Linux Lecture-2

Topics

- Interacting with Linux
- Shells
- File System
- Directories
- Commands

Interacting with Linux

GUI

- KDE
- GNOME
- XFCE
- LXDE
- MATE

Terminal (Command Line)

- Also called a shell
- Just like Windows MS DOS Linux also have a command line through which it can be operated.
- The Linux terminal is a text-based interface used to control a Linux computer. It's just one of the many tools provided to Linux users for accomplishing any given task, but it's widely considered the most efficient method available. Outside of writing code, it's certainly the most direct method possible.

Shell

- The Shell is a Command Line Interpreter. It translates commands entered by the user and converts them into the language that is understood by the Kernel.
- A Linux shell is an interactive program that accepts commands from user via key board, parse them from left to right and execute them. Most of the shells available in todays Linux provides the features of executing user commands and programs, I/O handling, programming ability (scripts and binaries). Example shells are Bourne shell, Bourne Again Shell, C Shell, Korn Shell.
- Shell script is a list of commands which are listed in the order of execution.

Bourne Shell

- 1977
- Created by Stephen Bourne.
- Bourne shell is useful even today and in some cases as the default root shell.
- Its grammar is similar to Algorithm Language (ALGOL)
- The Bourne shell had two primary goals: command interpreter and scripting.

C Shell

- 1978
- The C shell was developed by Bill Joy
- Objective was to create a scripting language similar to C programming language.
- This was useful given that C was a primary language in use back then which also made it easier and faster to use.

Korn Shell

- 1983
- Developed by David Korn.
- The Korn shell combined features of both Bourne and C shells.
- It includes features from C shell such as job control, command aliasing and command history.

TENEX C Shell

- 1983
- Started out as a derivative of the C shell but with a programmable command line completion and editing features added to it.

Bourne Again Shell (Bash)

- 1989
- One of the most widely used shell today.
- It was written be Brian Fox for the GNU project as a pre software replacement for the Bourne Shell.
- Shows all features from the Bourne shell but is more efficient and easy to use.
- It supports filename globing, piping, command substitution and control structures for conditional testing and iteration.

Other Shells

- Many shells were evolved later such as Public Domain Korn Shell, Almquist Shell and Extensible Shell bringing in new features and dialects of their own suitable for different needs.
- 1990 zsh
- 1992 POSIX
- 1993 es
- 1994 scsh
- 1996 dash
- 1999 psh
- 2003 mksh

File System

- At its base form a file system is just a way to organize your drive.
- It determines the structure in which data is stored and retrieved.
- Without this structure it would be very difficult to tell where one file ends and the other begins.
- FAT 32, NTFS and Ext are some examples of File system.

File System

- **FAT32:** is a simple file system that is supported for reading and writes on all major operating systems. It has no security and does not perform well with large files.
- **NTFS:** makes improvements on FAT with security and in many cases contiguous reads, but it still suffers some similar aliments.
- **Ext:** is generally a good choice for working with most files, however small files would benefit more from contiguous allocation.

Contiguous and Noncontiguous Memory Allocation

- **Contiguous Memory Allocation :** Contiguous memory allocation is basically a method in which a single contiguous section/part of memory is allocated to a process or file needing it. Because of this all the available memory space resides at the same place together, which means that the freely/unused available memory partitions are not distributed in a random fashion here and there across the whole memory space.
- **Non-Contiguous** memory allocation: is basically a method on the contrary to contiguous allocation method, allocates the memory space present in different locations to the process as per it’s requirements. As all the available memory space is in a distributed pattern so the freely available memory space is also scattered here and there.

Contiguous and Noncontiguous Memory Allocation

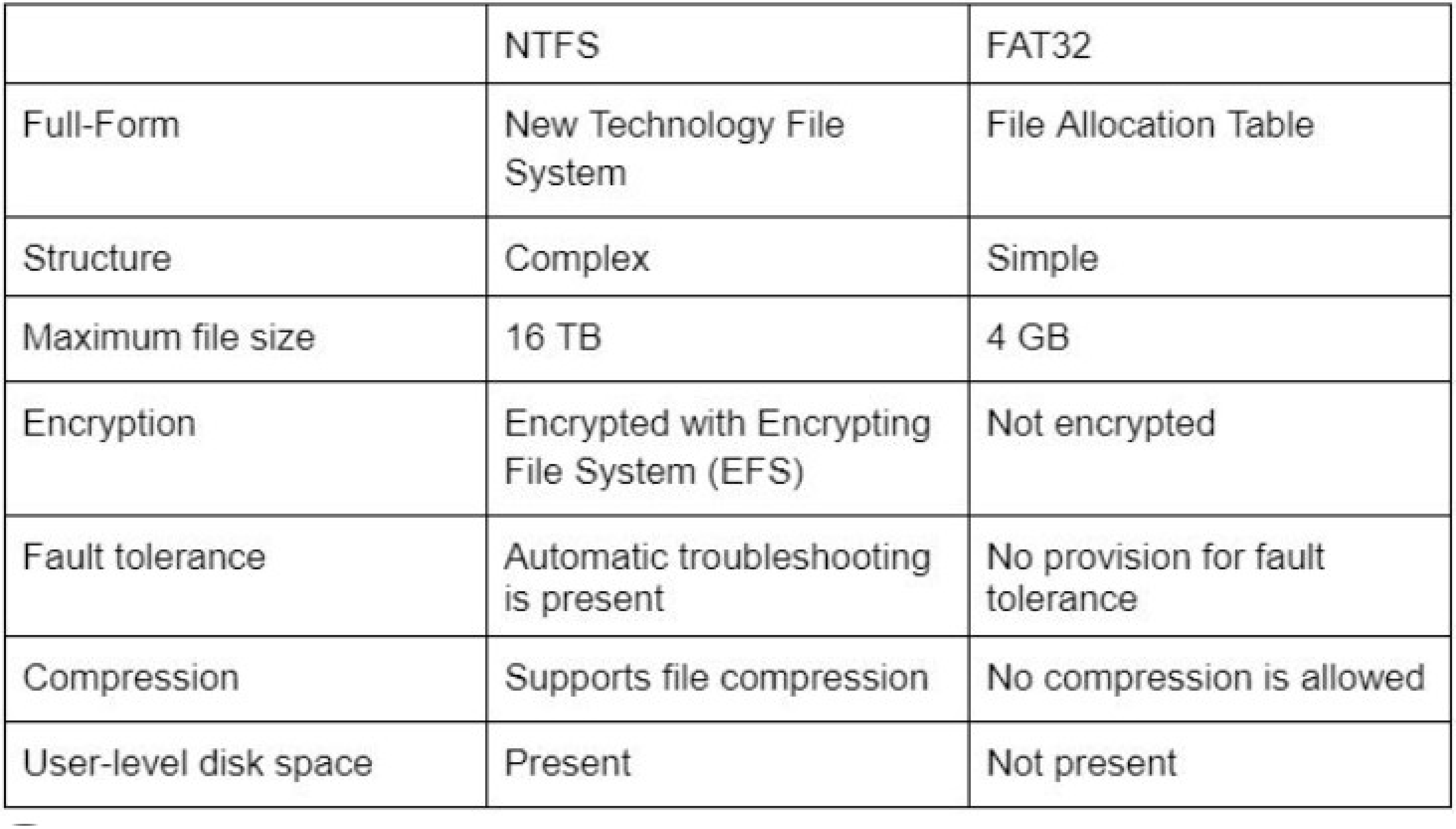
Contiguous

- Overhead is minimum as not much address translations are there while executing a process.
- Faster in Execution
- It is easier for the OS to control.
- Wastage of memory is there.
- In contiguous memory allocation, swapped-in processes are arranged in the originally allocated space.

Noncontiguous

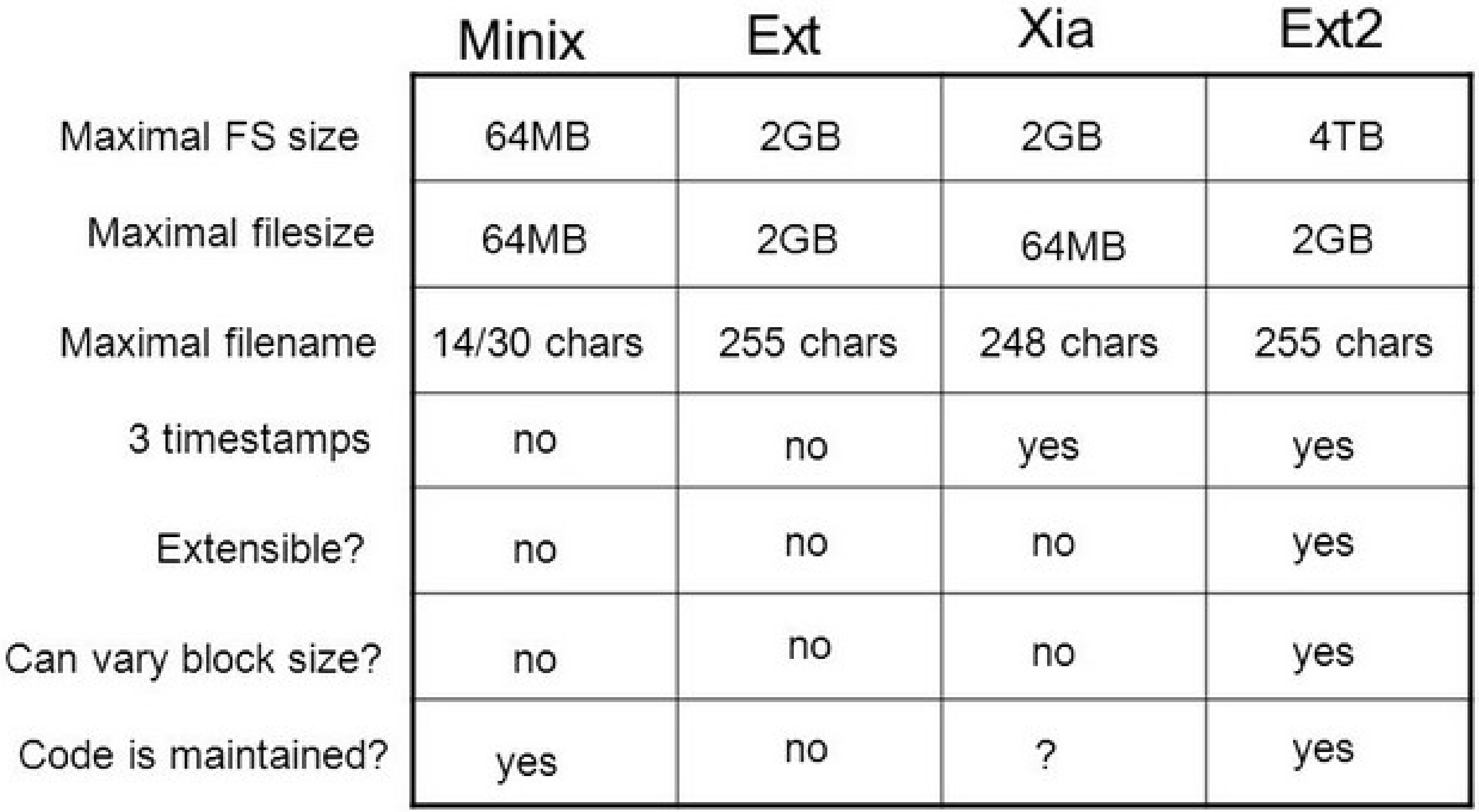
- More Overheads are there as there are more address translations.
- Slower in Execution
- It is difficult for the OS to control.
- No memory wastage is there
- In non-contiguous memory allocation, swapped-in processes can be arranged in any place in the memory

Brief Comparison



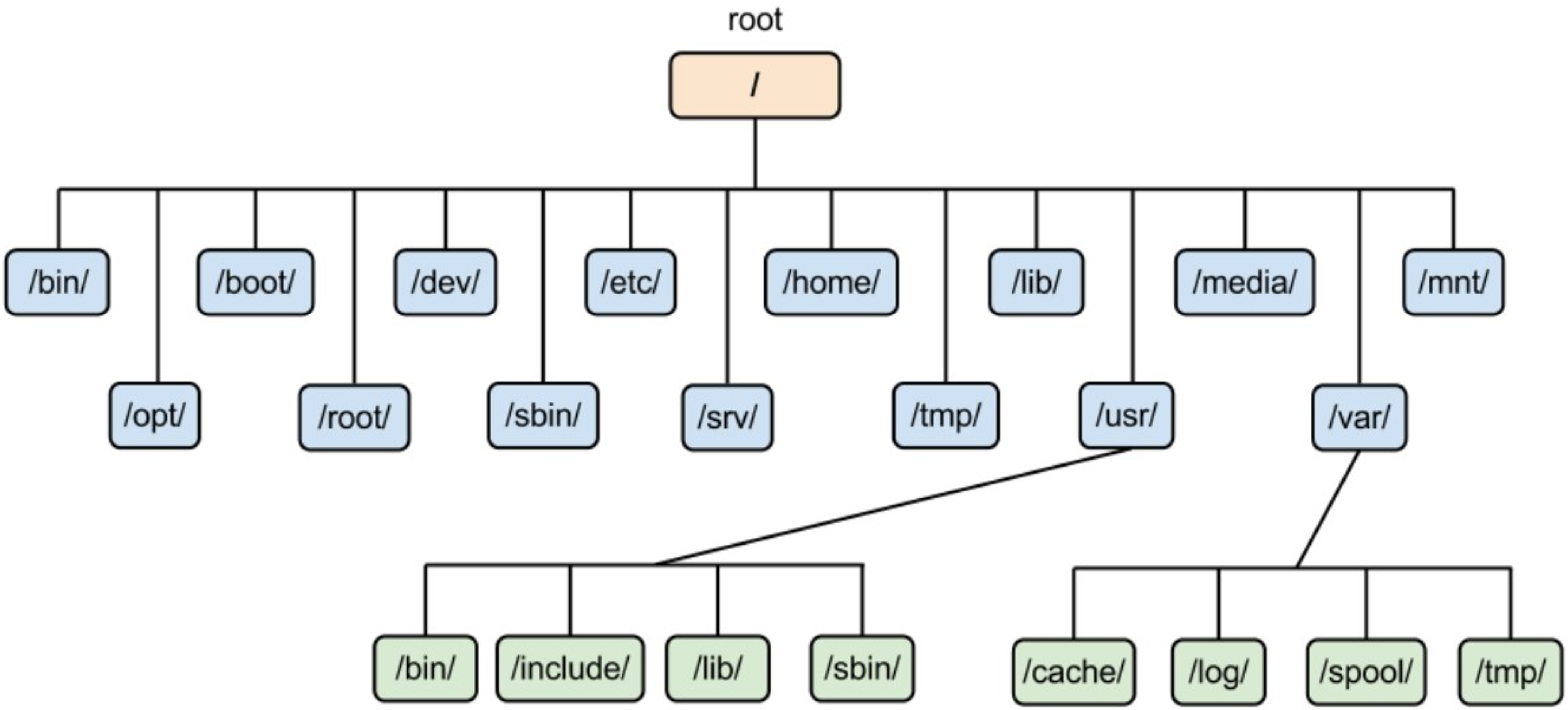
	NTFS	FAT32
Full-Form	New Technology File System	File Allocation Table
Structure	Complex	Simple
Maximum file size	16 TB	4 GB
Encryption	Encrypted with Encrypting File System (EFS)	Not encrypted
Fault tolerance	Automatic troubleshooting is present	No provision for fault tolerance
Compression	Supports file compression	No compression is allowed
User-level disk space	Present	Not present

Brief Comparison

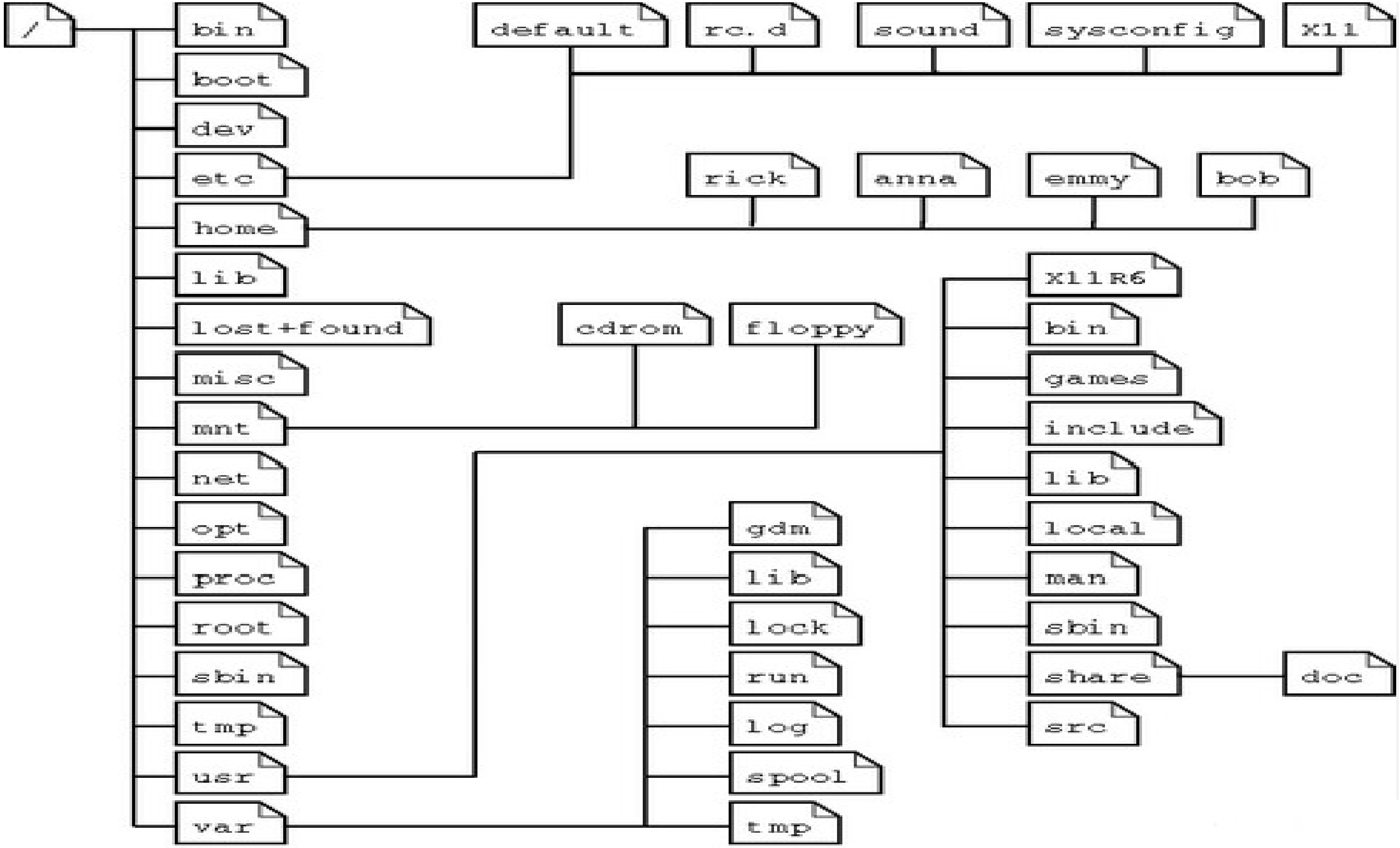


	Minix	Ext	Xia	Ext2
Maximal FS size	64MB	2GB	2GB	4TB
Maximal filesize	64MB	2GB	64MB	2GB
Maximal filename	14/30 chars	255 chars	248 chars	255 chars
3 timestamps	no	no	yes	yes
Extensible?	no	no	no	yes
Can vary block size?	no	no	no	yes
Code is maintained?	yes	no	?	yes

Linux File System



Linux File System



Linux Directory

- The Linux file system is partitioned into separate directories, denoted by forward-slash character (/).
  – Tree structure starting with base (/) root directory.
- Current directory you are in is called the current working directory.
- Use the pwd command to print the current working directory and the ls command to list the contents of the directory.

Directories

These are the common **top-level directories** associated with the **root** directory:

- /bin – binary or executable programs.
- /etc – system configuration files.
- /home – home directory. It is the default current directory.
- /opt – optional or third-party software.
- /tmp – temporary space, typically cleared on reboot.
- /usr – User related programs.
- /var – log files.

Directories

Some other directories in the Linux system:

- /boot- It contains all the boot-related information files and folders such as conf, grub, etc.
- /dev – It is the location of the device files such as dev/sda1, dev/sda2, etc.
- /lib – It contains kernel modules and a shared library.
- /lost+found – It is used to find recovered bits of corrupted files.
- /media – It contains subdirectories where removal media devices inserted.
- /mnt – It contains temporary mount directories for mounting the file system.
- /proc – It is a virtual and pseudo-file system to contains info about the running processes with a specific process ID or PID.
- /run – It stores volatile runtime data.
- /sbin – binary executable programs for an administrator.
- /srv – It contains server-specific and server-related files.
- /sys – It is a virtual filesystem for modern Linux distributions to store and allows modification of the devices connected to the system.

Directories

Linux Kernel File:

- /boot/vmlinux **–** The Linux kernel file.

Device Files:

- /dev/hda **–** Device file for the first IDE HDD.
- /dev/hdc **–** A pseudo-device that output garbage output is redirected to /dev/null.

Directories

System Configuration Files:

- /etc/bashrc – It is used by bash shell that contains system defaults and aliases.
- /etc/crontab – A shell script to run specified commands on a predefined time interval.
- /etc/exports – It contains information on the file system available on the network.
- /etc/fstab – Information of the Disk Drive and their mount point.
- /etc/group – It is a text file to define Information of Security Group.
- /etc/grub.conf – It is the grub bootloader configuration file.
- /etc/init.d – Service startup Script.
- /etc/lilo.conf – It contains lilo bootloader configuration file.
- /etc/hosts – Information of IP and corresponding hostnames.
- /etc/hosts.allow – It contains a list of hosts allowed accessing services on the local machine.
- /etc/host.deny – List of hosts denied to access services on the local machine.
- /etc/inittab – INIT process and their interaction at the various run level.
- /etc/issue – Allows editing the pre-login message.

Directories

System Configuration Files:

- /etc/modules.conf – It contains the configuration files for the system modules.
- /etc/motd – It contains the message of the day.
- /etc/mtab – Currently mounted blocks information.
- /etc/passwd – It contains username, password of the system, users in a shadow file.
- /etc/printcap – It contains printer Information.
- /etc/profile – Bash shell defaults.
- /etc/profile.d – It contains other scripts like application scripts, executed after login.
- /etc/rc.d – It avoids script duplication.
- /etc/rc.d/init.d – Run Level Initialisation Script.
- /etc/resolv.conf – DNS being used by System.
- /etc/security – It contains the name of terminals where root login is possible.
- /etc/skel – Script that initiates new user home directory.
- /etc/termcap – An ASCII file that defines the behavior of different types of the terminal.
- /etc/X11 – Directory tree contains all the conf files for the X-window System.

Directories

User Related Files:

- /usr/bin – It contains most of the executable files.
- /usr/bin/X11 – Symbolic link of /usr/bin.
- /usr/include – It contains standard include files used by C program.
- /usr/share – It contains architecture independent shareable text files.
- /usr/lib – It contains object files and libraries.
- /usr/sbin – It contains commands for Super User, for System Administration.

Directories

Virtual and Pseudo Process Related Files:

- /proc/cpuinfo – CPU Information
- /proc/filesystems – It keeps the useful info about the processes that are running currently.
- /proc/interrupts – it keeps the information about the number of interrupts per IRQ.
- /proc/ioports – Contains all the Input and Output addresses used by devices on the server.
- /proc/meminfo – It reports the memory usage information.
- /proc/modules – Currently using kernel module.
- /proc/mount – Mounted File-system Information.
- /proc/stat – It displays the detailed statistics of the current system.
- /proc/swaps – It contains swap file information.

Directories

Version Information File:

- /version **–** It displays the Linux version information.

Log Files:

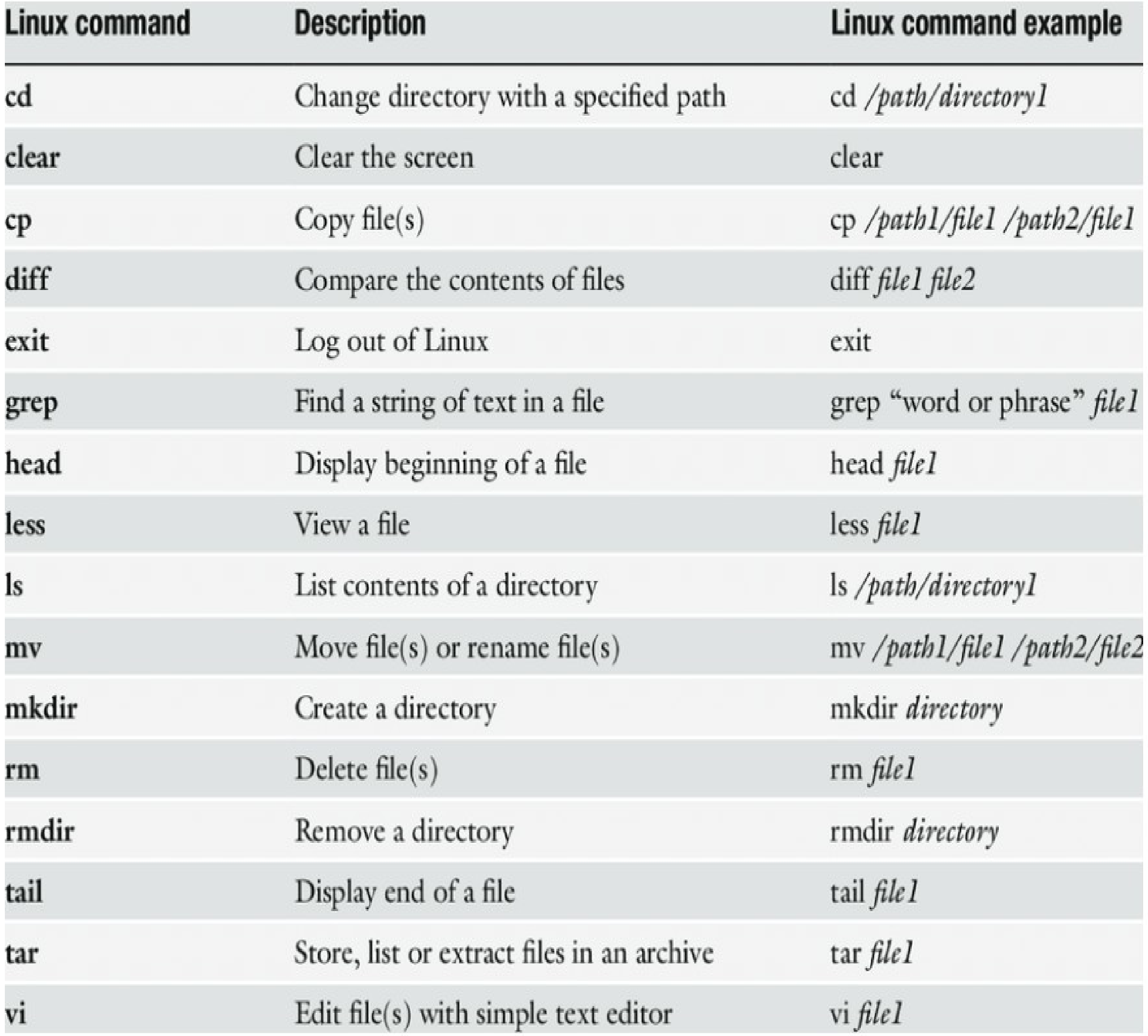
- /var/log/lastlog – It stores user last login info.
- /var/log/messages – It has all the global system messages.
- /var/log/wtmp – It keeps a history of login and logout information.

Linux File Names

- File names on Linux are case sensitive.
  – So are commands because these are just executable files!
- Linux file names don’t have dot extensions like Windows.
- A file name starting with a period (.) is called a hidden file and isn’t displayed in a standard directory listing.

Linux Shell Commands

- A shell command can be internal/built-in or External
- The code to execute an internal command is part of the shell process, e.g., cd, dot, echo, pwd.
- The code to process an external command resides in a file in the form of a binary executable program file or a shell script, e.g., cat, ls, mkdir, more.
- The general syntax of a shell command is **command [option(s)] [argument(s)]**
- After reading the command the shell determines whether the command is internal or external
- It processes all internal commands by using the corresponding code segments that are within its own code
- To execute an external command, it searches the command in the search path. Directories names stored in the PATH variable. [echo $PATH]



Linux command	Description	Linux command example
cd	Change directory with a specified path	cd */path/directory1*
clear	Clear the screen	clear
cp	Copy file(s)	cp */path1/file1 /path2/file1*
diff	Compare the contents of files	diff *file1 file2*
exit	Log out of Linux	exit
grep	Find a string of text in a file	grep “word or phrase” *file1*
head	Display beginning of a file	head *file1*
less	View a file	less *file1*
ls	List contents of a directory	ls */path/directory1*
mv	Move file(s) or rename file(s)	mv */path1/file1 /path2/file2*
mkdir	Create a directory	mkdir *directory*
rm	Delete file(s)	rm *file1*
rmdir	Remove a directory	rmdir *directory*
tail	Display end of a file	tail *file1*
tar	Store, list or extract files in an archive	tar *file1*
vi	Edit file(s) with simple text editor	vi *file1*

Changing Directory

- The cd command is used to change the current working directory to a new one.
- Two options:
- Specify an absolute path name:

```
$ cd /home/bob/documents
```

- Specify a relative path name:

```
$ cd documents
```

change to documents sub-directory located in present directory

```
$ cd ../databases
```

change to databases sub-directory located in parent directory

```
$ cd ~
```

change to user’s home directory

Managing Directory

- Use the mkdir command to create a new directory.
  - $ mkdir test
    - Creates a sub-directory called test in the present directory.
  - $ mkdir /home/bob/test
    - Creates a sub-directory called test in the /home/bob directory.
- Use the rmdir command to remove a directory.
  - $ rmdir test

Creating Files

- Easy way to create an empty text file is to use the touch command.
  - $ touch myfile.txt
- You can also use one of the basic text editors to create a file.
  - $ nano myfile.txt (recommended for beginners)
  - $ vi myfile.txt

Viewing Files

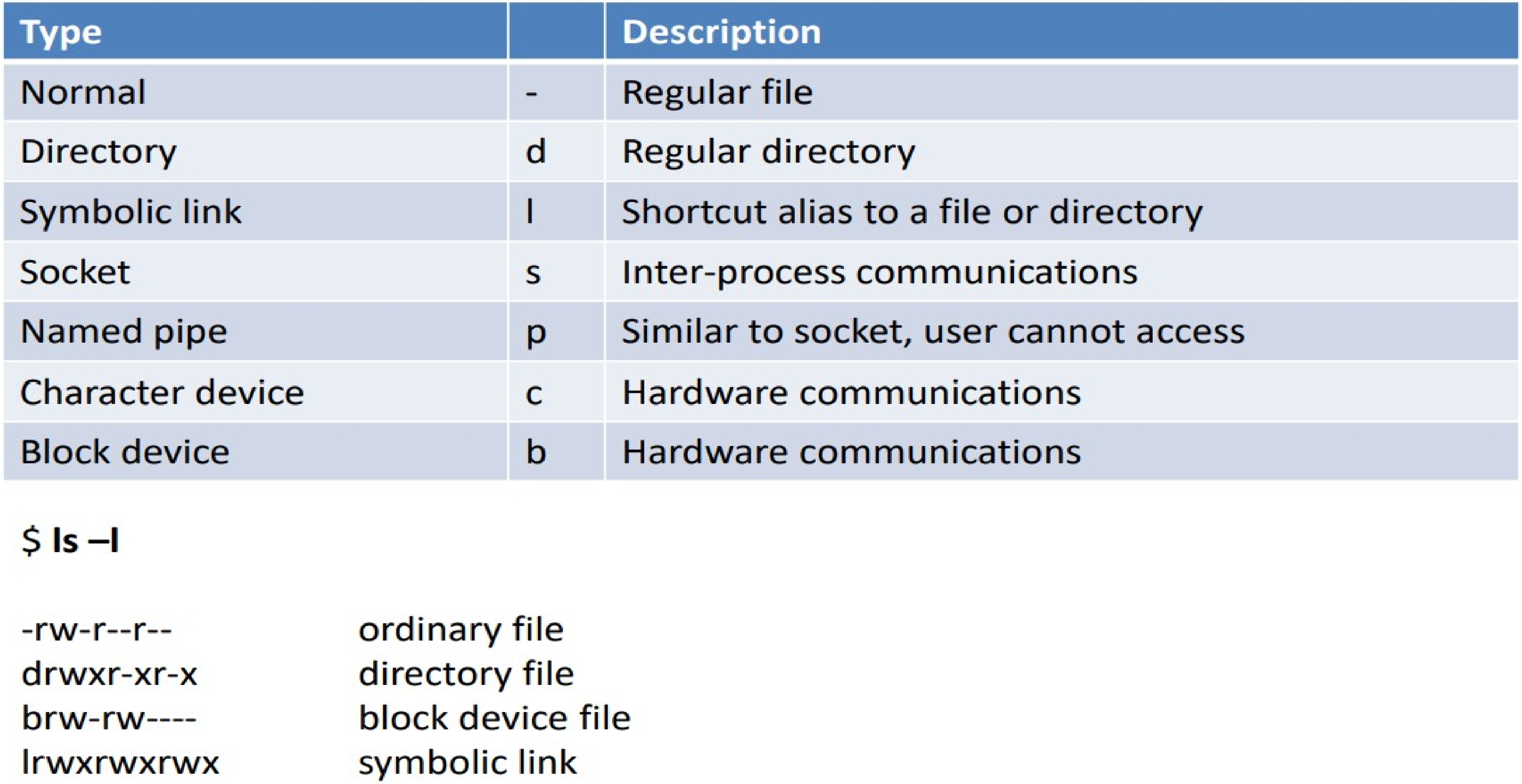
- Linux provides a couple different tools to view the contents of text files.
  - $ cat myfile.txt
  - $ more myfile.txt (similar to cat but with pagination)
  - $ less myfile.txt (less is more than more)

Listing Files

- The ls command allows you to list the files in a directory.

  - $ ls
  - $ ls /home/ubuntu-user
  - $ ls ~
- Add the –la option to the command to see more file details.
  - $ ls –la
  - $ ls –la /home/ubuntu-user

Linux File Types



Type		Description
Normal	-	Regular file
Directory	d	Regular directory
Symbolic link	l	Shortcut alias to a file or directory
Socket	s	Inter-process communications
Named pipe	p	Similar to socket, user cannot access
Character device	c	Hardware communications
Block device	b	Hardware communications

$ ls –l

-rw-r--r--	ordinary file
drwxr-xr-x	directory file
brw-rw----	block device file
lrwxrwxrwx	symbolic link

Managing Files

- Copy the file to another file location using the copy command.
  - $ cp myfile.txt myfile.bak
- Move the file (rename) using the mv command.
  - $ mv myfile.txt myfile2.txt
- Delete a file using the rm command.
  - $ rm myfile2.txt

Linux Command Help

- Linux provides a couple different methods to get helpful information about a command:

  ```
  $ man <command>
  ```

  ```
  $ <command> --help
  ```

- Sometimes the best way to get help is just using a web search engine.

Command Pipelines

- A command pipe allows you to take the stdout (output) of a command and send it to the stdin (input) of another command.
- A pipe is denoted using the vertical bar character (|).

  - $ ls /var/www/html | sort

- List the files /var/www/html directory and output the listing in alphabetical order

Shutdown and Restart

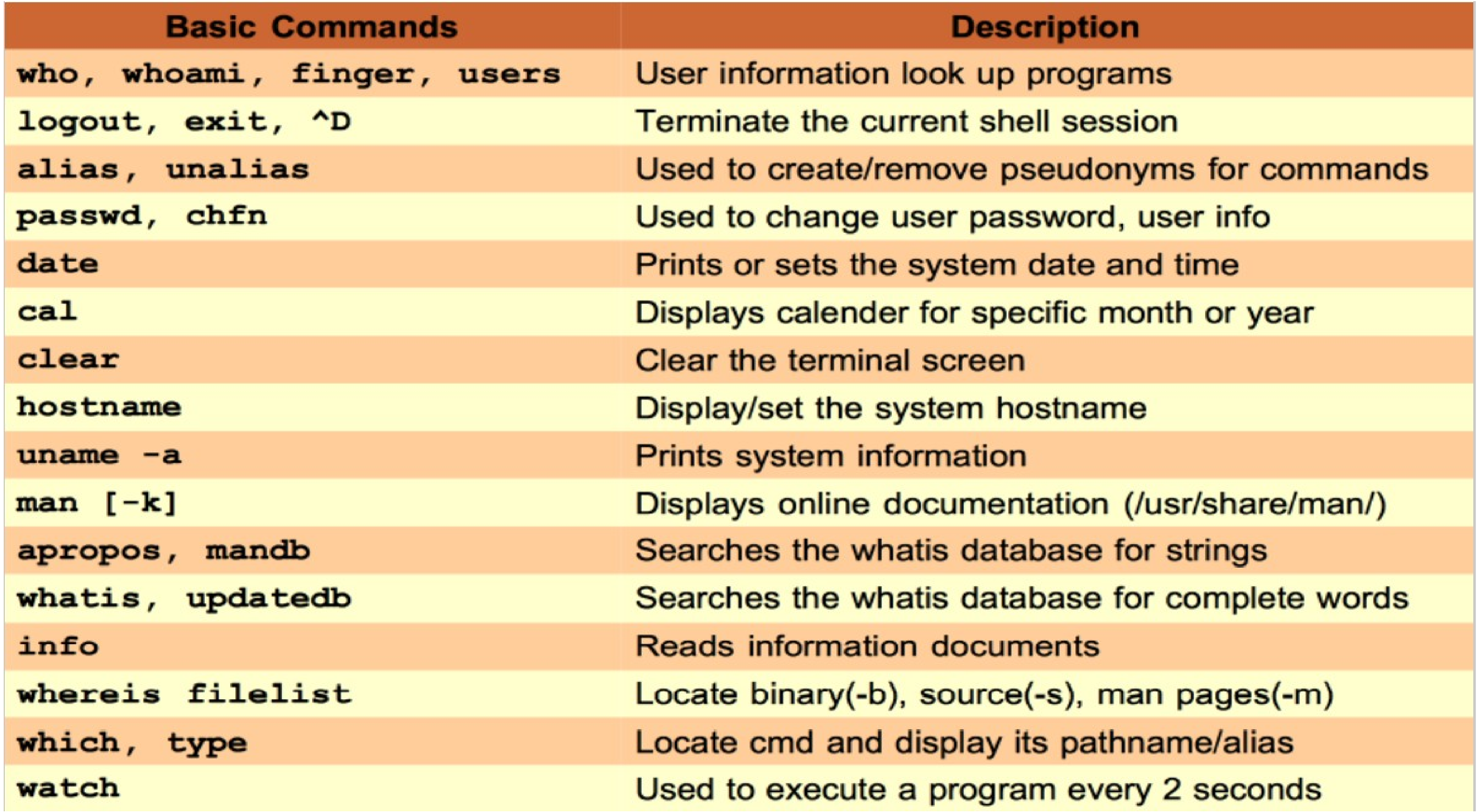
- The shutdown command is used to shutdown a system (requires super-user privileges).

– $ sudo shutdown

- To tell the system to automatically restart after shutting down:

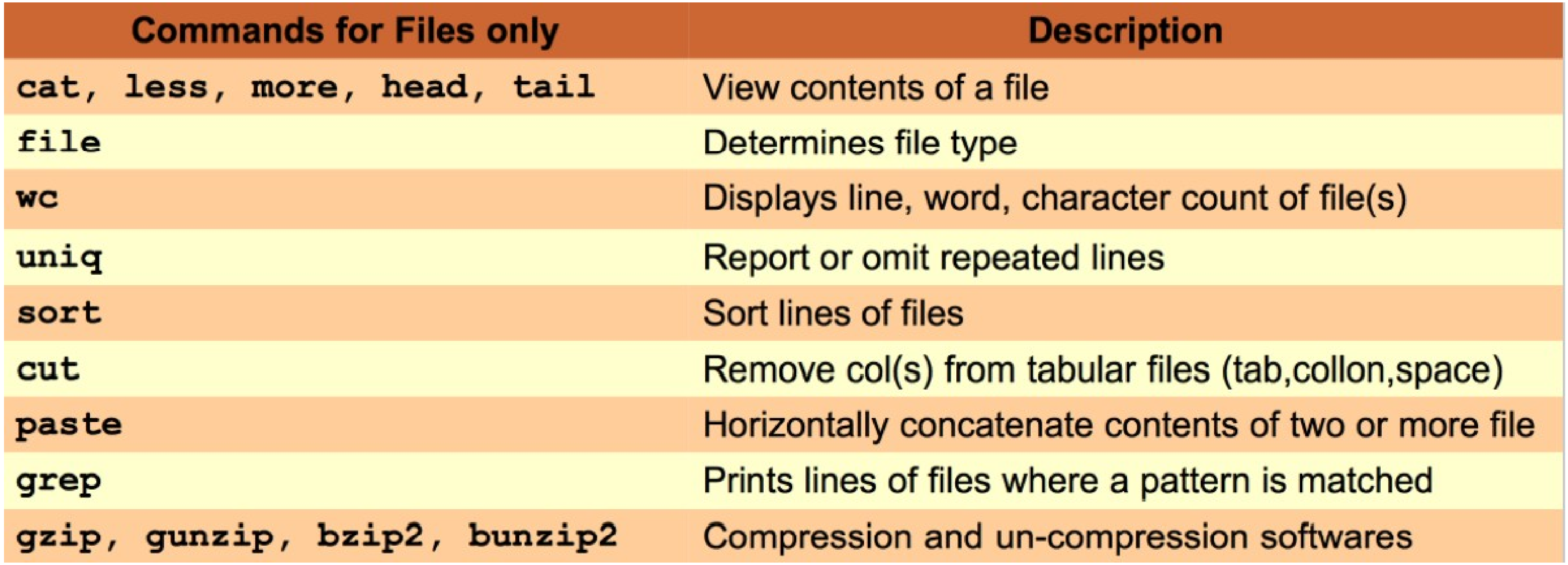
– $ sudo shutdown –r now

Basic Commands

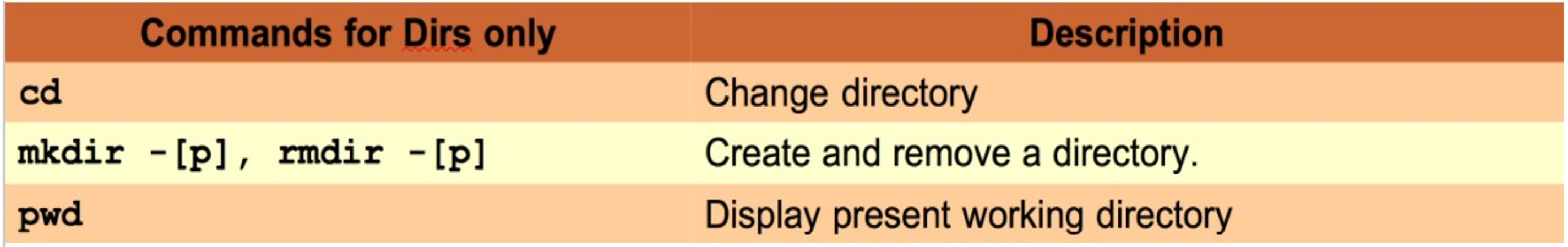


Basic Commands	Description
`who, whoami, finger, users`	User information look up programs
`logout, exit, ^D`	Terminate the current shell session
`alias, unalias`	Used to create/remove pseudonyms for commands
`passwd, chfn`	Used to change user password, user info
`date`	Prints or sets the system date and time
`cal`	Displays calender for specific month or year
`clear`	Clear the terminal screen
`hostname`	Display/set the system hostname
`uname -a`	Prints system information
`man [-k]`	Displays online documentation (/usr/share/man/)
`apropos, mandb`	Searches the whatis database for strings
`whatis, updatedb`	Searches the whatis database for complete words
`info`	Reads information documents
`whereis filelist`	Locate binary(-b), source(-s), man pages(-m)
`which, type`	Locate cmd and display its pathname/alias
`watch`	Used to execute a program every 2 seconds

Files and Directories Commands

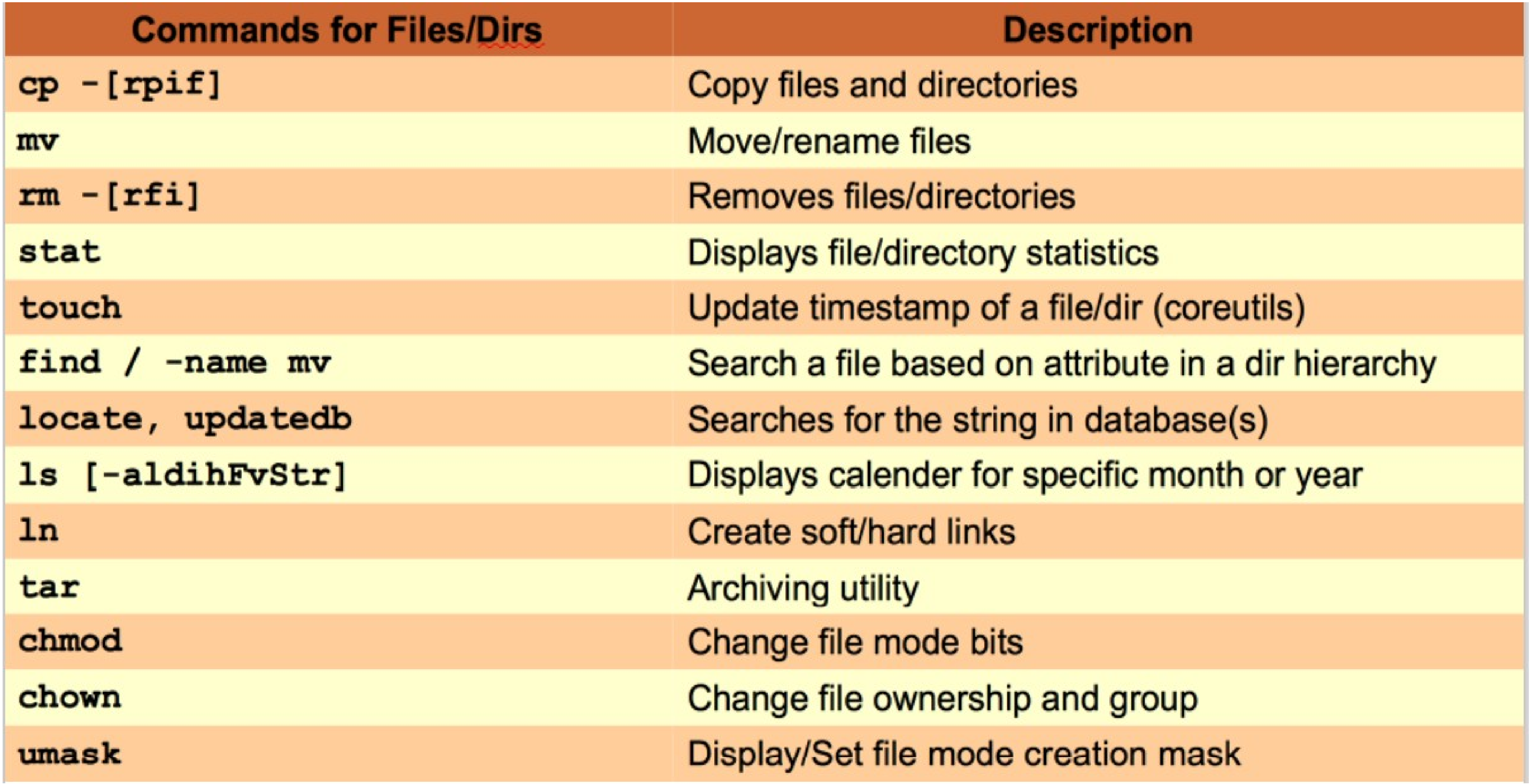


Commands for Files only	Description
`cat, less, more, head, tail`	View contents of a file
`file`	Determines file type
`wc`	Displays line, word, character count of file(s)
`uniq`	Report or omit repeated lines
`sort`	Sort lines of files
`cut`	Remove col(s) from tabular files (tab,collon,space)
`paste`	Horizontally concatenate contents of two or more file
`grep`	Prints lines of files where a pattern is matched
`gzip, gunzip, bzip2, bunzip2`	Compression and un-compression softwares



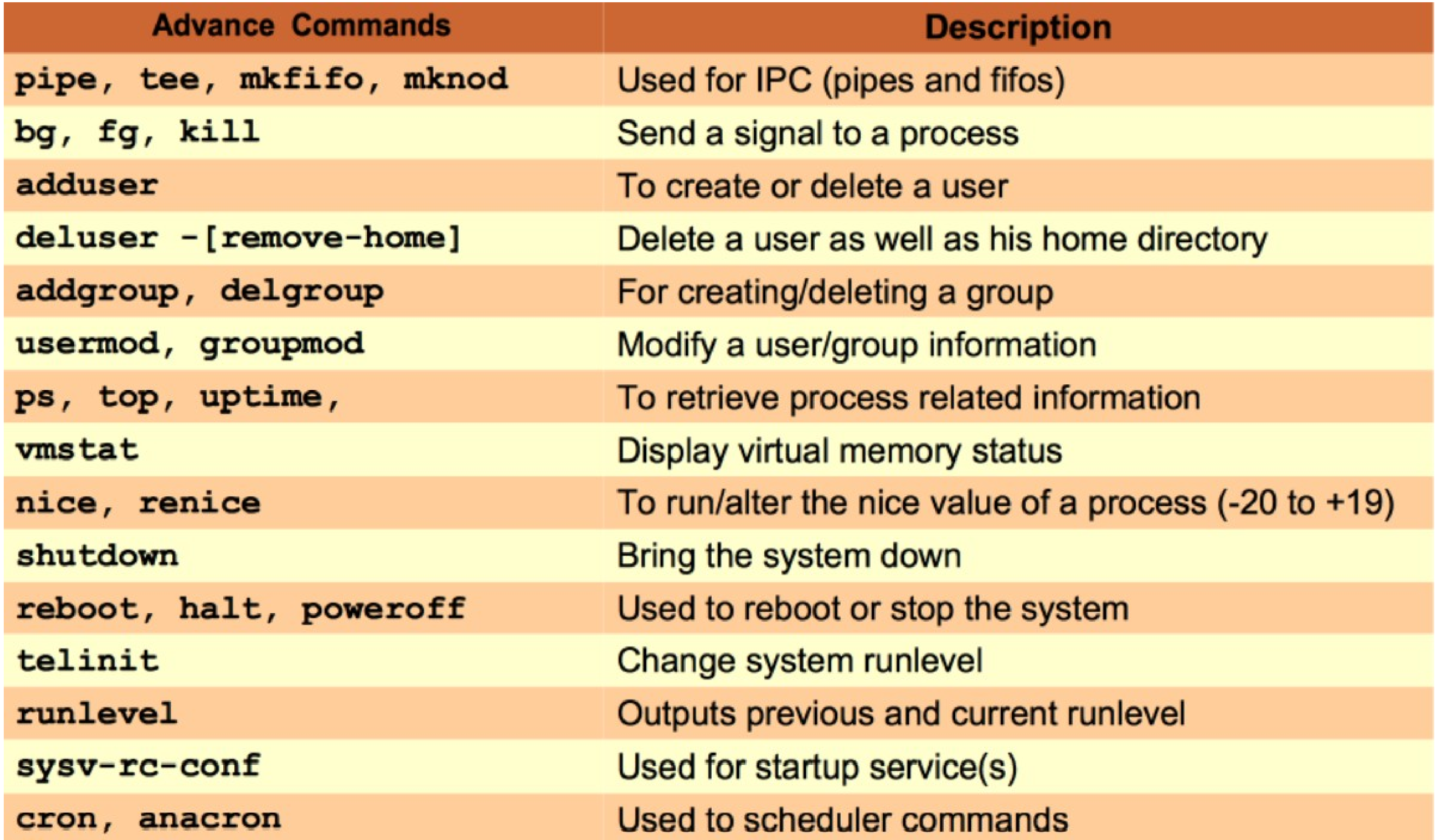
Commands for Dirs only	Description
`cd`	Change directory
`mkdir -[p], rmdir -[p]`	Create and remove a directory.
`pwd`	Display present working directory

Files and Directories Commands



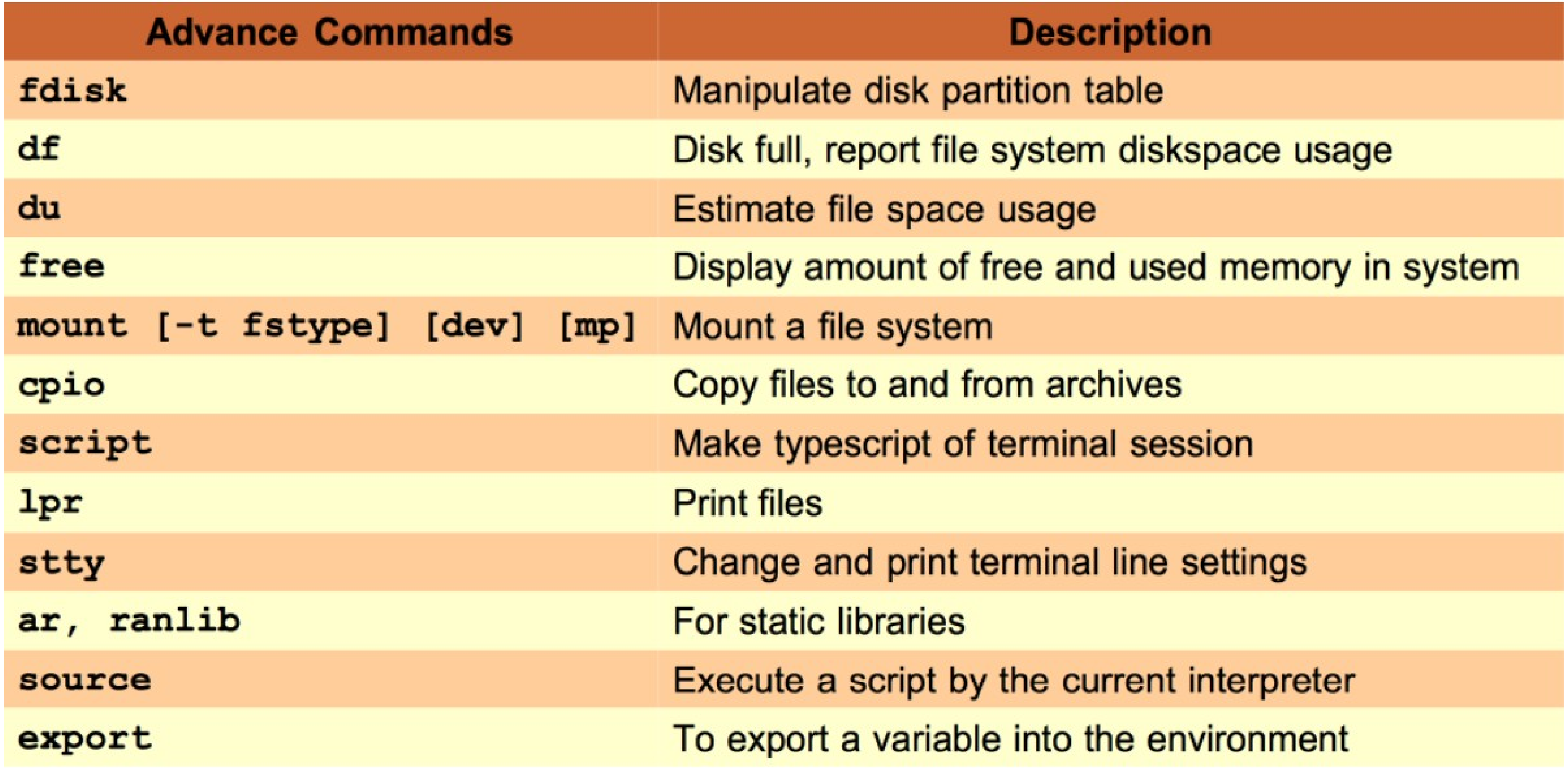
Commands for Files/Dirs	Description
`cp -[rpif]`	Copy files and directories
`mv`	Move/rename files
`rm -[rfi]`	Removes files/directories
`stat`	Displays file/directory statistics
`touch`	Update timestamp of a file/dir (coreutils)
`find / -name mv`	Search a file based on attribute in a dir hierarchy
`locate, updatedb`	Searches for the string in database(s)
`ls [-aldihFvStr]`	Displays calender for specific month or year
`ln`	Create soft/hard links
`tar`	Archiving utility
`chmod`	Change file mode bits
`chown`	Change file ownership and group
`umask`	Display/Set file mode creation mask

Advance Commands



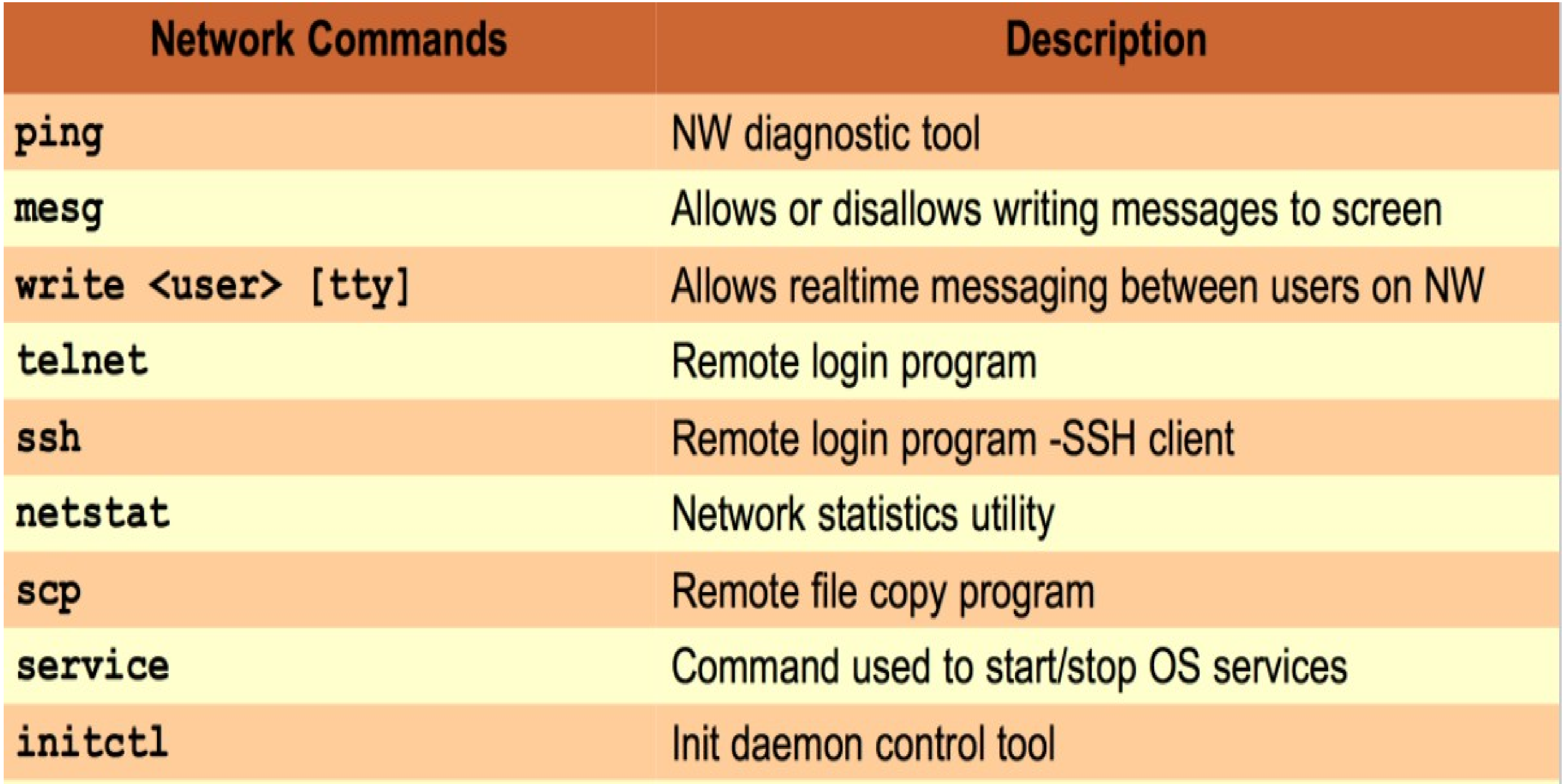
Advance Commands	Description
`pipe, tee, mkfifo, mknod`	Used for IPC (pipes and fifos)
`bg, fg, kill`	Send a signal to a process
`adduser`	To create or delete a user
`deluser -[remove-home]`	Delete a user as well as his home directory
`addgroup, delgroup`	For creating/deleting a group
`usermod, groupmod`	Modify a user/group information
`ps, top, uptime,`	To retrieve process related information
`vmstat`	Display virtual memory status
`nice, renice`	To run/alter the nice value of a process (-20 to +19)
`shutdown`	Bring the system down
`reboot, halt, poweroff`	Used to reboot or stop the system
`telinit`	Change system runlevel
`runlevel`	Outputs previous and current runlevel
`sysv-rc-conf`	Used for startup service(s)
`cron, anacron`	Used to scheduler commands

Advance Commands



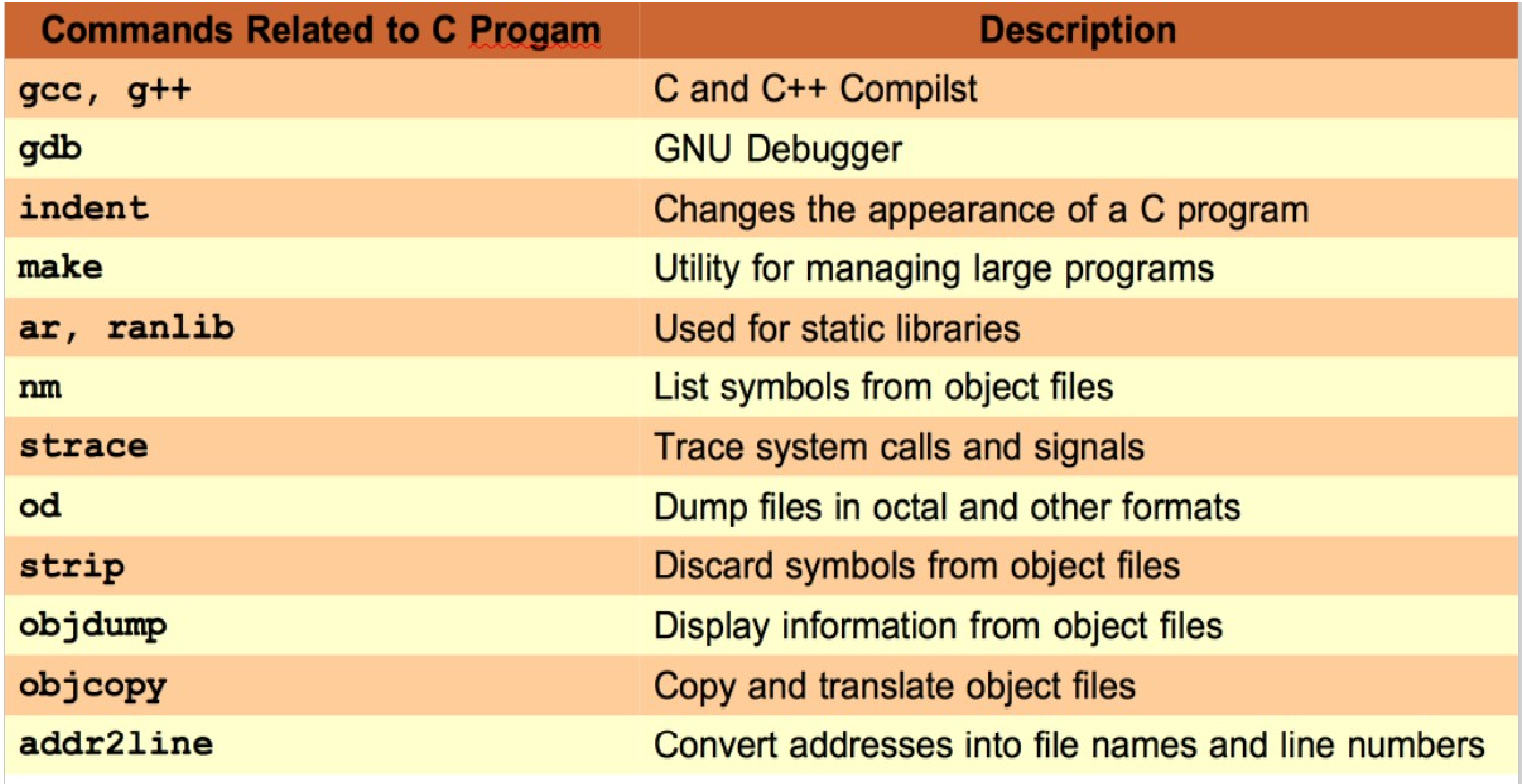
Advance Commands	Description
`fdisk`	Manipulate disk partition table
`df`	Disk full, report file system diskspace usage
`du`	Estimate file space usage
`free`	Display amount of free and used memory in system
`mount [-t fstype] [dev] [mp]`	Mount a file system
`cpio`	Copy files to and from archives
`script`	Make typescript of terminal session
`lpr`	Print files
`stty`	Change and print terminal line settings
`ar, ranlib`	For static libraries
`source`	Execute a script by the current interpreter
`export`	To export a variable into the environment

Network Commands



Network Commands	Description
`ping`	NW diagnostic tool
`mesg`	Allows or disallows writing messages to screen
`write <user> [tty]`	Allows realtime messaging between users on NW
`telnet`	Remote login program
`ssh`	Remote login program -SSH client
`netstat`	Network statistics utility
`scp`	Remote file copy program
`service`	Command used to start/stop OS services
`initctl`	Init daemon control tool

C Programming Commands



Commands Related to C Progam	Description
`gcc, g++`	C and C++ Compilst
`gdb`	GNU Debugger
`indent`	Changes the appearance of a C program
`make`	Utility for managing large programs
`ar, ranlib`	Used for static libraries
`nm`	List symbols from object files
`strace`	Trace system calls and signals
`od`	Dump files in octal and other formats
`strip`	Discard symbols from object files
`objdump`	Display information from object files
`objcopy`	Copy and translate object files
`addr2line`	Convert addresses into file names and line numbers

Linux Shell Shortcuts

- The BASH shell provides a simple, yet powerful command entry interface. Useful shortcuts include:

– Ctrl+a: move the cursor to the beginning of the command line

– Ctrl+c: terminate a running program and return to the shell prompt

– Ctrl+d: log out of the current shell

– Ctrl+e: move cursor to the end of the command line

– Ctrl+l: clear the shell terminal screen

– Ctrl+r: search the command history

– <tab>: autocomplete file name

– <tab><tab>: show command completion possibilities

– <up-arrow>: repeat last command (or scroll through history)