report/omit the repeated lines

This command is more specific to recognizing duplicates. Usually requires a sorted input as the comparison is made on adjacent lines only

-d : Print only duplicate lines

-c: Prefix count to occurrences

-u : print only unique lines

- **Comm:** Outputs lines common to two files or unique to them, provided the files are sorted. If the files are not sorted, the output is indeterminate. Options control the manner of identification, e.g. outputting only common lines.

- Without any options, it prints output in three columns - lines unique to file1, line unique to file2

  and lines common to both files

-1: suppress lines unique to file1

-2: suppress lines unique to file2

-3 : suppress lines common to both files

- **Cmp :** compare two files byte by byte

Useful to compare binary files. If the two files are same, no output is displayed (exit status 0).

If there is a difference, it prints the first difference - line number and byte location (exit status 1)

- **Diff :** compare files line by line

Useful to compare old and new versions of text files
All the differences are printed, which might not be desirable if files are too long

Options:

-s : convey message when two files are same

-y: two column output

-i: ignore case while comparing

-w: ignore white-spaces

Tr: Translate or delete characters:

-d: delete the specified characters

tr a-z A-Z < test_list.txt : convert lowercase to uppercase

tr -d ._ < test_list.txt : delete the dot and underscore characters

Back- up Utilities

- Most Unix systems come with several basic backup software options, including dd, cpio, tar, and dump.
- Tar

tar stands for “tape archive.” tar was originally written to backup data to tape, but is in fact a general purpose library program. A library program is a program which copies a large number of files into one larger file for simplified data management. Those files can later be extracted from the library for use.

For example, the tar command to backup your filesystem to a tape unit named /dev/st0 would be:

- tar -cvf /dev/st0 /

- **Cpio** : cpio stands for “copy input output.” cpio simply copies its input to its output. cpio is not capable of finding files on its own, so it is usually used in combination with the find command.

For example, the cpio command to backup your filesystem to a tape unit named /dev/st0 would be:

```
find / -print |cpio -ocBv /dev/st0
```

- **Dd** : dd is a very low-level command for copying data. dd is most frequently used for making exact copies of phyiscal hard drives for computer forensics work.

For example, the dd command to backup a hard drive named /dev/hda to another hard drive named /dev/hdb would be:

```
dd if=/dev/hda of=/dev/hdb
```

- dump / rdump / ufsdump

dump is a newer Unix command than dd, tar, or cpio. dump is designed for simplified backups of an entire filesystem.

For example, the dump command to backup your filesystem to a tape unit named /dev/nst0 would be:

```
dump -0ua -f /dev/nst0 /
```

- rdump stands for “remote dump.” rdump is used to backup a filesystem to a remote host.

Under Solaris, dump is called “ufsdump.”