### Note on DOS (Disk Operating System)

DOS, or Disk Operating System, refers to a family of operating systems that dominated the personal computer market during the 1980s and the early 1990s. The most well-known version is MS-DOS (Microsoft Disk Operating System), which was developed by Microsoft and became the standard operating system for IBM-compatible personal computers.

##### History of DOS:

- CP/M and 86-DOS: The history of DOS can be traced back to CP/M (Control Program for Microcomputers), an operating system developed by Digital Research. In the early 1980s, IBM wanted an operating system for their first personal computer, the IBM PC. Microsoft purchased 86-DOS (QDOS), a CP/M clone, from Seattle Computer Products and adapted it for the IBM PC.

- MS-DOS: Microsoft licensed MS-DOS to IBM for use on the IBM PC. Eventually, Microsoft retained the rights to sell MS-DOS to other PC manufacturers, leading to widespread adoption. MS-DOS was characterized by its command-line interface, where users interacted with the system by typing commands.

- Versions and Evolution: MS-DOS went through various versions, with MS-DOS 6.22 being one of the most widely used versions. It saw significant improvements in file management, memory management, and device support.

##### Features of DOS:

1. Command-Line Interface (CLI): MS-DOS is primarily text-based, relying on a command-line interface for user interaction. Users input commands through a keyboard, and the system responds with text-based outputs.
2. File System: DOS uses a file allocation table (FAT) file system, which organizes and manages files on a disk. The file system supports simple file structures and basic attributes.
3. Memory Management: DOS operates in a relatively small memory footprint. Older versions of DOS were limited to 640 KB of RAM, and later versions introduced mechanisms like Extended Memory Specification (EMS) and eXtended Memory Manager (XMM) to utilize extended memory beyond this limit.
4. Single-Tasking: MS-DOS is a single-tasking operating system, meaning it can only execute one application at a time. Multitasking capabilities were introduced in later versions, but they were limited compared to modern operating systems.
5. Device Drivers: DOS relies on device drivers to communicate with hardware components. Users often needed to manually configure and load drivers for peripherals such as printers, displays, and disk drives.
6. Batch Processing: DOS supports batch files, which are sequences of commands stored in a text file. These files allow users to automate repetitive tasks by executing a series of commands with a single command.

##### Resource Management in DOS:

Resource management in DOS involves handling memory, files, and hardware resources efficiently. Given its origins in an era of limited computing resources, DOS had to be judicious in its use of available assets.

1. Memory Management: DOS manages memory using a combination of conventional, upper, and extended memory. It employs techniques like memory segmentation to make the most of the available RAM.

2. File System: The file system in DOS is hierarchical, organized through directories and subdirectories. Efficient file management is crucial for optimizing storage and retrieval of information.

3. Device Drivers: DOS relies on device drivers to facilitate communication with hardware components. Proper installation and configuration of these drivers are essential for seamless resource utilization.

4. Peripheral Management: DOS requires manual configuration of peripherals, and resource conflicts could arise if not managed properly. Users often had to allocate resources such as IRQs (Interrupt Request) and I/O addresses to avoid conflicts.

5. Task Switching: Multitasking in DOS is limited, but task-switching mechanisms were introduced in later versions. Efficient task switching helps in utilizing the CPU time effectively, especially in systems with multiple applications running.

### Note on Windows NT Operating System:

Windows NT (New Technology) is a family of operating systems developed by Microsoft. The first version, Windows NT 3.1, was released in 1993. Unlike its predecessor, MS-DOS, which was primarily designed for single-user systems, Windows NT was developed as a robust, multi-user, and multitasking operating system with a strong emphasis on security and stability. Windows NT laid the foundation for the modern Windows operating systems, including Windows 2000, XP, Vista, 7, 8, and 10.

The development of Windows NT marked a departure from the MS-DOS architecture, introducing a kernel that was separate from the user interface. It incorporated features inspired by the VMS (Virtual Memory System) operating system developed by Digital Equipment Corporation. Windows NT aimed to address the limitations of MS-DOS, offering improved performance, reliability, and support for advanced hardware and networking capabilities.

##### History of Windows NT:

- Windows NT 3.1 (1993): The first version of Windows NT was released in 1993. It was a 32-bit operating system with a modular architecture, separating the user interface from the kernel. This allowed for better stability and security compared to the MS-DOS-based Windows systems.

- Windows NT 4.0 (1996): This version introduced the Windows 95 interface and included improvements to performance and compatibility. It featured the Windows Shell and included the Windows NT Server and Windows NT Workstation editions.

- Windows 2000 (NT 5.0, 2000): Windows 2000, released in 2000, combined the consumer-oriented Windows 98 interface with the Windows NT architecture. It brought improvements in stability, security, and support for new technologies.

- Windows XP (NT 5.1, 2001): Windows XP was a highly successful version of Windows and combined the Windows 2000 codebase with a redesigned user interface. It became widely adopted and remained in use for many years.

##### Features of Windows NT:

1. Security: One of the defining features of Windows NT is its focus on security. It introduced a robust security model with user-level permissions, access control lists (ACLs), and a secure login system. This made it suitable for both personal and enterprise environments.
2. Multitasking and Multi-User Support: Windows NT was designed to support multitasking, allowing multiple applications to run simultaneously. It also had built-in support for multiple users, making it suitable for server environments where simultaneous access by multiple users is essential.
3. Stability and Reliability: Windows NT was renowned for its stability and reliability. The separation of the kernel from the user interface prevented one application from crashing the entire system. This feature was crucial for mission-critical applications and enterprise environments.
4. 32-Bit Architecture: Windows NT was a 32-bit operating system from the start, providing improved performance and support for larger amounts of memory compared to its 16-bit counterparts. This was a significant advancement in the era when 32-bit computing was becoming more prevalent.
5. Plug and Play: Windows NT introduced the Plug and Play architecture, making it easier to install and configure hardware devices. This feature simplified the process of adding or removing hardware components from a system.

##### Resource Management in Windows NT:

1. Memory Management: Windows NT features advanced memory management, including virtual memory support, process isolation, and a demand-paged virtual memory system. This allows efficient utilization of system memory and better handling of memory-intensive applications.
2. File System: Windows NT supports the NTFS (New Technology File System), a robust and secure file system that provides features like file encryption, compression, and support for large file sizes. NTFS enhances file organization and data integrity.
3. Device Drivers and Hardware Abstraction Layer (HAL): Windows NT uses a Hardware Abstraction Layer (HAL) to isolate the operating system from hardware variations. This allows for better compatibility and easier integration of device drivers, enhancing resource management for various hardware configurations.
4. Task Scheduler: The Windows NT Task Scheduler enables users to automate tasks and optimize resource usage by scheduling processes at specific times. This helps in efficient resource allocation and reduces manual intervention.
5. Networking Capabilities: Windows NT was designed with robust networking capabilities, making it suitable for server environments. It supports features like domain-based networking, TCP/IP protocol, and Active Directory for centralized network management.

### Note on Linux Operating System:

##### History of Linux:

Linux is an open-source, Unix-like operating system kernel that originated in 1991 with the efforts of Linus Torvalds, a Finnish computer science student. Linus started developing Linux as a hobby project, inspired by the Unix operating system. Over time, a community of developers worldwide contributed to its growth, creating a powerful and flexible operating system.

The Linux kernel, combined with GNU (GNU's Not Unix) utilities, resulted in a complete and free Unix-like operating system. Linux rapidly gained popularity due to its open-source nature, which allows users to access and modify the source code. This collaborative development model has led to the creation of numerous distributions or "distros," each tailored to specific user needs.

##### Features of Linux:

1. Open Source: Linux is open-source software, meaning its source code is freely available for anyone to view, modify, and distribute. This openness fosters collaboration and innovation within a global community of developers.
2. Multiuser and Multitasking: Linux supports multiple users and tasks concurrently. Users can log in and run various processes simultaneously, making it suitable for both personal and server use.
3. Stability and Reliability: Linux is known for its stability and reliability. Its robust design and separation of user space and kernel space contribute to fewer system crashes and increased uptime.
4. Security: Linux has a strong security model with features such as user-level permissions, access controls, and a robust firewall system. Regular security updates and patches from the community help address vulnerabilities promptly.
5. File System Support: Linux supports various file systems, including ext4, XFS, Btrfs, and more. Each file system has unique features, allowing users to choose the one that best suits their needs.
6. Networking Capabilities: Linux excels in networking, supporting a wide range of protocols and services. It is widely used as a server operating system, powering a significant portion of internet servers, routers, and other networking devices.
7. Command-Line Interface (CLI) and Graphical User Interface (GUI): Linux offers a powerful command-line interface, allowing users to interact with the system through text commands. Additionally, it supports graphical user interfaces like GNOME and KDE for users who prefer a more visually intuitive experience.

##### Resource Management in Linux:

1. Memory Management: Linux efficiently manages system memory, utilizing mechanisms like virtual memory, swap space, and memory sharing. The kernel is responsible for memory allocation and ensures optimal use of available resources.
2. Process Management: Linux supports multitasking and multiprocessing, allowing multiple processes to run concurrently. The kernel manages process scheduling, prioritizing tasks based on their priority and resource requirements.
3. File System Management: Linux supports a variety of file systems, and its virtual file system provides a unified interface for interacting with different file system types. This flexibility allows users to choose file systems based on their specific needs.
4. Device Management: Linux uses a modular approach to device drivers, allowing the kernel to load only the necessary drivers for detected hardware. This enhances resource efficiency and simplifies the addition or removal of hardware components.
5. Power Management: Linux supports power management features, allowing systems to optimize energy consumption. This is particularly important for mobile devices and laptops to extend battery life.
6. Task Scheduling: The Linux kernel includes a scheduler that efficiently allocates CPU time to processes. It employs various scheduling algorithms to ensure fair distribution of resources among running tasks.

### DOS commands

**ASSOC**

Displays or modifies file extension associations

ASSOC [.ext[=[fileType]]]

.ext Specifies the file extension to associate the file type with fileType

Type ASSOC without parameters to display the current file associations. If ASSOC is invoked with just a file extension, it displays the current file association for that file extension. Specify nothing for the file type and the command will delete the association for the file extension.ATTRIB

Displays or changes file attributes.

Example:

C:\Users\Student>assoc .txt

.txt=txtfile

**ATTRIB**

Display or change the attributes of a file or directory. Here is the syntax and some examples:

attrib [+R|-R] [+A|-A] [+S|-S] [+H|-H] [drive:][path][filename] [/S [/D]]

Parameters:

- `+R` / `-R`: Sets or clears the Read-only attribute.

- `+A` / `-A`: Sets or clears the Archive attribute.

- `+S` / `-S`: Sets or clears the System attribute.

- `+H` / `-H`: Sets or clears the Hidden attribute.

- `[drive:][path][filename]`: Specifies the file or directory for which you want to display or change attributes.

- `/S`: Processes matching files in the current directory and all of its subdirectories.

- `/D`: Processes folders as well.

Examples:

1. To display the attributes of a file:

attrib filename.txt

2. To add the Read-only attribute to a file:

attrib +R filename.txt

3. To remove the Hidden attribute from a directory and all its subdirectories:

attrib -H /S /D C:\example\directory

4. To set the Archive attribute for all files in a directory and its subdirectories:

attrib +A /S /D C:\example\directory\\*.\*

**CHDIR | CD**

Displays the name of or changes the current directory.

CHDIR [/D] [drive:][path]

CHDIR [..]

CD [/D] [drive:][path]

CD [..]

.. Specifies that you want to change to the parent directory.

Type CD drive: to display the current directory in the specified drive.

Type CD without parameters to display the current drive and directory.

Use the /D switch to change current drive in addition to changing current directory for a drive.

**CHKDSK**

Checks a disk and displays a status report.

CHKDSK [volume[[path]filename]]] [/F] [/V] [/R] [/X] [/I] [/C] [/L[:size]] [/B] [/scan] [/spotfix]

volume Specifies the drive letter (followed by a colon), point, or volume name.

filename FAT/FAT32 only: Specifies the files to check for fragmentation.

/F Fixes errors on the disk.

/V On FAT/FAT32: Displays the full path and name of every file on the disk.

On NTFS: Displays cleanup messages if any.

/R Locates bad sectors and recovers readable information (implies /F, when /scan not specified).

/L:size NTFS only: Changes the log file size to the specified number of kilobytes. If size is not specified, displays current size.

/X Forces the volume to dismount first if necessary. All opened handles to the volume would then be invalid (implies /F).

/I NTFS only: Performs a less vigorous check of index entries.

/C NTFS only: Skips checking of cycles within the folder structure.

/B NTFS only: Re-evaluates bad clusters on the volume (implies /R)

/scan NTFS only: Runs an online scan on the volume

/forceofflinefix NTFS only: (Must be used with "/scan") Bypass all online repair; all defects found are queued for offline repair (i.e. "chkdsk /spotfix").

/perf NTFS only: (Must be used with "/scan") Uses more system resources to complete a scan as fast as

possible. This may have a negative performance impact on

other tasks running on the system.

/spotfix NTFS only: Runs spot fixing on the volume

/sdcleanup NTFS only: Garbage collect unneeded security descriptor data (implies /F).

/offlinescanandfix Runs an offline scan and fix on the volume.

/freeorphanedchains FAT/FAT32/exFAT only: Frees any orphaned cluster chains instead of recovering their contents.

/markclean FAT/FAT32/exFAT only: Marks the volume clean if no corruption was detected, even if /F was not specified.

The /I or /C switch reduces the amount of time required to run Chkdsk by skipping certain checks of the volume.

**CHKNTFS**

Displays or modifies the checking of disk at boot time.

CHKNTFS volume [...]

CHKNTFS /D

CHKNTFS /T[:time]

CHKNTFS /X volume [...]

CHKNTFS /C volume [...]

volume Specifies the drive letter (followed by a colon), mount point, or volume name.

/D Restores the machine to the default behavior; all drives are checked at boot time and chkdsk is run on those that are dirty.

/T:time Changes the AUTOCHK initiation countdown time to the specified amount of time in seconds. If time is not specified, displays the current setting.

/X Excludes a drive from the default boot-time check. Excluded drives are not accumulated between command invocations.

/C Schedules a drive to be checked at boot time; chkdsk will run if the drive is dirty.

If no switches are specified, CHKNTFS will display if the specified drive is dirty or scheduled to be checked on next reboot.

**CLS**

Clears the screen.

**CMD**

Starts a new instance of the Windows command interpreter

CMD [/A | /U] [/Q] [/D] [/E:ON | /E:OFF] [/F:ON | /F:OFF] [/V:ON | /V:OFF]

[[/S] [/C | /K] string]

/C Carries out the command specified by string and then terminates

/K Carries out the command specified by string but remains

/S Modifies the treatment of string after /C or /K (see below)

/Q Turns echo off

/D Disable execution of AutoRun commands from registry (see below)

/A Causes the output of internal commands to a pipe or file to be ANSI

/U Causes the output of internal commands to a pipe or file to be Unicode

/T:fg Sets the foreground/background colors (see COLOR /? for more info)

/E:ON Enable command extensions (see below)

/E:OFF Disable command extensions (see below)

/F:ON Enable file and directory name completion characters (see below)

/F:OFF Disable file and directory name completion characters (see below)

/V:ON Enable delayed environment variable expansion using ! as the delimiter. For example, /V:ON would allow !var! to expand the variable var at execution time. The var syntax expands variables at input time, which is quite a different thing when inside of a FOR loop.

/V:OFF Disable delayed environment expansion.

Note that multiple commands separated by the command separator '&&' are accepted for string if surrounded by quotes. Also, for compatibility reasons, /X is the same as /E:ON, /Y is the same as /E:OFF and /R is the same as /C. Any other switches are ignored.

**COLOR**

color//

Sets the default console foreground and background colors.

COLOR [attr]

attr Specifies color attribute of console output

Color attributes are specified by TWO hex digits -- the first corresponds to the background; the second the foreground. Each digit can be any of the following values:

0 = Black 8 = Gray

1 = Blue 9 = Light Blue

2 = Green A = Light Green

3 = Aqua B = Light Aqua

4 = Red C = Light Red

5 = Purple D = Light Purple

6 = Yellow E = Light Yellow

7 = White F = Bright White

If no argument is given, this command restores the color to what it was when CMD.EXE started. This value either comes from the current console window, the /T command line switch or from the DefaultColor registry value.

The COLOR command sets ERRORLEVEL to 1 if an attempt is made to execute the COLOR command with a foreground and background color that are the same.

Example: "COLOR fc" produces light red on bright white

**COMP**

Compares the contents of two files or sets of files.

COMP [data1] [data2] [/D] [/A] [/L] [/N=number] [/C] [/OFF[LINE]] [/M]

data1 Specifies location and name(s) of first file(s) to compare.

data2 Specifies location and name(s) of second files to compare.

/D Displays differences in decimal format.

/A Displays differences in ASCII characters.

/L Displays line numbers for differences.

/N=number Compares only the first specified number of lines in each file.

/C Disregards case of ASCII letters when comparing files.

/OFF[LINE] Do not skip files with offline attribute set.

/M Do not prompt for compare more files.

To compare sets of files, use wildcards in data1 and data2 parameters.

**FC**

Compares two files or sets of files and displays the differences between them

FC [/A] [/C] [/L] [/LBn] [/N] [/OFF[LINE]] [/T] [/U] [/W] [/nnnn]

[drive1:][path1]filename1 [drive2:][path2]filename2

FC /B [drive1:][path1]filename1 [drive2:][path2]filename2

/A Displays only first and last lines for each set of differences.

/B Performs a binary comparison.

/C Disregards the case of letters.

/L Compares files as ASCII text.

/LBn Sets the maximum consecutive mismatches to the specified number of lines.

/N Displays the line numbers on an ASCII comparison.

/OFF[LINE] Do not skip files with offline attribute set.

/T Does not expand tabs to spaces.

/U Compare files as UNICODE text files.

/W Compresses white space (tabs and spaces) for comparison.

/nnnn Specifies the number of consecutive lines that must match after a mismatch.

[drive1:][path1]filename1

Specifies the first file or set of files to compare.

[drive2:][path2]filename2

Specifies the second file or set of files to compare.

**CONVERT**

Converts a FAT volume to NTFS.

CONVERT volume /FS:NTFS [/V] [/CvtArea:filename] [/NoSecurity] [/X]

volume Specifies the drive letter (followed by a colon), mount point, or volume name.

/FS:NTFS Specifies that the volume will be converted to NTFS.

/V Specifies that Convert will be run in verbose mode.

/CvtArea:filename

Specifies a contiguous file in the root directory that will be the place holder for NTFS system files.

/NoSecurity Specifies that the security settings on the converted files and directories allow access by all users.

/X Forces the volume to dismount first if necessary. All open handles to the volume will not be valid.

**COPY**

Copies one or more files to another location.

COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/L] [/A | /B ] source [/A | /B]

[+ source [/A | /B] [+ ...]] [destination [/A | /B]]

source Specifies the file or files to be copied.

/A Indicates an ASCII text file.

/B Indicates a binary file.

/D Allow the destination file to be created decrypted destination Specifies the directory and/or filename for the new file(s).

/V Verifies that new files are written correctly.

/N Uses short filename, if available, when copying a file with a non-8dot3 name.

/Y Suppresses prompting to confirm you want to overwrite an existing destination file.

/-Y Causes prompting to confirm you want to overwrite an existing destination file.

/Z Copies networked files in restartable mode.

/L If the source is a symbolic link, copy the link to the target instead of the actual file the source link points to.

The switch /Y may be preset in the COPYCMD environment variable. This may be overridden with /-Y on the command line. Default is to prompt on overwrites unless COPY command is being executed from

within a batch script.

To append files, specify a single file for destination, but multiple files

for source (using wildcards or file1+file2+file3 format).

**C0PY CON**

The `COPY CON` command in MS-DOS is used to create or edit a text file directly from the command prompt. The name "CON" is short for console, representing the console input. This command is often used to quickly create small text files without using a separate text editor.

Here's how you can use the `COPY CON` command:

COPY CON filename.txt

- To end input:

- To end the input and save the file, press `Ctrl + Z` (or `Ctrl + D` on some systems), and then press Enter.

- Example:

- To create a new text file named "example.txt," you would enter:

COPY CON example.txt

This is some text.

Press Ctrl+Z to save and exit.

- Note:

- The `CON` device is used for console input. When you use `COPY CON`, you are essentially copying from the console (your keyboard) to a file.

Keep in mind that the `COPY CON` method is suitable for small text files, and it might not be the most convenient way to handle larger or more complex documents. For more extensive text editing tasks, you may prefer to use a dedicated text editor or word processor.

**CLIP**

Description:

Redirects output of command line tools to the Windows clipboard. This text output can then be pasted into other programs.

Parameter List:

/? Displays this help message.

Examples:

DIR | CLIP Places a copy of the current directory listing into the Windows clipboard.

CLIP < README.TXT Places a copy of the text from readme.txt on to the Windows clipboard.

**DATE**

Displays or sets the date.

DATE [/T | date]

Type DATE without parameters to display the current date setting and a prompt for a new one. Press ENTER to keep the same date.

**DEL**

Deletes one or more files.

DEL [/P] [/F] [/S] [/Q] [/A[[:]attributes]] names

ERASE [/P] [/F] [/S] [/Q] [/A[[:]attributes]] names

names Specifies a list of one or more files or directories. Wildcards may be used to delete multiple files. If a directory is specified, all files within the directory will be deleted.

/P Prompts for confirmation before deleting each file.

/F Force deleting of read-only files.

/S Delete specified files from all subdirectories.

/Q Quiet mode, do not ask if ok to delete on global wildcard

/A Selects files to delete based on attributes

attributes R Read-only files S System files

H Hidden files A Files ready for archiving

I Not content indexed Files L Reparse Points

O Offline files - Prefix meaning not

If Command Extensions are enabled DEL and ERASE change as follows:

The display semantics of the /S switch are reversed in that it shows you only the files that are deleted, not the ones it could not find.

**DIR**

Displays a list of files and subdirectories in a directory.

DIR [drive:][path][filename] [/A[[:]attributes]] [/B] [/C] [/D] [/L] [/N] [/O[[:]sortorder]] [/P] [/Q] [/R] [/S] [/T[[:]timefield]] [/W] [/X] [/4]

[drive:][path][filename]

Specifies drive, directory, and/or files to list.

/A Displays files with specified attributes.

attributes D Directories R Read-only files

H Hidden files A Files ready for archiving

S System files I Not content indexed files

L Reparse Points O Offline files

- Prefix meaning not

/O List by files in sorted order.

sortorder N By name (alphabetic) S By size (smallest first)

E By extension (alphabetic) D By date/time (oldest first)

G Group directories first - Prefix to reverse order

/S Displays files in specified directory and all subdirectories.

/W Uses wide list format.

**DISKPART**

Displays or configures Disk Partition properties.

**DOSKEY**

Edits command lines, recalls Windows commands, and creates macros.

DOSKEY [/REINSTALL] [/LISTSIZE=size] [/MACROS[:ALL | :exename]]

[/HISTORY] [/INSERT | /OVERSTRIKE] [/EXENAME=exename] [/MACROFILE=filename]

[macroname=[text]]

/REINSTALL Installs a new copy of Doskey.

/LISTSIZE=size Sets size of command history buffer.

/MACROS Displays all Doskey macros.

/MACROS:ALL Displays all Doskey macros for all executables which have Doskey macros.

/MACROS:exename Displays all Doskey macros for the given executable.

/HISTORY Displays all commands stored in memory.

/INSERT Specifies that new text you type is inserted in old text.

/OVERSTRIKE Specifies that new text overwrites old text.

/EXENAME=exename Specifies the executable.

/MACROFILE=filename Specifies a file of macros to install.

macroname Specifies a name for a macro you create.

text Specifies commands you want to record.

UP and DOWN ARROWS recall commands; ESC clears command line; F7 displays command history; ALT+F7 clears command history; F8 searches command history; F9 selects a command by number; ALT+F10 clears macro definitions.

The following are some special codes in Doskey macro definitions:

$T Command separator. Allows multiple commands in a macro.

$1-$9 Batch parameters. Equivalent to %1-%9 in batch programs.

$\* Symbol replaced by everything following macro name on command line.

**ECHO**

Displays messages, or turns command-echoing on or off.

ECHO [ON | OFF]

ECHO [message]

Type ECHO without parameters to display the current echo setting.

**ERASE**

Deletes one or more files.

DEL [/P] [/F] [/S] [/Q] [/A[[:]attributes]] names

ERASE [/P] [/F] [/S] [/Q] [/A[[:]attributes]] names

names Specifies a list of one or more files or directories. Wildcards may be used to delete multiple files. If a directory is specified, all files within the directory will be deleted.

/P Prompts for confirmation before deleting each file.

/F Force deleting of read-only files.

/S Delete specified files from all subdirectories.

/Q Quiet mode, do not ask if ok to delete on global wildcard

/A Selects files to delete based on attributes

attributes R Read-only files S System files

H Hidden files A Files ready for archiving

I Not content indexed Files L Reparse Points

O Offline files - Prefix meaning not

**EXIT**

Quits the CMD.EXE program (command interpreter).

**FIND**

Searches for a text string in a file or files.

FIND [/V] [/C] [/N] [/I] [/OFF[LINE]] "string" [[drive:][path]filename[ ...]]

/V Displays all lines NOT containing the specified string.

/C Displays only the count of lines containing the string.

/N Displays line numbers with the displayed lines.

/I Ignores the case of characters when searching for the string.

/OFF[LINE] Do not skip files with offline attribute set.

"string" Specifies the text string to find.

[drive:][path]filename

Specifies a file or files to search.

If a path is not specified, FIND searches the text typed at the prompt or piped from another command.

**FINDSTR**

Searches for strings in files.

FINDSTR [/B] [/E] [/L] [/R] [/S] [/I] [/X] [/V] [/N] [/M] [/O] [/P] [/F:file]

[/C:string] [/G:file] [/D:dir list] [/A:color attributes] [/OFF[LINE]]

strings [[drive:][path]filename[ ...]]

/B Matches pattern if at the beginning of a line.

/E Matches pattern if at the end of a line.

/L Uses search strings literally.

/R Uses search strings as regular expressions.

/S Searches for matching files in the current directory and all subdirectories.

/I Specifies that the search is not to be case-sensitive.

/X Prints lines that match exactly.

/V Prints only lines that do not contain a match.

/N Prints the line number before each line that matches.

/M Prints only the filename if a file contains a match.

/O Prints character offset before each matching line.

/P Skip files with non-printable characters.

/OFF[LINE] Do not skip files with offline attribute set.

/A:attr Specifies color attribute with two hex digits. See "color /?"

/F:file Reads file list from the specified file(/ stands for console).

/C:string Uses specified string as a literal search string.

/G:file Gets search strings from the specified file(/ stands for console).

/D:dir Search a semicolon delimited list of directories

strings Text to be searched for.

[drive:][path]filename

Specifies a file or files to search.

Use spaces to separate multiple search strings unless the argument is prefixed with /C. For example, 'FINDSTR "hello there" x.y' searches for "hello" or "there" in file x.y. 'FINDSTR /C:"hello there" x.y' searches for

"hello there" in file x.y.

**FORMAT**

Formats a disk for use with Windows.

FORMAT volume [/FS:file-system] [/V:label] [/Q] [/L[:state]] [/A:size] [/C] [/I:state] [/X] [/P:passes] [/S:state]

FORMAT volume [/V:label] [/Q] [/F:size] [/P:passes]

FORMAT volume [/V:label] [/Q] [/T:tracks /N:sectors] [/P:passes]

FORMAT volume [/V:label] [/Q] [/P:passes]

FORMAT volume [/Q]

volume Specifies the drive letter (followed by a colon), mount point, or volume name.

/FS:filesystem Specifies the type of the file system (FAT, FAT32, exFAT, NTFS, UDF, ReFS).

/V:label Specifies the volume label.

/Q Performs a quick format. Note that this switch overrides /P.

/C NTFS only: Files created on the new volume will be compressed by default.

/X Forces the volume to dismount first if necessary. All opened handles to the volume would no longer be valid.

/R:revision UDF only: Forces the format to a specific UDF version (1.02, 1.50, 2.00, 2.01, 2.50). The default revision is 2.01.

/D UDF 2.50 only: Metadata will be duplicated.

/L[:state] NTFS Only: Overrides the default size of file record. default, a non-tiered volume will be formatted with small size file records and a tiered volume will be formatted with large size file records. /L and /L:enable forces format to use large size file records and /L:disable forces format to use small size file records.

/A:size Overrides the default allocation unit size. Default settings are strongly recommended for general use.

**FSUTIL**

Displays or configures the file system properties.

The `fsutil` command is a Windows Command Prompt utility that provides various file and volume-related operations. In particular, the `fsutil` command has functionality related to managing file systems, file streams, hard links, and quotas. Here are some common tasks you can perform using `fsutil`:

1. File System Information:

- `fsutil fsinfo drives`: Lists all available drives on the system.

- `fsutil fsinfo volumeinfo C:`: Displays information about the specified volume (replace "C:" with the appropriate drive letter).

2. File Streams:

- `fsutil stream`: Displays information about streams on a file.

- `fsutil stream query <filename>`: Lists all named streams on a file.

3. Hard Links:

- `fsutil hardlink create <linkname> <filename>`: Creates a hard link between two files.

- `fsutil hardlink list <filename>`: Lists all hard links associated with a file.

4. Quota Management:

- `fsutil quota query C:`: Displays disk space usage and quota information for a specified volume.

- `fsutil quota enforce C:`: Enables quota enforcement on the specified volume.

5. File Compression:

- `fsutil compression query C:`: Displays the compression state of files on a specified volume.

- `fsutil compression set <filename>`: Sets the compression state of a file.

6. Sparse Files:

- `fsutil sparse queryflag <filename>`: Displays the sparse file attribute for a file.

- `fsutil sparse setflag <filename>`: Sets the sparse file attribute for a file.

7. Object ID:

- `fsutil objectid query C:`: Displays the object ID for a specified volume.

- `fsutil objectid setinfo <filename>`: Sets the object ID for a file.

8. Transactional NTFS:

- `fsutil transaction start <filename>`: Starts a transaction for a specified file.

- `fsutil transaction commit <filename>`: Commits a transaction for a specified file.

**FTYPE**

Displays or modifies file types used in file extension associations

FTYPE [fileType[=[openCommandString]]]

fileType Specifies the file type to examine or change openCommandString Specifies the open command to use when launching files of this type.

Type FTYPE without parameters to display the current file types that have open command strings defined. FTYPE is invoked with just a file type, it displays the current open command string for that file type.

Specify nothing for the open command string and the FTYPE command will delete the open command string for the file type. Within an open command string %0 or %1 are substituted with the file name being launched through the assocation. %\* gets all the parameters and %2 gets the 1st parameter, %3 the second, etc. %~n gets all the remaining parameters starting with the nth parameter, where n may be between 2 and 9, inclusive. For example:

ASSOC .pl=PerlScript

FTYPE PerlScript=perl.exe %1 %\*

would allow you to invoke a Perl script as follows:

script.pl 1 2 3

If you want to eliminate the need to type the extensions, then do the

following:

set PATHEXT=.pl;%PATHEXT%

and the script could be invoked as follows:

script 1 2 3

**HELP**

Provides Help information for commands.

**MD | MKDIR**

Creates a directory.

MKDIR [drive:]path

MD [drive:]path

If Command Extensions are enabled MKDIR changes as follows:

MKDIR creates any intermediate directories in the path, if needed. For example, assume \a does not exist then:

mkdir \a\b\c\d

is the same as:

mkdir \a

chdir \a

mkdir b

chdir b

mkdir c

chdir c

mkdir d

which is what you would have to type if extensions were disabled.

**MOVE**

Moves one or more files from one directory to another directory.

Moves files and renames files and directories.

To move one or more files:

MOVE [/Y | /-Y] [drive:][path]filename1[,...] destination

To rename a directory:

MOVE [/Y | /-Y] [drive:][path]dirname1 dirname2

[drive:][path]filename1 Specifies the location and name of the file or files you want to move.

destination Specifies the new location of the file. Destination can consist of a drive letter and colon, a directory name, or a combination. If you are moving only one file, you can also include a filename if you want to rename the file when you move it.

[drive:][path]dirname1 Specifies the directory you want to rename.

dirname2 Specifies the new name of the directory.

/Y Suppresses prompting to confirm you want to overwrite an existing destination file.

/-Y Causes prompting to confirm you want to overwrite an existing destination file.

**NET USER:**

- Syntax: `net user [username [password | \*] [options]] [/domain]`

- Description: Manages user accounts on a Windows system. You can use it to create, modify, or display user accounts.

Examples:

- Display information about a user:

net user username

- Change a user's password:

net user username newpassword

**NOTEPAD:**

- Syntax: `notepad [filename]`

- Description: Opens the Notepad text editor. If a filename is provided, it opens that specific file in Notepad.

Example:

notepad example.txt

\*When windows notepad application is installed.

**PATH**

Displays or sets a search path for executable files.

PATH [[drive:]path[;...][;%PATH%]

PATH ;

Type PATH ; to clear all search-path settings and direct cmd.exe to search only in the current directory.

Type PATH without parameters to display the current path.

Including %PATH% in the new path setting causes the old path to be appended to the new setting.

**PRINT**

Prints a text file.

PRINT [/D:device] [[drive:][path]filename[...]]

/D:device Specifies a print device.

**PROMPT**

Changes the cmd.exe command prompt.

PROMPT [text]

text Specifies a new command prompt.

prompt can be made up of normal characters and the following special codes:

$A & (Ampersand)

$B | (pipe)

$C ( (Left parenthesis)

$D Current date

$E Escape code (ASCII code 27)

$F ) (Right parenthesis)

$G > (greater-than sign)

$H Backspace (erases previous character)

$L < (less-than sign)

$N Current drive

$P Current drive and path

$Q = (equal sign)

$S (space)

$T Current time

$V Windows version number

$\_ Carriage return and linefeed

$$ $ (dollar sign)

**RD**

Removes (deletes) a directory.

RMDIR [/S] [/Q] [drive:]path

RD [/S] [/Q] [drive:]path

/S Removes all directories and files in the specified directory in addition to the directory itself. Used to remove a directory tree.

/Q Quiet mode, do not ask if ok to remove a directory tree with /S

**RECOVER**

Recovers readable information from a bad or defective disk.

RECOVER [drive:][path]filename

**REM**

Records comments (remarks) in a batch file or CONFIG.SYS.

REM [comment]

**REN |RENAME**

Renames a file or files.

RENAME [drive:][path]filename1 filename2.

REN [drive:][path]filename1 filename2.

**REPLACE**

Replaces files.

REPLACE [drive1:][path1]filename [drive2:][path2] [/A] [/P] [/R] [/W]

REPLACE [drive1:][path1]filename [drive2:][path2] [/P] [/R] [/S] [/W] [/U]

[drive1:][path1]filename Specifies the source file or files.

[drive2:][path2] Specifies the directory where files are to be replaced.

/A Adds new files to destination directory. Cannot use with /S or /U switches.

/P Prompts for confirmation before replacing a file or adding a source file.

/R Replaces read-only files as well as unprotected files.

/S Replaces files in all subdirectories of the destination directory. Cannot use with the /A switch.

/W Waits for you to insert a disk before beginning.

/U Replaces (updates) only files that are older than source files. Cannot use with the /A switch.

**RMDIR**

Removes (deletes) a directory.

RMDIR [/S] [/Q] [drive:]path

RD [/S] [/Q] [drive:]path

/S Removes all directories and files in the specified directory in addition to the directory itself. Used to remove a directory tree.

/Q Quiet mode, do not ask if ok to remove a directory tree with /S

**SET**

Displays, sets, or removes cmd.exe environment variables.

SET [variable=[string]]

variable Specifies the environment-variable name.

string Specifies a series of characters to assign to the variable.

Type SET without parameters to display the current environment variables.

**SHUTDOWN**

Allows proper local or remote shutdown of machine.

Usage: shutdown [/i | /l | /s | /sg | /r | /g | /a | /p | /h | /e | /o] [/hybrid] [/soft] [/fw] [/f]

[/m \\computer][/t xxx][/d [p|u:]xx:yy [/c "comment"]]

No args Display help. This is the same as typing /?.

/? Display help. This is the same as not typing any options.

/i Display the graphical user interface (GUI).

/l Log off. This cannot be used with /m or /d options.

/s Shutdown the computer.

/sg Shutdown the computer. On the next boot, if Automatic Restart Sign-On is enabled, automatically sign in and lock last interactive user. After sign in, restart any registered applications.

/r Full shutdown and restart the computer.

/g Full shutdown and restart the computer. After the system is rebooted,

if Automatic Restart Sign-On is enabled, automatically sign in and lock last interactive user.

After sign in, restart any registered applications.

/a Abort a system shutdown. This can only be used during the time-out period.

Combine with /fw to clear any pending boots to firmware.

/p Turn off the local computer with no time-out or warning. Can be used with /d and /f options.

/h Hibernate the local computer. Can be used with the /f option.

/hybrid Performs a shutdown of the computer and prepares it for fast startup.

/f Force running applications to close without forewarning users.

**START**

Starts a separate window to run a specified program or command.

START ["title"] [/D path] [/I] [/MIN] [/MAX] [/SEPARATE | /SHARED] [/LOW | /NORMAL | /HIGH | /REALTIME | /ABOVENORMAL | /BELOWNORMAL] [/NODE <NUMA node>] [/AFFINITY <hex affinity mask>] [/WAIT] [/B] [/MACHINE <x86|amd64|arm|arm64>][command/program] [parameters]

"title" Title to display in window title bar.

path Starting directory.

B Start application without creating a new window. The application has ^C handling ignored. Unless the application enables ^C processing, ^Break is the only way to interrupt the application.

I The new environment will be the original environment passed to the cmd.exe and not the current environment.

MIN Start window minimized.

MAX Start window maximized.

**SUBST**

Associates a path with a drive letter.

SUBST [drive1: [drive2:]path]

SUBST drive1: /D

drive1: Specifies a virtual drive to which you want to assign a path.

[drive2:]path Specifies a physical drive and path you want to assign to a virtual drive.

/D Deletes a substituted (virtual) drive.

Type SUBST with no parameters to display a list of current virtual drives.

**SYSTEMINFO**

Displays machine specific properties and configuration.

SYSTEMINFO [/S system [/U username [/P [password]]]] [/FO format] [/NH]

Description:

This tool displays operating system configuration information for a local or remote machine, including service pack levels.

Parameter List:

/S system Specifies the remote system to connect to.

/U [domain\]user Specifies the user context under which the command should execute.

/P [password] Specifies the password for the given user context. Prompts for input if omitted.

/FO format Specifies the format in which the output is to be displayed. values: "TABLE", "LIST", "CSV".

/NH Specifies that the "Column Header" should not be displayed in the output.

Valid only for "TABLE" and "CSV" formats.

/? Displays this help message.

**TASKLIST**

TASKLIST [/S system [/U username [/P [password]]]] [/M [module] | /SVC | /V] [/FI filter] [/FO format] [/NH]

Description:

This tool displays a list of currently running processes on either a local or remote machine.

Parameter List:

/S system Specifies the remote system to connect to.

/U [domain\]user Specifies the user context under which the command should execute.

/P [password] Specifies the password for the given user context. Prompts for input if omitted.

/M [module] Lists all tasks currently using the given exe/dll name. If the module name is not specified all loaded modules are displayed.

/SVC Displays services hosted in each process.

/APPS Displays Store Apps and their associated processes.

/V Displays verbose task information.

/FI filter Displays a set of tasks that match a given criteria specified by the filter.

/FO format Specifies the output format. Valid values: "TABLE", "LIST", "CSV".

/NH Specifies that the "Column Header" should not be displayed in the output. Valid only for "TABLE" and "CSV" formats.

/? Displays this help message.

**TASKKILL**

TASKKILL [/S system [/U username [/P [password]]]]

{ [/FI filter] [/PID processid | /IM imagename] } [/T] [/F]

Description:

This tool is used to terminate tasks by process id (PID) or image name.

Parameter List:

/S system Specifies the remote system to connect to.

/U [domain\]user Specifies the user context under which the command should execute.

/P [password] Specifies the password for the given user context. Prompts for input if omitted.

/FI filter Applies a filter to select a set of tasks. Allows "\*" to be used. ex. imagename eq acme\*

/PID processid Specifies the PID of the process to be terminated. Use TaskList to get the PID.

/IM imagename Specifies the image name of the process to be terminated. Wildcard '\*' can be used specify all tasks or image names.

/T Terminates the specified process and any child processes which were started by it.

/F Specifies to forcefully terminate the process(es).

/? Displays this help message.

**TIME**

Displays or sets the system time.

TIME [/T | time]

Type TIME with no parameters to display the current time setting and a prompt for a new one. Press ENTER to keep the same time.

If Command Extensions are enabled the TIME command supports the /T switch which tells the command to just output the current time, without prompting for a new time.

**TITLE**

Sets the window title for the command prompt window.

TITLE [string]

string Specifies the title for the command prompt window.

**TREE**

Graphically displays the folder structure of a drive or path.

TREE [drive:][path] [/F] [/A]

/F Display the names of the files in each folder.

/A Use ASCII instead of extended characters.

**TYPE**

Displays the contents of a text file or files.

TYPE [drive:][path]filename

**VER**

Displays the Windows version.

VER

**VOL**

Displays the disk volume label and serial number, if they exist.

VOL [drive:]

**XCOPY**

Copies files and directory trees.

XCOPY source [destination] [/A | /M] [/D[:date]] [/P] [/S [/E]] [/V] [/W] [/C] [/I] [/-I] [/Q] [/F] [/L] [/G] [/H] [/R] [/T] [/U] [/K] [/N] [/O] [/X] [/Y] [/-Y] [/Z] [/B] [/J] [/EXCLUDE:file1[+file2][+file3]...] [/COMPRESS]

source Specifies the file(s) to copy.

destination Specifies the location and/or name of new files.

/A Copies only files with the archive attribute set, doesn't change the attribute.

/M Copies only files with the archive attribute set, turns off the archive attribute.

/D:m-d-y Copies files changed on or after the specified date.

If no date is given, copies only those files whose source time is newer than the destination time.

/EXCLUDE:file1[+file2][+file3]...

Specifies a list of files containing strings. Each string should be in a separate line in the files. When any of the strings match any part of the absolute path of the file to be copied, that file will be excluded from being copied. For example, specifying a string like \obj\ or .obj will exclude all files underneath the directory obj or all files with the .obj extension respectively.

/P Prompts you before creating each destination file.

/S Copies directories and subdirectories except empty ones.

/E Copies directories and subdirectories, including empty ones. Same as /S /E. May be used to modify /T.

/V Verifies the size of each new file.

/W Prompts you to press a key before copying.

/C Continues copying even if errors occur.

/I If destination does not exist and copying more than one file, assumes that destination must be a directory.

/-I If destination does not exist and copying a single specified file, assumes that destination must be a file.

/Q Does not display file names while copying.

/F Displays full source and destination file names while copying.

/L Displays files that would be copied.

/G Allows the copying of encrypted files to destination that does not support encryption.

/H Copies hidden and system files also.

/R Overwrites read-only files.

/T Creates directory structure, but does not copy files. Does not include empty directories or subdirectories. /T /E includes empty directories and subdirectories.

/U Copies only files that already exist in destination.

/K Copies attributes. Normal Xcopy will reset read-only attributes.

/N Copies using the generated short names.

/O Copies file ownership and ACL information.

/X Copies file audit settings (implies /O).

/Y Suppresses prompting to confirm you want to overwrite an existing destination file.

/-Y Causes prompting to confirm you want to overwrite an existing destination file.

/Z Copies networked files in restartable mode.

/B Copies the Symbolic Link itself versus the target of the link.

/J Copies using unbuffered I/O. Recommended for very large files.

/COMPRESS Request network compression during file transfer where applicable.

/SPARSE Preserves the sparse state when copying a sparse file.

The switch /Y may be preset in the COPYCMD environment variable. This may be overridden with /-Y on the command line.

### Batch Programs

1. Create a menu and display it. The user will select one of the menu option depending on the option selected a DOS command should be executed. Menu should contain the following:
   1. List all files.
   2. Create a directory.
   3. Display today’s date and time.
   4. Make abc.txt file as read-only
   5. Exit from menu

@echo off

:Menu

cls

echo Menu:

echo a. List all files.

echo b. Create a directory.

echo c. Display today's date and time.

echo d. Make abc.txt file as read-only.

echo e. Exit from menu

setlocal enabledelayedexpansion

set /p choice=Enter Your choice:

set dirname=

if "%choice%"=="a" (

dir

) else if "%choice%"=="b" (

set /p dirname=Enter directory name:

if not exist "!dirname!" (

mkdir "!dirname!"

echo Directory "!dirname!" created successfully.

) else (

echo Directory "!dirname!" already exists.

)

) else if "%choice%"=="c" (

echo %date% %time%

) else if "%choice%"=="d" (

attrib +r abc.txt

) else if "%choice%"=="e" (

exit

) else (

echo Invalid choice. Please try again.

)

endlocal

pause

goto Menu

1. Execute a C language program from the batch code.

@echo off

REM Check if GCC is installed or not installed

where gcc > nul 2>&1

if %errorlevel% equ 0 (

echo Compiling and running C program...

gcc demo.c -o demo.exe

demo.exe

) else (

echo GCC is not installed

)

1. Create an employee table in database and insert records in it. This should be done by calling the SQL script from the batch code.

@echo off

REM Check if SQL\*Plus is installed or not installed

where sqlplus> nul 2>&1

if %errorlevel% equ 0 (

echo Executing SQL script...

sqlplus darshil/darshil @script.sql

) else (

echo SQL\*Plus for Oracle is not installed.

)

1. Display the arithmetic table of ‘n’ (or 2). This value n should be supplied by the user from the command prompt.

@echo off

set /p num=Enter a number:

setlocal enabledelayedexpansion

for /l %%i in (1, 1, 10) do (

set /a result= !num! \* %%i

echo !num! \* %%i = !result!

)

endlocal

1. Call a batch program2 from another batch program1. Batch program2 should be creating a user MCA201 and also create its home path as C:\MCA2\MCA201.

userCreationCall.bat

@echo off

call userCreation.bat

userCreation.bat

@echo off

echo Creating user MCA201 and home path...

net user MCA201 /add

md C:\MCA2\MCA201

1. Copy all the .txt files from C:\D1 to three directories D:\D1, D:\D2, D:\D3.

@echo off

xcopy C:\D1\\*.txt D:\D1\

xcopy C:\D1\\*.txt D:\D2\

xcopy C:\D1\\*.txt D:\D3\

1. Your current directory is C:\D1, which contains some batch code which involves another batch file lying in D:\D1 from batch program1 go to path of batch program2, execute it and come back to original path.

customCall.bat

@echo off

cd /d D:\D1

call custom.bat

cd /d C:\D1

custom.bat

@echo off

echo This is custom bat file in different path

1. Execute a batch code from a C program.

#include <stdlib.h>

int main() {

system("demo.bat");

return 0;

}

demo.bat

@echo off

echo This is demo batch file running from c code.

**Linux Shell Scripts**

1. Display the menu for selecting from the list of Shell Assignments given to you.

The user should input the choice of selection from the list, and on the basis of their choice,a particular shell script should be executed .

clear

echo "1. File Listing"

echo "2. Create Directory"

echo "3. View File Content"

echo "4. Copy File"

echo "5. Delete Files"

read ch

case $ch in

1) ls;;

2) echo "Enter Directory Name:"

read d

mkdir $d;;

3) echo "Enter File Name:"

read f

cat $f;;

4) echo "Enter Source:"

read s

echo "Enter Destination:"

read d

cp $s $d;;

5) echo "Enter File Name:"

read fl

rm $fl;;

\*) echo "Invalid Choice."

esac

1. Display the current date & time.

#!/bin/bash

date

1. Display the current date in words.

#!/bin/bash

date +"%d %B %Y"

1. Display the greetings “Good Morning” or “Good Afternoon” or “Good Evening” according to the system time, when the user logs in.

#!/bin/bash

current\_hour=$(date +"%H")

if [ $current\_hour > 0 ] && [ $current\_hour < 12 ]; then

echo "Good Morning"

elif [ $current\_hour > 12 ] && [ $current\_hour < 18 ]; then

echo "Good Afternoon"

else

echo "Good Evening"

fi

1. Display the list of files in the current directory.

#!/bin/bash

ls

1. Display the list of files in the directory as given by the user input.

#!/bin/bash

echo "Enter directory path:"

read dir\_path

ls "$dir\_path"

1. Display the list of files in the
   1. ascending order of size of file

#!/bin/bash

ls -S

* 1. descending order of size of file

#!/bin/bash

ls -rS

* 1. ascending order of time of creation/modification

ls -t

1. Given the file name, check whether it is directory or an ordinary file or a system file or a link to a file.

#!/bin/bash

file\_name=$1

if [ -h "$file\_name" ]; then

echo "$file\_name is a symbolic link"

elif [ -f "$file\_name" ]; then

echo "$file\_name is a regular file"

elif [ -d "$file\_name" ]; then

echo "$file\_name is a directory"

elif [ -c "$file\_name" ]; then

echo "$file\_name is a character special file"

elif [ -b "$file\_name" ]; then

echo "$file\_name is a block special file"

elif [ -p "$file\_name" ]; then

echo "$file\_name is a named pipe (FIFO)"

elif [ -S "$file\_name" ]; then

echo "$file\_name is a socket"

else

echo "$file\_name is not found or of unknown type"

fi

1. If a file is a link to a file, then display whether it is hard or soft link.

#!/bin/bash

file\_name=$1

if [ -h "$file\_name" ]; then

if [ -e "$file\_name" ]; then

echo "$file\_name is a soft link"

else

echo "$file\_name is a broken soft link"

fi

elif [ -f "$file\_name" ]; then

echo "$file\_name is a regular file"

elif [ -d "$file\_name" ]; then

echo "$file\_name is a directory"

else

echo "$file\_name is not found or of unknown type"

fi

1. Display the name and size of file, which occupies
   1. maximum diskspace

#!/bin/bash

du -a | sort -nr | head -n 1

#!/bin/bash

ls -S | head -1

* 1. minimum diskspace

#!/bin/bash

du -a | sort -n | head -n 1

#!/bin/bash

ls -rS | head -1

1. Given the file name, display the list of files which are
   1. hard links to the given file.

#!/bin/bash

find -samefile demo.txt

* 1. soft links to the given file.

#!/bin/bash

find -lname demo.txt

1. Given the filename, the access previlege, the access privilege gruop add or remove the acccess previlege on the given file or for a given group.

#!/bin/bash

# To add permission for the user:

chmod u+x demo.txt

# To remove permission for the user:

chmod u-x demo.txt

# To add permission for the group:

chmod g+x demo.txt

# To remove permission for the group:

chmod g-x demo.txt

echo "Permission Changed Successfully."

1. Display the list of users who are currently logged in.

#!/bin/bash

who

1. Given the username, find out whether the user is currently logged in or out.

#!/bin/bash

read user

who | grep -w user

1. Given the username, display the time elapsed, since the given user is logged in.

#!/bin/bash

read user

who | grep -w user | awk '{print $4}'

1. Display the count of users who are currently logged in.

#!/bin/bash

who | wc -l

1. Display the 3 users,
   1. who have logged in first

#!/bin/bash

who | sort -k4 | head -n 3

* 1. who have logged in last

#!/bin/bash

who | sort -k4 | tail -n 3

1. Given the shell script filename, make the shell script available to all the users of the system.

#!/bin/bash

read file

sudo cp file /usr/local/bin/

sudo chmod +x /usr/local/bin/$file