INDEX

“**Command Line Mastery”**

Project Title:

“Command Line Mastery: A Comprehensive Internship Program on Windows Command Prompt”.

Objective of project:

The primary goal of this project is to provide interns with a thorough understanding and practical expertise in using the Micros; Command Prompt. The project aims to equip them with the skills necessary to perform various tasks in a Windows environment efficiently, from basic file management to advanced system diagnos8cs and network troubleshooting.

The command prompt, also known as the command line interface (CLI) or terminal, is a text-based interface in which the user interacts with the computer by typing commands. Here are some common objectives or tasks related to the command prompt:

- Navigation
- File and Directory Management
- File Display and Text Editing
- System Information
- Network Commands
- Scripting and Batch Processing
- Permissions and Security
- System Maintenance
- Environment Variables
- Remote Access
- Help and Documentation

The commands and their usage can vary between operating systems (e.g., Windows, Linux, macOS), so it's essential to be aware of the specific commands for the system you are working on. Additionally, always exercise caution when using commands, especially those that can modify or delete files, to avoid unintended consequences.

INTRODUCTION

1. Introduction to Command Prompt

The Command Prompt in Windows is like a special chat with your computer where you type commands instead of clicking on icons. Most people use the regular way with icons, but the Command Prompt lets you do things using only text. It's like giving secret instructions to your computer in a special language. You can do basic tasks like moving around your computer's folders, copying files, and creating new folders. But it's also useful for more advanced tasks, especially when you need to fix something or change settings that you can't easily reach through regular buttons and menus. The Command Prompt is like having a special tool to do important work behind the scenes on your computer. It's a bit like being a computer wizard! A command-line interface (CLI) is another way of talking to your computer, but instead of clicking buttons, you type specific instructions to make the computer do things like running programs, managing files, or changing settings. It's like sending your computer a message in a language it understands.

You can use the Command Prompt to do basic tasks like moving around your computer's folders, copying files, and creating new folders. But it's also handy for more advanced tasks, especially ones that involve controlling and managing your computer at a deeper level. For example, if you need to fix something or change settings that are not easily accessible through the regular buttons and menus, the Command Prompt is where you go. It's like having a special tool to do some important behind-the-scenes work on your computer. A command-line interface (CLI) is like talking to your computer by typing commands in a text-based area. Instead of clicking icons or buttons, you type specific instructions to make the computer do things like running programs, managing files, or changing settings. It's a way to communicate with your computer using text commands.

Fig.1.1. Command prompt window

The command prompt is valuable for users who want to delve deeper into the functionality of their operating system, automate tasks, or troubleshoot issues efficiently. The command prompt, often referred to as the command line interface (CLI), is a method of

interacting with a computer's operating system through text-based commands. Instead of relying on graphical elements like icons and buttons, users input commands directly, allowing for efficient and precise execution of various tasks. This interface is present in the operating systems of many computers, such as Windows, Linux, and macOS, although the specific commands may differ between these systems. The command prompt provides users with a way to communicate with the computer by typing text commands. It serves as an alternative to graphical user interfaces (GUIs) commonly found in modern operating systems. This text-based approach offers a direct and powerful means of performing tasks, ranging from basic file management to advanced system configurations. Whether accessing the command prompt on Windows, Linux, or macOS, users can navigate through directories, create, delete, and manipulate files, and retrieve system information. It is a versatile tool that empowers users to carry out a wide array of operations, making it particularly valuable for system administrators, developers, and users who seek more control over their computing environment.

Following are the steps to open windows command prompt(cmd):

>Search for “cmd” in search bar of windows

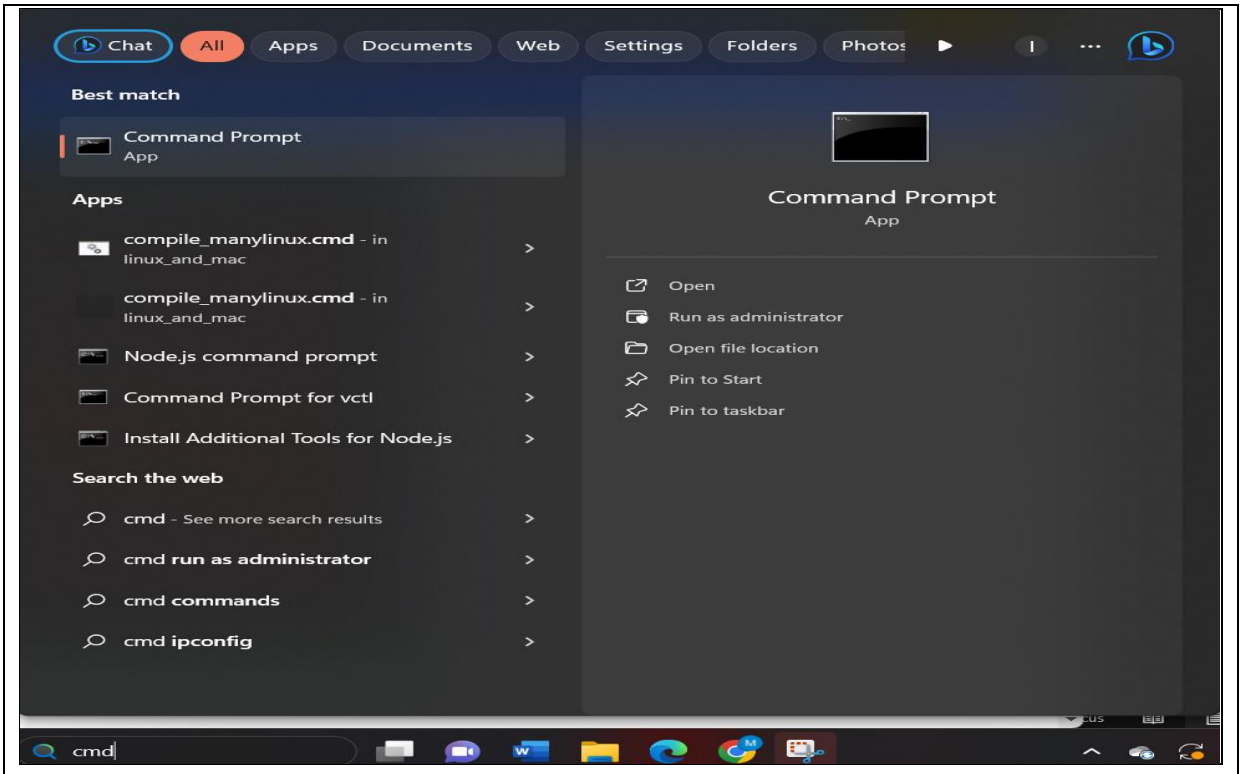
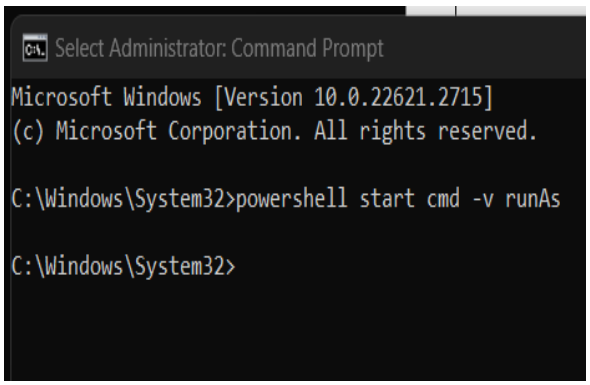


Fig.1.2

>Click on “Run as administrative “option to open promt window.

> powershell start cmd -v runAs This command opens another command prompt window as an administrator.

>Run “”driverquery command will list all install drivers.



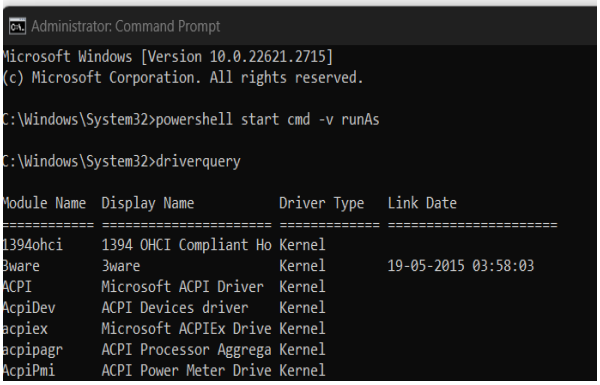


Fig.1.3 **Fig.1.4**

>Command “systeminfo” shows all details about your system.

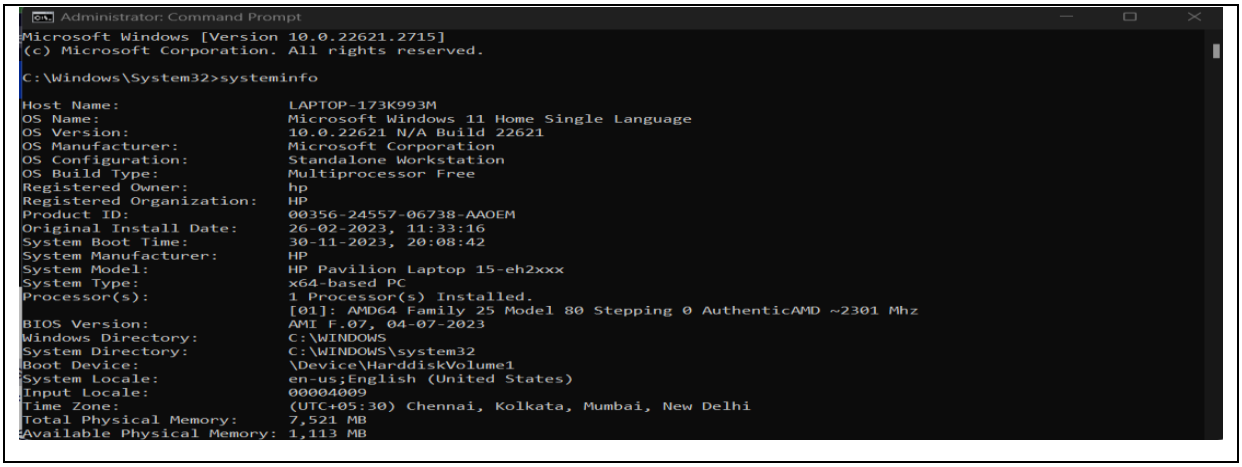
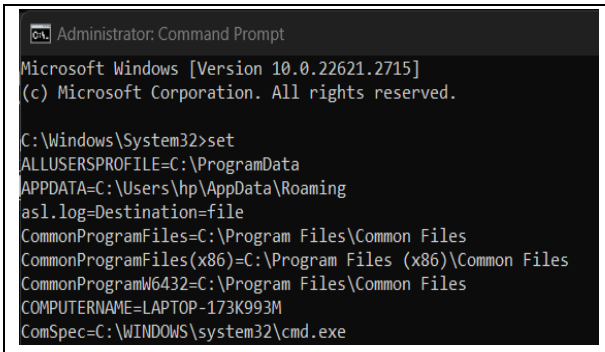


Fig.1.5

>command “ set” Shows your system Environment Variables

> “clip” – Copies an Item to the Clipboard and “ dir | clip” copies all the content of the present working directory to the clipboard



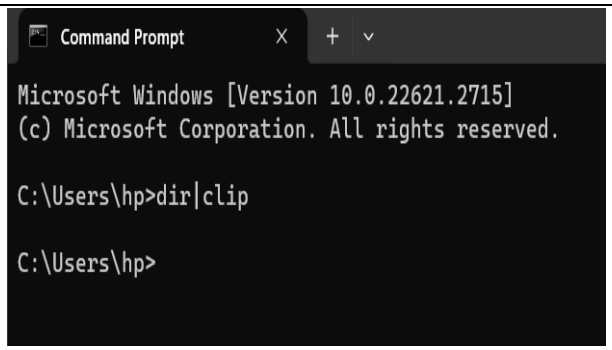


Fig.1.6 **Fig.1.7**

> **“**title” Changes the Command Prompt Window Title Using the Format

title window-title-name

> **“**fc**”** use to Compares Two Similar Files on windows

“**Command Line Mastery”**

If you are a programmer you want to quickly see what differs between two files, you can use this command and then the full path to the two files. For example fc “file-1-path” “file-2-path”.

>Command “cipher” can use to wipe the drive clean and encrypt such files.

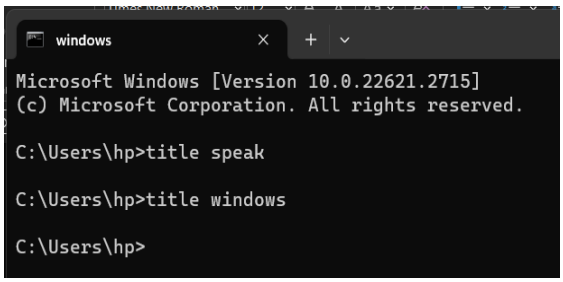


Fig.1.8

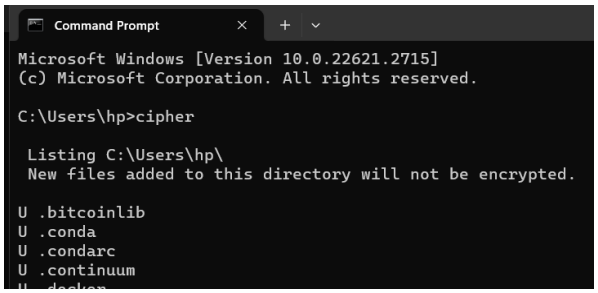


Fig.1.9

> “netstat -an” Shows Open Ports, their IP Addresses and States

> “ping” command Shows a Website IP Address.

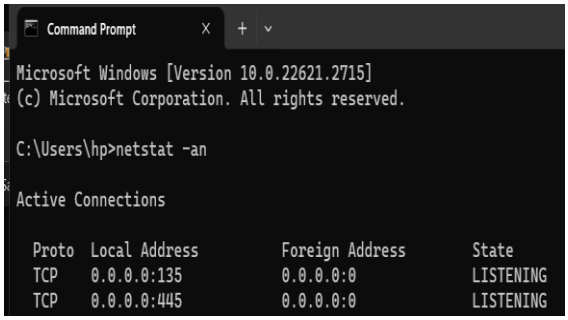


Fig.1.10

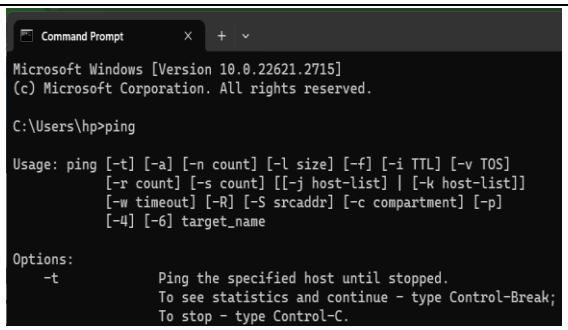


Fig.1.11

>”color attr” command Changes the Text Color of the Command Prompt

>”color 5 “command Changes the color of the terminal to purple.

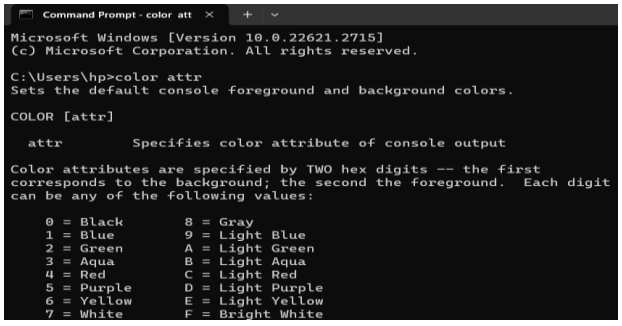


Fig.1.12

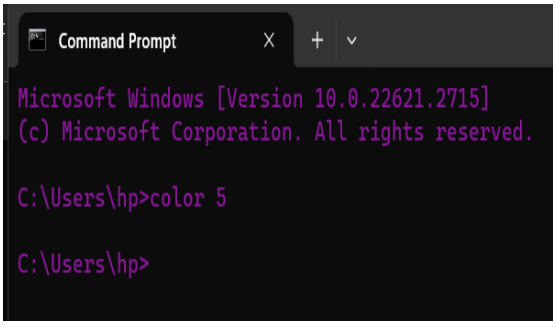


Fig.1.13

>”ipconfig” command Shows Information about PC IP Addresses and Connections.
>”sfc” System File Checker command scans your computer for corrupt files and repairs them.

> ”powercfg “ Controls Configurable Power Settings, You can use this command with its several extensions to show information about the power state of your system. You can enter “powercfg help” to show those extensions.

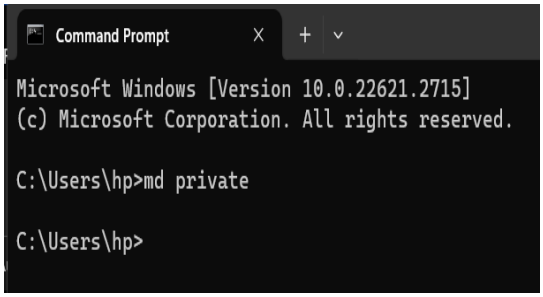
1.1 Navigating the Command Prompt interface:

Navigating the command prompt interface involves using commands to move around the file system, explore directories, and interact with files. Below are key commands and concepts for effective navigation in the command prompt interface:

Making directory:

Command: md (making directory)

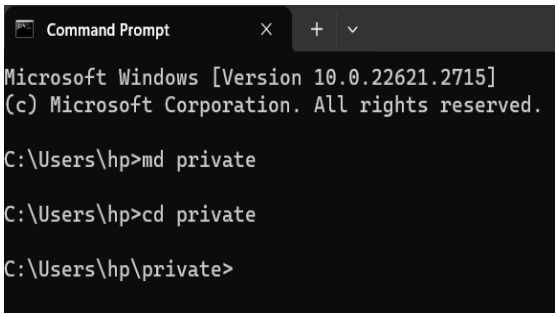
Usage: make a new directory



Changing Directory:

Command: cd (change directory)

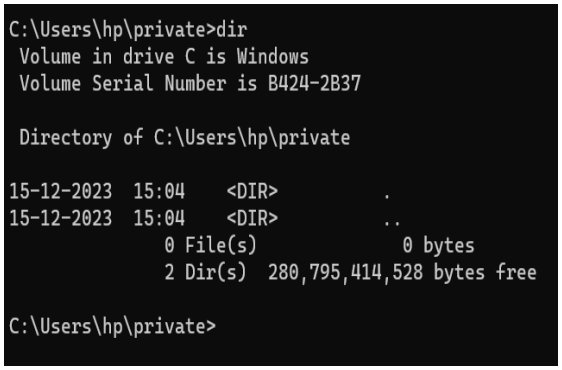
Usage: To move to a specific directory, use cd [directory_path] To move up one level, use cd



Listing Contents:

Commands: dir (Windows) or ls (Unix-like systems)

Usage: To list files and directories in the current directory, use dir (Windows) or ls (Unix).
To list with detailed information, use dir /A (Windows) or ls -l (Unix).



Path Structure:

Paths can be absolute or relative.

Absolute Path: Specifies the full path from the root directory

Relative Path: Specifies the path relative to the current directory

Autocomplete:

Pressing the Tab key can autocomplete commands or directory/file names, reducing the need for typing the entire name.

Wildcard Characters:

Use wildcard characters to represent multiple files or directories.

* represents any characters.

? represents a single character.

Path Shortcuts (Windows):

Use special directory names as shortcuts on Windows,%USERPROFILE% represents the user's profile directory, %HOMEPATH% represents the user's home directory.

Copying and Pasting:

On some systems, you can copy text from elsewhere and right-click to paste it into the command prompt.

Drive Switching (Windows):

On Windows, switch between drives using the drive letter followed by a colon.

Example: D: switches to the D drive.

History Navigation:

Use the up and down arrow keys to navigate through previously entered commands.

Creating Directories:

Command: mkdir (make directory)

Usage: mkdir [directory_name] creates a new directory.

Removing Directories:

Command: rmdir or rd (remove directory)

Usage: rmdir [directory_name] removes a directory.
These commands will help you efficiently navigate the command prompt interface, making it a powerful tool for managing your computer's file system.

1.2 Basic commands:

1. dir (Directory Listing):
Purpose: Lists the files and directories in the current directory.
Syntax: dir [options] [path]

2. cd (Change Directory):
Purpose: Changes the current working directory.
Syntax: cd [directory_path]

3. mkdir (Make Directory):
Purpose: Creates a new directory.
Syntax: mkdir [directory_name]

4. rmdir or rd (Remove Directory):
Purpose: Deletes a directory.
Syntax: rmdir [directory_name] or rd [directory_name]

5. copy (Copy Files):
Purpose: Copies one or more files to another location.
Syntax: copy [source] [destination]

6. move (Move or Rename Files):
Purpose: Moves files or renames them.
Syntax: move [source] [destination]

7. del (Delete Files):
Purpose: Deletes one or more files.
Syntax: del [file_name]

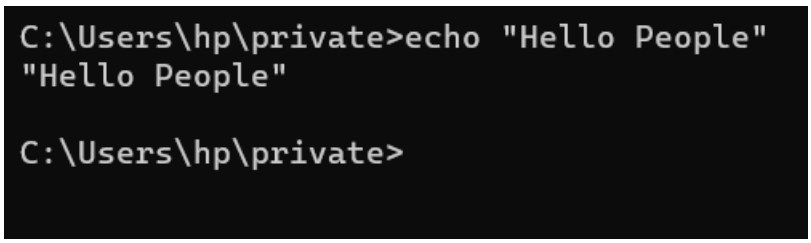
8. ren (Rename Files):Renames a file or files

Syntax: ren filename.extension new-name.extension

9. echo (Display Message or Redirect Output):

Purpose: Displays messages or redirects output to a file .Shows Custom Messages or Messages from a Script or File

Syntax: echo [message] or echo [message] > [file]

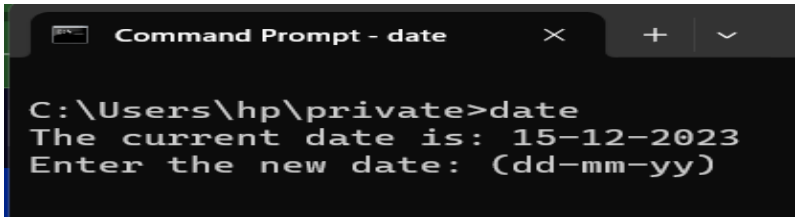


10.Time(display time)

Purpose: Shows and changes the current time.

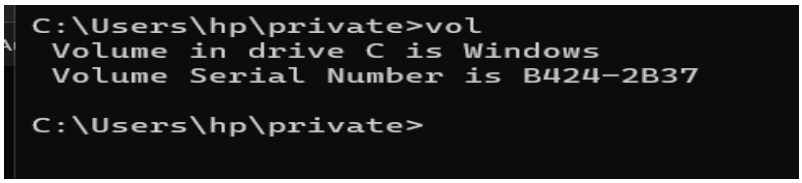
11. Date (shows date)

Purpose: Shows and changes the current date



12. vol

Purpose: Shows the Serial Number and Label Info of the Current Drive.



13. dism

Purpose: Runs the Deployment Image Service Management Tool.

14.mkdir

Purpose: Creates a folder

15. rmdir

Purpose: use to Deletes a folder.

16.more

Purpose: Shows More Information or the Content of a File

Syntax: more (file name)

17.move

Purpose: use to Moves a file or folder to a Specified Folder.

Syntax: move (file name)(folder name)

18.cls (Clear the entire window)

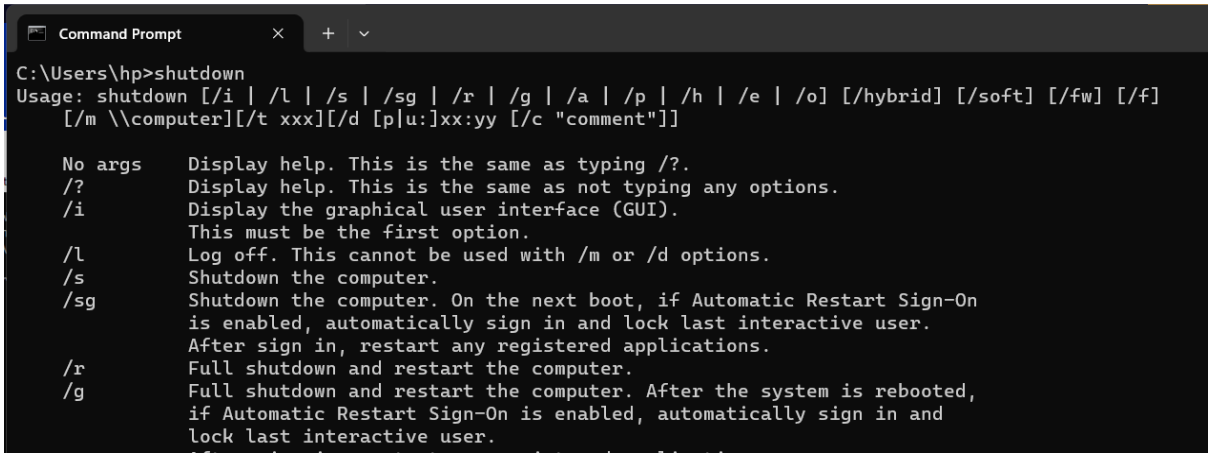
Purpose: clear the command line.

19. exit

Purpose: Closes the Command Line

20.shutdown

Purpose: Shuts down, Restarts, Hibernates, Sleeps the Computer, You can shut down, restart, and sleep your PC from the command line. Enter shutdown in the command line so you can see the extensions you can use to perform the actions. For example, shutdown /r will restart your computer.



These commands form the basic building blocks for file and directory manipulation in the command prompt. Understanding and practicing these commands will empower you to navigate and manage your file system efficiently from the command line**.**

2. File and Directory Management

File and directory management using the Command Prompt. File and directory management involves creating, moving, copying, and organizing files and folders. Here are the key commands for these tasks:

Commands	Purpose	Example
mkdir	Creates a new directory	mkdir new folder
cd	Changes the current working directory	cd Documents: Moves into the "Documents" directory. cd ..: Moves up one directory level.
dir (on Windows) ls (on Unix-like systems)	Lists files and directories in the current directory.	dir: Lists files and directories. dir /a: Lists hidden files as well. ls -l: Lists with detailed information.
copy	Copies one or more files to another location.	copy file.txt C:\Backup
move	Moving or renaming files	move file.txt C:\Documents: Moves "file.txt" to the "Documents" directory. move oldfile.txt newfile.txt: Renames "oldfile.txt" to "newfile.txt."
find	It is useful when you have a rough idea about file location.	$ find /home/user -name "*.html"
type	Type is a built-in command which displays the contents of a text file. Use the type command to view a text file without modifying it.	type[file_name]
del	Deletes one or more files	del unwanted.txt
rmdir or rd	Removes and empty directory	rmdir OldFolder
Displaying File Content (type or more):	Displays the content of a text file.	type readme.txt
Renaming Files (ren)	ren [old_name] [new_name]	ren oldfile.txt newfile.txt
Searching for Files (find or dir /s)	Searches for files in a directory (including subdirectories).	find C:\Users -name *.txt or dir /s *.txt

Disk Usage (dir /s or du)	Shows the disk usage of files and directories.	dir /s or du -h Documents
Free Disk Space (df)	Displays free disk space.	Df

These commands form the basic toolkit for managing files and directories in the Command Prompt. Understanding and using these commands efficiently will empower you to organize and manipulate your file system effectively from the command line. Always exercise caution, especially when using commands that involve deletion or modification of data.

These commands are essential for managing files and directories in the command prompt. They provide a powerful set of tools for organizing, manipulating, and interacting with the file system from a text-based interface. Understanding these commands enables users to efficiently perform common file and directory operations.

3. Advanced File Operations

Advanced file operations in the Command Prompt involve more complex tasks such as searching for files with specific criteria, working with file permissions, and using advanced commands for file manipulation.

1. Finding Files with Specific Criteria (find):

Command: findstr (Windows) or grep (Unix-like systems)

Purpose: Searches for lines in files that match a specified pattern.

findstr /s /i "search_pattern" *.txt

This command searches for "search_pattern" in all text files (*.txt) in the current directory and its subdirectories (/s), and the /i flag makes the search case-insensitive.

2. Recursive File and Directory Listing (dir /s):

Command: dir /s

Purpose: Lists files and directories recursively.

dir /s /b /a:-d: This command lists all directories (/a:-d) in the current directory and its subdirectories (/s) in a bare format (/b).

3. Using Wildcards in File Operations (* and?):

Purpose: Wildcards are used as placeholders in file operations.

copy *.txt C:\Backup: This command copies all files with the ".txt" extension to the "Backup" directory.

del file?.txt: This command deletes files like "file1.txt," "fileA.txt," etc., where ? matches any single character.

4. File and Directory Permissions (icacls):

Command: icacls (Windows)

Purpose: Displays or modifies access control lists (ACLs) for files and directories.

icacls filename.txt /grant:r User:(D,RX): This command grants read and execute permissions to the specified user (User) for the file "filename.txt."

5. Redirecting Output to a File (> and >>):

Purpose: Redirects command output to a file.

dir > directory_list.txt: This command lists the current directory contents and writes the output to a file named "directory_list.txt."

echo Hello >> greetings.txt: This command appends the text "Hello" to an existing file named "greetings.txt."

6. Creating Symbolic Links (mklink):

Command: mklink (Windows)

Purpose: Creates a symbolic link to a file or directory.

mklink link_name target:This command creates a symbolic link named "link_name" that points to the specified "target" file or directory.

7. Batch File Operations with for Loop:

Command: for (Windows)

Purpose: Performs batch file operations.

for %i in (*.txt) do copy %i C:\Backup:This command copies all text files in the current directory to the "Backup" directory using a for loop.

These advanced commands and techniques provide more sophisticated ways to manipulate files and directories in the Command Prompt. Understanding and using these operations can enhance your efficiency and flexibility in managing files and directories from the command line.

3.1 Batch file scripting basics

A batch file is a script or a sequence of commands that are stored in a plain text file with a ".bat" or ".cmd" extension. These files contain a series of commands that the Command Prompt can execute sequentially. file with a ".bat" extension on a Windows computer, it means it's a batch file. A batch file is like a set of instructions that tell the computer to do specific things in order. It's a way of automating tasks or running a sequence of commands on your computer. The ".bat" extension simply indicates that it's a batch file recognized by the Windows operating system.

Creating a Batch File:

Text Editor: You can create a batch file using a simple text editor like Notepad.

File Extension: Save the file with a ".bat" or ".cmd" extension. For example, myscript.bat.

Writing Batch Commands:

Command Lines: Each line in a batch file represents a command that the Command Prompt will execute.

Comments: Add comments using REM at the beginning of a line to explain the purpose of a command.

Commands: Use regular Command Prompt commands in the batch file. For example:

Variables and Parameters:

Variables (SET):You can set and use variables.

“**Command Line Mastery”**

Command-Line Parameters:You can use %1, %2, etc., to represent parameters passed to the batch file.

Control Flow:

Conditional Statements (IF):Use IF statements for conditional execution.

Running the Batch File:

Double-Click: Double-clicking a batch file executes it in the Command Prompt.

Command Prompt:You can run a batch file from the Command Prompt by typing its name.

Example Batch File:Here's a simple example of a batch file that echoes a message:

REM: Indicates a comment.

ECHO: Displays a message.

PAUSE: Pauses execution to keep the Command Prompt window open.

Some popular basic commands includes as following:

Commands	**Explanation**
echo	Display message on the screen
title	Changes the title text at the top of the command window.
rem	Used for comments or explanations in the code, but doesn't affect the actual execution.
cls	Clears everything from the command prompt screen.
pause	Pauses the execution of the batch file, giving you time to decide what to do next.
Start "" [website]	Opens a website using the default web browser.
ipconfig	Shows information about your computer's network, like IP addresses.
ping	Tests the connection to a specific IP address by sending and receiving data.

Steps to creating a batch files:

>Open a simple text editor like Notepad on your computer.

>Start with @echo off:

Type @echo off as the first line.

>On new lines, add these commands:
title [title of the batch script]: Gives your batch file a title.
echo [first line]: Displays a message.
pause: Pauses execution, giving you time to read the message.
@echo off
>echo Congratulations and welcome to your first batch file!
>pause
>Click on "File" and select "Save As."
Save the batch file with a .bat, .btm, or .cmd extension.
Put the entire file name in double quotes (e.g., "MyBatchFile.bat").
>Double-click the saved batch file to run and execute the commands.
Edit the Batch File:
>Right-click the batch file, select "Edit."
>The file will open in Notepad (or another text editor).
Avoid spaces in the file name to prevent issues. Choose a unique name for your batch file. Double-clicking the batch file runs it. Right-click and select "Edit" to make changes. These steps help you create a simple batch file that executes commands when you run it. It's like a little program you can make to do specific things on your computer.

3.2 Automating tasks using batch scripts

Batch files are handy for automating repetitive tasks, and by combining simple commands, comments, and control structures, you can create useful scripts to streamline your workflow in the Command Prompt.

To create a basic batch file use these steps:
>Open Start.
>Search for Notepad and click the top result to open the text editor.
>Type the following lines in the text file to create a batch file:
@ECHO OFF
ECHO Hello World! Your first batch file was printed on the screen successfully.
PAUSE
@ECHO OFF :Shows the message on a clean line disabling the display prompt. Usually, this line goes at the beginning of the file. (You can use the command without the "@" symbol, but it's recommended to include it to show a cleaner return.)
ECHO :The command prints the text after the space on the screen.

PAUSE: Allows the window to stay open after the command has been executed. Otherwise, the window will close automatically as soon as the script finishes executing. You can use this command at the end of the script or after a specific command when running multiple tasks and wanting to pause between each line.

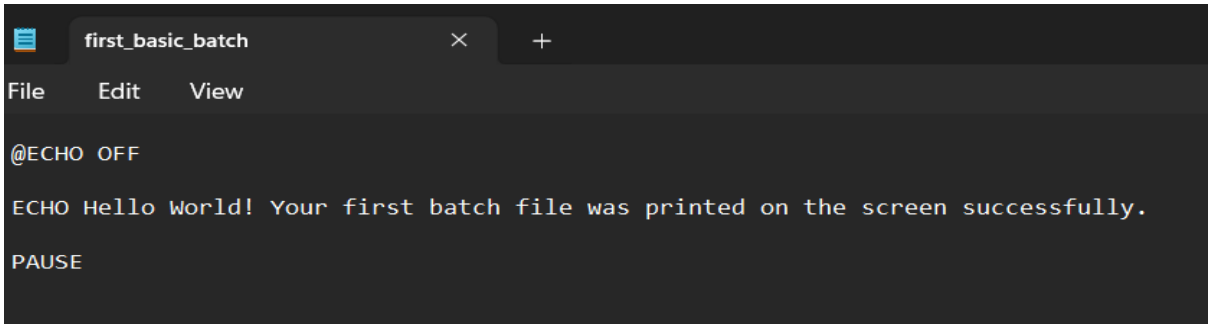


Fig.3.1

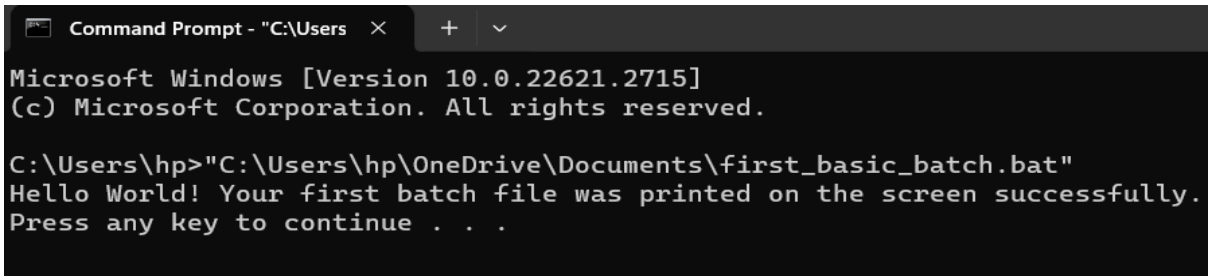


Fig.3.2

3.3 Introduction to xcopy and file backup strategies

xcopy is a command-line utility in Windows used for copying multiple files or entire directory trees from one location to another. It provides more options and flexibility compared to the basic copy command. This command is available from within the Command Prompt in all Windows operating systems including Windows 11. You can also access the command in MS-DOS as a DOS command.

Syntax: xcopy source destination [/switches]

Common Switches:

- /s: Copies directories and subdirectories, but not empty ones.
- /e: Copies directories and subdirectories, including empty ones.
- /y: Suppresses prompts to confirm you want to overwrite an existing destination file.
- /d:mm-dd-yyyy: Copies files changed on or after the specified date.

File Backup Strategies using xcopy:

- Basic Backup:

  Command:

xcopy C:\SourceFolder*.* D:\Backup\ /s /e /y

Explanation: This copies all files and subdirectories from "C:\SourceFolder" to "D:\Backup," including empty directories, without prompting for confirmation.

- Incremental Backup:

  Command:

  xcopy C:\SourceFolder*.* D:\Backup\ /s /e /y /d:mm-dd-yyyy

  Explanation: This copies only files changed on or after a specified date, useful for incremental backups.

- Mirror Backup:

  Command:

  xcopy C:\SourceFolder*.* D:\Backup\ /s /e /y /mir

  Explanation: The /mir switch mirrors the source directory to the destination, deleting files in the destination that no longer exist in the source.

- Logging Backup:

  Command:

  xcopy C:\SourceFolder*.* D:\Backup\ /s /e /y /l > backup_log.txt

  Explanation: The /l switch performs a dry run (lists files without copying), and > backup_log.txt redirects the output to a log file for review.

  Explanation: This script backs up files from "C:\SourceFolder" to "D:\Backup," copying only files modified on or after the specified date without prompting for confirmation.

Adjust the source and destination paths based on your setup.Regularly test your backup script to ensure it captures the desired files.Combine xcopy with other commands or scripts for comprehensive backup strategies.

Using xcopy in batch scripts provides a versatile and efficient way to handle file backup tasks in the Windows command line.

Xcopy Command Syntax:

- source :This defines the files or top level folder that you want to copy from. The source is the only required parameter. Use quotes around source if it contains spaces.
- Destination: This option specifies the location where the source files or folders should be copied to. If no destination is listed, the files or folders will be copied to the same folder you run the xcopy command from. Use quotes around destination if it contains spaces.

- /a: Using this option will only copy archive files found in source. You can not use /a and /m together.
- /b: Use this option to copy the symbolic link itself instead of the link target. This option was first available in Windows Vista.
- /c: This option forces xcopy to continue even if it encounters an error.
- /d [:date]: Use the command with /d option and a specific date, in MM-DD-YYYY format, to copy files changed on or after that date. You can also use this option without specifying a specific date to copy only those files in source that are newer than the same files that already exist in destination. This is helpful when using xcopy to perform regular file backups.
- /e: When used alone or with /s, this option is the same as /s but will also create empty folders in destination that were also empty in source. The /e option can also be used together with the /t option to include empty directories and subdirectories found in source in the directory structure created in destination.
- /f: This option will display the full path and file name of both the source and destination files being copied.
- /g: Using xcopy with this option allows you to copy encrypted files in source to a destination that does not support encryption. This option will not work when copying files from an EFS encrypted drive to a non-EFS encrypted drive.
- /h: The command doesn't copy hidden files or system files by default but will when using this option.
- /i: Use the /i option to force xcopy to assume that destination is a directory. If you don't use this option, and you're copying from source that is a directory or group of files and copying to destination that doesn't exist, the xcopy command will prompt you enter whether destination is a file or directory.
- /j: This option copies files without buffering, a feature useful for very big files. This option was first available in Windows 7.
- /k: Use this option when copying read-only files to retain that file attribute in destination.
- /l: Use this option to show a list of the files and folders to be copied... but no copying is actually done. The /l option is useful if you're building a complicated command with several options and you'd like to see how it would function hypothetically.

- /m: This option is identical to the /a option but xcopy will turn off the archive attribute after copying the file. You can not use /m and /a together.
- /n: This option creates files and folders in destination using short file names. This option is only useful when you're using the command to copy files to a destination that exists on an drive formatted to an older file system like FAT that does not support long file names.
- /o: Retains ownership and Access Control List (ACL) information in the files written in destination.
- /p: When using this option, you'll be prompted before the creation of each file in destination.
- /q: A kind of opposite of the /f option, the /q switch will put xcopy into "quiet" mode, skipping the on-screen display of each file being copied.
- /r: Use this option to overwrite read-only files in destination. If you don't use this option when you want to overwrite a read-only file in destination, you'll be prompted with an "Access denied" message and the command will stop running.
- /s: Use this option to copy directories, subdirectories, and the files contained within them, in addition to the files in the root of source. Empty folders will not be recreated.
- /t: This option forces the xcopy command to create a directory structure in destination but not to copy any of the files. In other words, the folders and subfolders found in source will be created but there we be no files. Empty folders will not be created.
- /u: This option will only copy files in source that are already in destination.
- /v: This option verifies each file as it's written, based on its size, to make sure they're identical. Verification was built in to the command beginning in Windows XP, so this option does nothing in later versions of Windows and is only included for compatibility with older MS-DOS files.
- /w: Use the /w option to present a "Press any key when ready to being copying file(s)" message. The command will begin copying files as instructed after you confirm with a key press. This option is not the same as the /p option which asks for verification before each file copy.
- /x:: This option copies file audit settings and System Access Control List (SACL) information. You imply /o when you use the /x option.

- /y : Use this option to stop the command from prompting you about overwriting files from source that already exist in destination.
- /-y: Use this option to force the command to prompt you about overwriting files. This might seem like a strange option to exist since this is the default behavior of xcopy but the /y option may be preset in the COPYCMD environment variable on some computers, making this option necessary.
- /z: This option allows the xcopy command to safely stop copying files when a network connection is lost and then resume copying from where it left off once the connection is reestablished. This option also shows the percentage copied for each file during the copy process.
- /exclude:file1[+file2][+file3]... :This option allows you to specify one or more file names containing a list of search strings you want the command to use to determine files and/or folders to skip when copying.
- /? : Use the help switch with xcopy to show detailed help about the command. Executing xcopy /? is the same as using the help command to execute help xcopy.

The Windows command line provides a way to back up your files and folders using a command called xcopy. This command comes with different options, known as switches, that let you specify conditions for copying files.

What is xcopy: In simple words x copy is like a command that helps you copy stuff from one place to another on your computer.

Why we use it: You can use xcopy to make a copy of your important files and folders. It's like creating a safety net for your stuff.

Where to Copy: You can copy your files to a USB drive or an external hard disk. These are like extra storage spaces you can carry around.

Different Conditions with Switches: Xcopy has special rules, called switches, that you can use to tell it exactly what kind of files to copy.

For example, /s means copy subdirectories, /e means copy everything (including empty folders), and /y means don't ask, just do it.

Different Conditions with Switches: Xcopy has special rules, called switches, that you can use to tell it exactly what kind of files to copy.

For example, /s means copy subdirectories, /e means copy everything (including empty folders), and /y means don't ask, just do it.

How to Do It: You open the command prompt, type xcopy, say where your files are, where you want to copy them, and add any special rules you want. If you have important photos in a folder on your desktop and want to copy them to your USB drive, you type something like:

“xcopy C:\Users\YourName\Desktop\Photos*.* E:\Backup\ /s /y”

This says, "Hey computer, copy all the photo files from my desktop to my USB drive, including subfolders, and don't bother asking me."

Why Backup: Think of it like making a copy of your favourite video game progress. If something happens to the original, you still have your backup.

Remember, xcopy is like a digital copy-paste command, but with extra powers to handle different situations when you're copying your important stuff around.

4. System Diagnostics and Management

System diagnostic and management are like the doctors for your computer. They use tools and processes to check, analyze, and make sure everything in your computer is working well. This includes checking the physical parts like the heart (hardware components), the brain (software applications), and the overall health of the whole body (the system). The main aim is to find any problems, fix them, and make sure your computer works smoothly and efficiently. It's like giving your computer a regular health check to keep it in good shape.

4.1 Key components are:

Key components	Purpose	Tools
Hardware Diagnostics	Identifying issues with physical components like RAM, hard drives, CPU, etc.	Built-in hardware diagnostics, third-party tools, and diagnostic software.
Software Diagnostics	Analyzing, software performance, detecting errors, and ensuring compatibility.	System logs, event viewers, antivirus scans, and application-specific diagnostic tools.
Network Diagnostics	Assessing network performance, identifying connectivity issues.	Network monitoring tools, command-line utilities (e.g., ping, traceroute), and network diagnostic commands.
Performance Monitoring	Continuous tracking of system resources, such as CPU usage, memory, and disk space.	Task Manager (Windows), Activity Monitor (macOS), and various third-party performance monitoring tools.
System Logs and Event Management	Capturing system events and errors for analysis and troubleshooting.	Windows Event Viewer, syslog (Unix/Linux), and centralized logging solutions.

4.2 Diagnostic and Management Tasks

a. Fault Detection:

Description: Identifying hardware or software components that are not functioning correctly.

Tasks: Running hardware tests, checking error logs, and monitoring system alerts.

b. Performance Optimization:

Description: Enhancing system efficiency by managing resource usage.

Tasks: Analyzing performance metrics, adjusting settings, and optimizing software configurations.

c. Security Monitoring:

Description: Identifying and addressing security vulnerabilities and threats.

Tasks: Regular security scans, monitoring firewall logs, and updating antivirus definitions.

d. Backup and Recovery:

Description: Ensuring data safety through regular backups and having recovery plans.

Tasks: Scheduled backups, testing restoration processes, and implementing disaster recovery strategies.

e. Updates and Patch Management:

Description: Keeping the system up-to-date with the latest software patches.

Tasks: Applying operating system updates, software patches, and firmware upgrades.

f. Resource Allocation:

Description: Optimizing the distribution of system resources to different applications and processes.

Tasks: Adjusting priority settings, managing background processes, and optimizing virtual memory.

4.3 Tools for System Diagnostic and Management

a. Operating System Utilities:

Examples: Task Manager (Windows), Activity Monitor (macOS), System Monitor (Linux).

b. Third-Party Diagnostic Software:

Examples: Speccy, HWiNFO, CPU-Z for hardware diagnostics; Wireshark for network diagnostics; Malwarebytes for malware detection.

c. Centralized Monitoring Solutions:

Examples: Nagios, Zabbix, and SolarWinds for monitoring multiple systems centrally.

d. Command-Line Tools:

Examples: Command Prompt and PowerShell (Windows), Terminal (macOS/Linux) for running diagnostic commands.

Effective system diagnostic and management practices are crucial for maintaining a stable and secure computing environment. Regular monitoring, proactive issue resolution, and strategic management contribute to the overall health and performance of a computer system. A well-managed system enhances productivity, reduces downtime, and ensures the security of sensitive data.

4.4 Using chkdsk, diskpart, and systeminfo.

1. chkdsk (Check Disk): System Diagnostics

Purpose: Checks the health of your computer's hard drive and file system.

Usage:

- Open Command Prompt.
- Type chkdsk C: and press Enter.
- If issues are found, it fixes them.

If your computer is acting slow or showing disk-related errors. Run chkdsk to identify and repair issues on the hard drive, ensuring smooth operation.

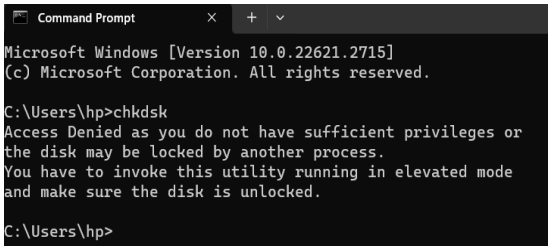


Fig.4.1

2. diskpart: System Diagnostics:

Purpose: Manages disk partitions and volumes on your computer.

Usage:

- Open Command Prompt.
- Type diskpart and press Enter.
- Use commands like list disk, select disk, and create partition.

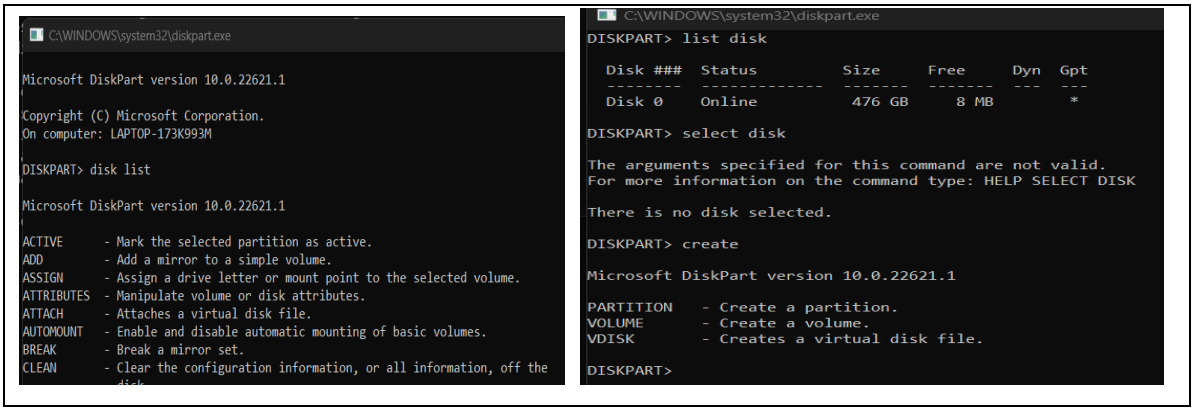


Fig.4.2 **Fig.4.3**

Need to organize or format storage space. Utilize diskpart to create, format, or resize partitions, optimizing storage management.

3. Systeminfo (System Information): System Diagnostics:

Purpose: Provides detailed information about the computer's hardware and software.

Offers insights into the system's configuration for troubleshooting and optimization.

Use Case:

Running systeminfo helps in understanding the computer's specifications, aiding in hardware upgrades, compatibility checks, or diagnosing performance issues.

- chkdsk: Checks and fixes hard drive issues, like a computer doctor for your storage.
- diskpart: Manages how your computer organizes its storage space, like a storage organizer.
- systeminfo: Provides a detailed report about your computer's parts, helping you understand and manage its overall health.

These tools work together to diagnose problems, manage storage effectively, and keep your computer running smoothly. It's like a team of specialists ensuring your computer stays healthy and performs at its best.

4.5 Understanding system logs and troubleshooting common issues

Definition: System logs are records generated by the operating system and various software components, capturing events, errors, and activities on a computer.

There are three types of System Logs:

- Event Logs: Windows Event Viewer, syslog (Linux/Unix), system logs (macOS).
- Application Logs: Specific to software applications.
- Security Logs: Records security-related events.

Common Issues Recorded in System Logs:

"Command Line Mastery"

a. Application Crashes:

Log Entry: Error or warning messages related to the crashed application.

Troubleshooting: Investigate the application-specific logs for details on the error.

b. Hardware Failures:

Log Entry: Disk errors, memory issues, or hardware warnings.

Troubleshooting: Run diagnostics (e.g., chkdsk) and check hardware-specific logs.

c. Security Incidents:

Log Entry: Unusual login attempts, security breaches, or unauthorized access.

Troubleshooting: Investigate security logs, strengthen security measures, and apply patches.

d. Network Connectivity Problems:

Log Entry: Network-related errors, failed connections, or DHCP issues.

Troubleshooting: Examine network logs, use diagnostic tools (e.g., ping, traceroute), and check network configuration.

e. Software Installation Failures:

Log Entry: Errors during software installation or updates.

Troubleshooting: Check application logs, ensure proper permissions, and review installation logs.

f. System Startup Issues:

Log Entry: Boot failures, driver issues, or system hangs during startup.

Troubleshooting: Inspect system logs during boot, update drivers, and perform system repairs.

4.6 Troubleshooting Process:

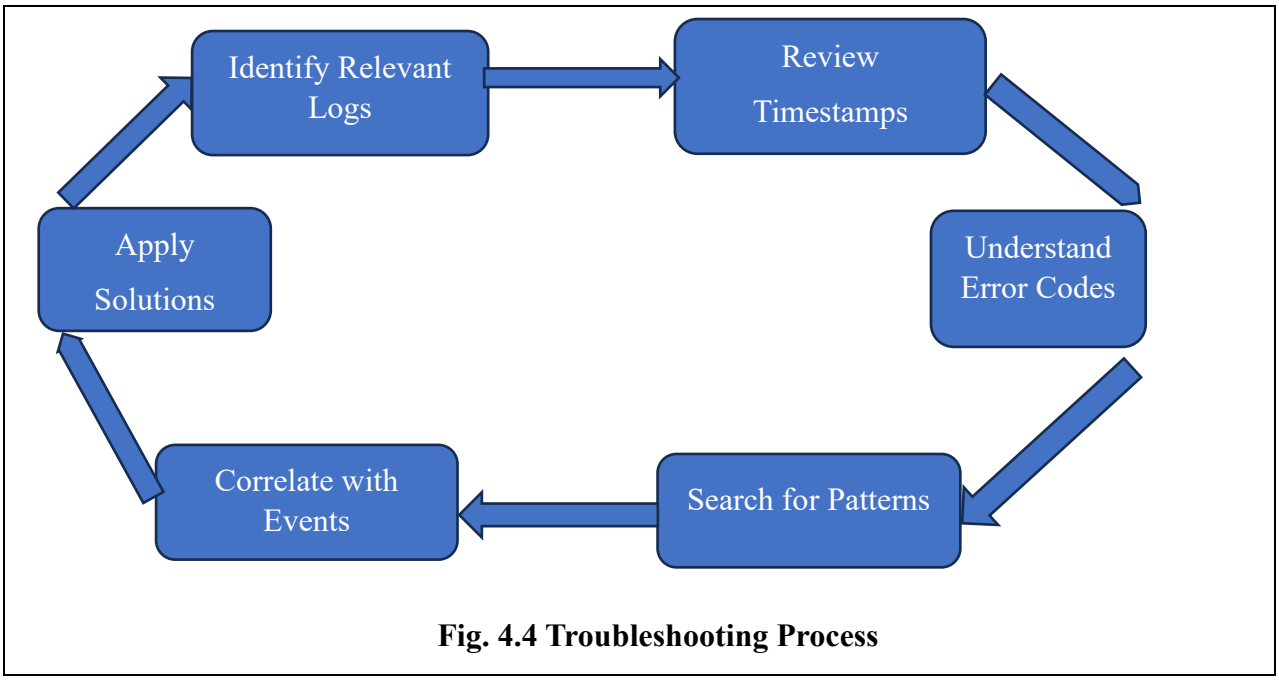


Fig. 4.4 Troubleshooting Process

“**Command Line Mastery”**

a. Identify Relevant Logs:

Look for logs related to the specific issue (e.g., application, system, security).

b. Review Timestamps:

Check when the issue occurred to narrow down relevant log entries.

c. Understand Error Codes:

Decode error codes or messages to understand the nature of the problem.

d. Search for Patterns:

Look for patterns or recurring events that might indicate a recurring issue.

e. Correlate with Events:

Correlate log entries with user actions, system events, or changes.

f. Apply Solutions:

Based on the analysis, apply appropriate solutions (e.g., update drivers, fix configurations, install patches).

Understanding system logs and troubleshooting common issues involve a systematic approach of analyzing logs, identifying patterns, and applying appropriate solutions. Regular log reviews and proactive measures help ensure the smooth functioning and security of a computer system.

5. Networking and Internet Utilities

Networking and internet utilities in the command prompt are essential tools for diagnosing, troubleshooting, and managing network-related tasks. These utilities can be powerful tools for troubleshooting network issues, checking configurations, and gathering information about network connections.

1. ping:

Purpose: Tests network connectivity and measures round-trip time for packets.

Usage: ping [hostname or IP address]

Example: ping www.google.com

2. tracert (traceroute):

Purpose: Traces the route taken by packets from source to destination.

Usage: tracert [hostname or IP address]

Example: tracert www.example.com

3. ipconfig:

Purpose: Displays and manages IP configuration for network interfaces.

Usage: ipconfig

Example: ipconfig /all

4. nslookup:

Purpose: Queries DNS to resolve domain names into IP addresses and vice versa.

Usage: nslookup [hostname or IP address]

Example: nslookup www.example.com

5. netstat:

Purpose: Displays active network connections, listening ports, and routing tables.

Usage: netstat

Example: netstat -an

6. arp:

Purpose: Displays and modifies the ARP cache (Address Resolution Protocol).

Usage: arp -a

Example: arp -a

7. route:

Purpose: Displays or modifies the computer's routing table.

Usage: route print

Example: route add 192.168.1.0 mask 255.255.255.0 192.168.0.1

8. telnet:

Purpose: Establishes a Telnet connection to a specified host.

Usage: telnet [hostname or IP address] [port]

Example: telnet www.example.com 80

9. troubleshooting network connectivity issues:

Purpose: Provides commands to diagnose and troubleshoot network problems.

Usage: troubleshooting network connectivity issues

Example: netsh int ip reset

10. netsh:

Purpose: Configures network settings directly from the command prompt.

Usage: netsh interface show interface

Example: netsh interface ipv4 show addresses

11. nbtstat:

Purpose: Displays NetBIOS over TCP/IP statistics.

Usage: nbtstat -a [hostname]

Example: nbtstat -a ComputerName

12. net use:

Purpose: Connects or disconnects a computer from a shared resource or network drive.

Usage: net use [drive letter] [\\server\share]

Example: net use Z: \\server\share

13. net view:

Purpose: Displays a list of resources shared on a network.

Usage: net view

Example: net view \\server

14. net user:

Purpose: Manages user accounts on a network.

Usage: net user [username] [password] /add

Example: net user JohnDoe P@ssw0rd /add

15. net group:

Purpose: Manages groups of user accounts on a network.

Usage: net group [groupname] /add

Example: net group Administrators /add

These utilities empower users and administrators to troubleshoot network issues, configure network settings, and manage resources efficiently directly from the command prompt. They are valuable for both basic and advanced network tasks.

5.1 Troubleshooting network connectivity issues

Troubleshooting network connectivity issues involves identifying and resolving problems that hinder the proper functioning of a computer or device on a network. Fixing network issues can be challenging, but having a step-by-step plan, using best practices, and relying on tools like Network Performance Monitor can make the process much easier. Brief guide on how to troubleshoot network connectivity issues**:**

Check Physical Connections:

- Ensure all cables (Ethernet, power, etc.) are securely connected.
- For wireless connections, verify that Wi-Fi is enabled, and the device is within range of the network.

Restart Networking Devices:

- Reboot your router, modem, and computer/device to refresh network settings.

Check Network Indicator:

- Look for network indicators on your device (Wi-Fi icon, Ethernet lights) to ensure a connection is established.

Ping Test:

- Use the ping command in the command prompt to test connectivity to a specific IP address or domain.
- Example: ping www.google.com
- A successful ping indicates a working connection; otherwise, investigate the issue.

Check IP Configuration:

- Use the ipconfig command to check your device's IP configuration.
- Example: ipconfig
- Ensure the IP address, subnet mask, and default gateway are correctly configured.

DNS Resolution:

- Use the nslookup command to check DNS resolution.
- Example: nslookup www.example.com
- Verify that the DNS server is resolving domain names correctly.

Firewall and Security Software:

- Temporarily disable firewalls or security software to check if they are blocking network access.

Check Router Configuration:

- Access your router's web interface to ensure proper configuration.
- Verify DHCP settings, wireless settings, and security configurations.

Update Network Drivers:

- Ensure network drivers for your network adapter are up-to-date.
- Visit the device manufacturer's website for driver updates.

Check for Interference:

- For wireless connections, check for interference from other electronic devices.
- Choose a less crowded Wi-Fi channel in your router settings.

Network Troubleshooter:

- Use built-in network troubleshooters provided by your operating system.
- On Windows, run the "Network Troubleshooter" in the Settings menu.

Review Event Logs:

- Check the system logs in the Event Viewer (Windows) for network-related errors or warnings.

Antivirus and Malware Scans:

- Run antivirus and anti-malware scans to ensure your system is not compromised.

Check Service Provider Status:

- Verify with your internet service provider (ISP) if there are any reported outages or issues in your area.

Contact Support:

- If all else fails, contact your network administrator, internet service provider, or device manufacturer for further assistance.

By systematically addressing these steps, you can identify and resolve common network connectivity issues. It's important to follow a structured approach, starting with physical connections and gradually moving to more advanced troubleshooting techniques.

5.2.etwork Troubleshooting Steps:

Check the hardware:

To start troubleshooting, check that all hardware is properly connected, turned on, and functioning. Ensure cords are secure, routers are in the correct positions, and nothing has been

accidentally switched off. Before delving into complex troubleshooting, perform a simple yet effective step: turn off and on the hardware. Power cycling your modem, router, and PC can often resolve basic issues. Just remember to keep each device off for at least 60 seconds before powering it back on. This straightforward approach is a staple in IT problem-solving.

Use ipconfig:

To find your router's IP address, open the command prompt and type "ipconfig." Look for the "Default Gateway" - that's your router's IP. Your computer's IP is next to "IP Address." If it starts with 169, there's an issue. Any other number means your computer has a valid IP from the router.

Use ping and tracert:

If your router is okay and your IP doesn't start with 169, but you still can't connect to the internet, use the ping tool. Try pinging a major server like Google by typing "ping 8.8.8.8" in the command prompt. Add "-t" at the end (ping 8.8.8.8 -t) to keep pinging while you troubleshoot. If pings fail, the command prompt gives info about the problem. Use "tracert 8.8.8.8" to see the path between your router and Google DNS servers. Identify where the error occurs; early errors suggest a local network issue.

Perform a DNS check:

To check if there's an issue with the server you're connecting to, use the "nslookup" command. If you check a website like google.com and get results like "Timed Out," "Server Failure," "Refused," "No Response from Server," or "Network Is Unreachable," it might mean there's a problem with the DNS server for that destination. You can also use "nslookup" to check your own DNS server.

Contact the ISP:

If everything mentioned above doesn't show any issues, reach out to your internet service provider (ISP) to check if they are experiencing problems. You can also check outage maps or use your smartphone to find information to see if others in your area are facing the same problem.

Check on virus and malware protection:

Ensure your antivirus and malware tools are working properly. Check if they have detected any issues that might be impacting a portion of your network, causing it to malfunction.

Review database logs:

Check all your database logs to ensure the databases are working as they should. If your network is fine but the database is full or not working correctly, it might lead to issues that impact your overall network performance.

5.3 Network troubleshooting flowchart:

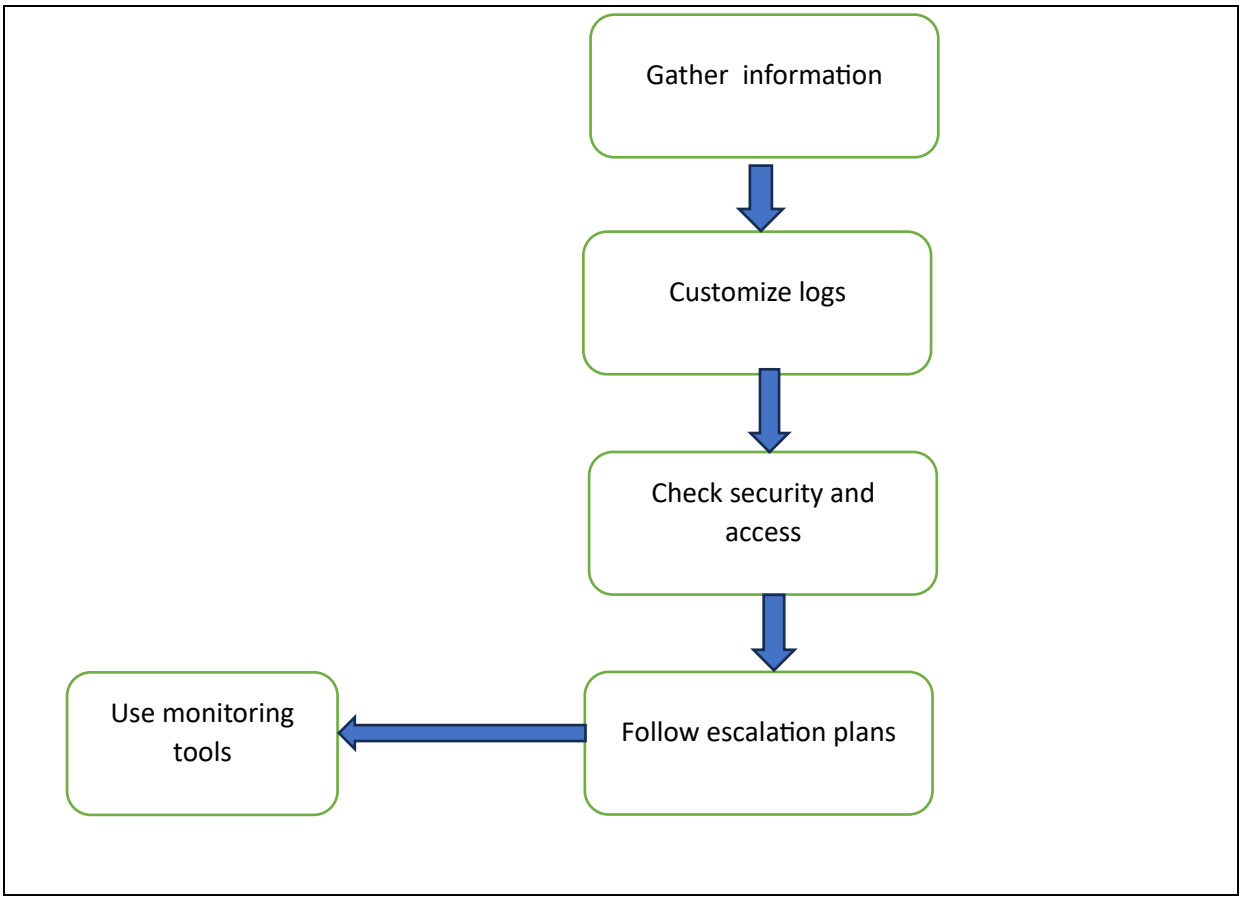


Fig.5.1 Network troubleshooting flowchart

Gather Information:

- Talk to the people experiencing network problems and understand what's going wrong.
- Also, check the network itself to see if you can find and understand the issue.
- Be careful not to confuse the symptoms with the actual cause of the problem.

Customize Logs:

- Set up logs (records of events) in a way that helps you understand and solve problems.
- Logs should clearly show what's happening, when it's happening, and where it's coming from (like a device's address).

Check Access and Security:

- Make sure everyone has the right permissions to access the network.
- Look into firewalls, antivirus, and malware tools to ensure they are working correctly and not causing any issues.

Follow an Escalation Plan:

- Have a clear plan for who handles what issues.
- Make sure everyone knows who to contact for different problems, so you don't waste time going back and forth between people who can't help.

Use Monitoring Tools:

- Instead of manually checking everything, use special tools to keep an eye on the network.

These tools help you gather all the important information quickly and avoid missing anything important. It's like having an extra set of eyes to catch and fix problems, especially in big businesses.

6. Practical Applications and Scenarios

6.1 Real-world scenarios to apply learned commands

Applying learned commands in real-world scenarios within the Command Prompt can help reinforce your skills and enhance your ability to troubleshoot and manage various tasks. Here are some practical scenarios where Command Prompt commands are commonly used:

Network Troubleshooting:

Scenario: Users are reporting intermittent connectivity issues.

Commands:

ipconfig to check IP configuration,

ping to test connectivity to the router and external servers,

tracert to trace the network path,

nslookup to troubleshoot DNS resolution issues.

File and Directory Management:

Scenario: Organizing and managing files and directories.

Commands:

dir to list files and directories.

cd to navigate through directories.

mkdir to create new directories.

copy and xcopy to copy files and directories.

del and rmdir to delete files and directories.

System Health Check:

Scenario: Assessing the overall health of the system.

Commands:

sfc /scannow to scan and repair system files.

chkdsk to check and fix disk errors.

systeminfo to display system information.

tasklist to view running processes.

Network Configuration:

Scenario: Configuring network settings.

Commands:

netsh to configure network interfaces.

ipconfig /release and ipconfig /renew to renew IP address.

net use to connect to network shares.

“**Command Line Mastery”**

Security and Permissions:

Scenario: Checking security settings and permissions.

Commands:

icacls to view and modify file and folder permissions.

netstat to check active network connections.

arp -a to view ARP cache.

Process Management:

Scenario: Managing running processes.

Commands:

tasklist to list running processes.

taskkill to terminate a process.

Registry Editing:

Scenario: Modifying registry settings.

Commands:

regedit to open the Registry Editor.

Scripting and Automation:

Scenario: Automating repetitive tasks.

Commands:

Creating and running batch scripts using echo, if, for, etc.

Remote System Management:

Scenario: Managing a remote system.

Commands:

psexec to execute commands on a remote system.

System Cleanup:

Scenario: Removing temporary files and folders.

Commands:

del to delete files.

rmdir /s /q to delete directories and their contents.

By encountering and addressing these scenarios, you'll gain practical experience and a deeper understanding of how to leverage Command Prompt commands in real-world situations. Remember to approach each scenario systematically, using the appropriate commands for the task at hand.

6.2 Group tasks to solve complex problems using Command Prompt

“**Command Line Mastery”**

To solve complex problems using the Command Prompt, you may need to use a variety of commands and perform specific tasks. Here are common commands and tasks used in troubleshooting complex issues:

- ipconfig:

Task: Check and display IP configuration details.

Usage: ipconfig

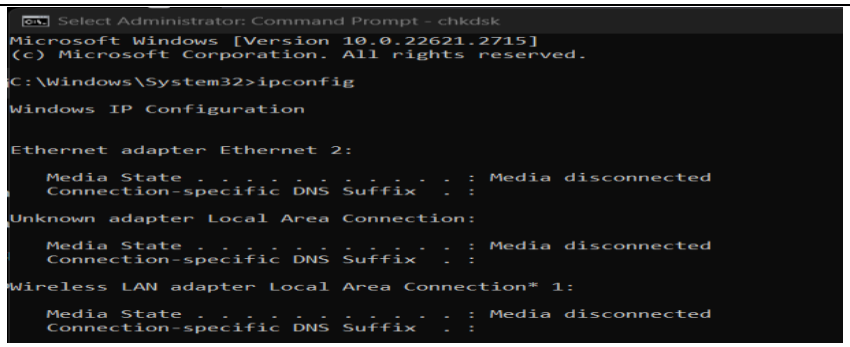


Fig.6.2.1

- ping:

Task: Test network connectivity to a specific IP address or domain.

Usage: ping [IP or domain]

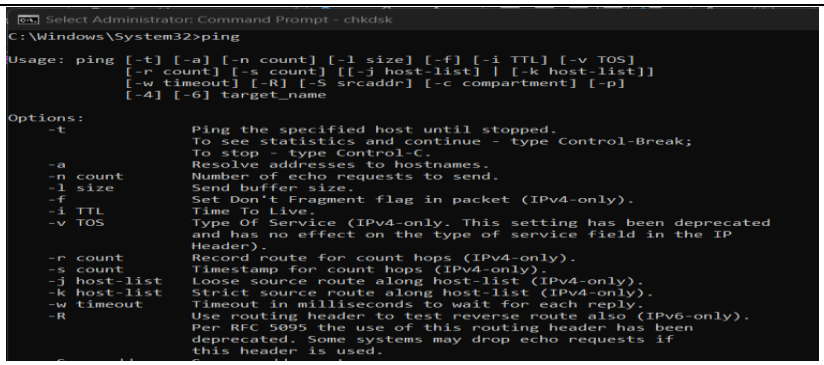


Fig.6.2.2

- tracert:

Task: Trace the route that packets take to reach a destination.

Usage: tracert [IP or domain]

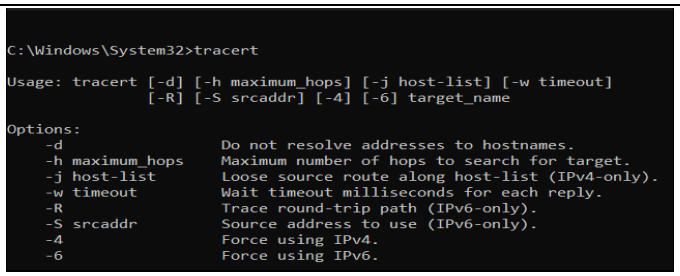


Fig.6.2.3

- nslookup:

Task: Query DNS servers for domain name information.

Usage: nslookup [domain]

- netstat:

Task: Display active network connections, routing tables, and listening ports.

Usage: netstat

- route:

Task: Display or modify the local IP routing table.

Usage: route print

- arp:

Task: Display and modify the ARP (Address Resolution Protocol) cache.

Usage: arp -a

- sfc:

Task: Scan and repair system files.

Usage: sfc /scannow

- chkdsk:

Task: Check a disk for errors and fix them.

Usage: chkdsk [drive letter]: /f

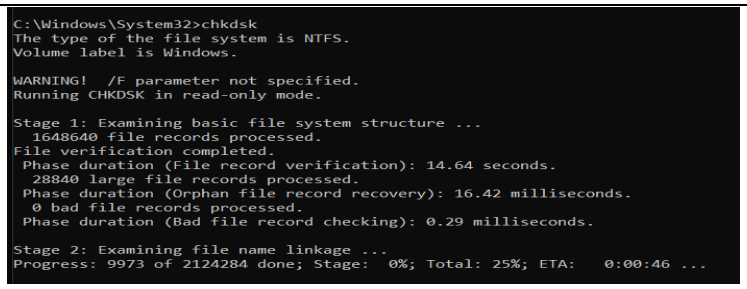


Fig.6.2.4

- systeminfo:

Task: Display detailed configuration information about the computer's hardware and software.

Usage: systeminfo

- tasklist:

Task: Display a list of currently running processes.

Usage: tasklist

- taskkill:

Task: Terminate a running process.

Usage: taskkill /F /IM [process name]

- netsh:

Task: Configure network interfaces and settings.

Usage: Various subcommands like netsh interface ipv4 show interfaces

- regedit:

Task: Edit the Windows Registry.

Usage: regedit

- echo:

Task: Display messages or enable/disable echoing of commands.

Usage: echo [message]

- cls:

Task: Clear the Command Prompt window.

Usage: cls

- mkdir and rmdir:

Task: Create or remove directories.

Usage: mkdir [directory name] and rmdir [directory name]

- copy and xcopy:

Task: Copy files or directories.

Usage: copy [source] [destination] and xcopy [source] [destination] /E

- net use:

Task: Connect to or disconnect from shared resources on a network.

Usage: net use [drive letter]: \\[computer]\[share]

These commands and tasks cover a range of troubleshooting scenarios, allowing you to diagnose and resolve complex issues through the Command Prompt. The specific commands and tasks you use will depend on the nature of the problem you are facing.

7. Project: Automation and Scripting Challenge

7.1 Create a batch script to automate a routine task or solve a problem

Creating a batch script to automate a routine task or solve a problem involves writing a series of commands in a plain text file with a .bat extension. Below is a simple example of a batch script that automates the process of backing up files from one directory to another.

>Text editor>write a code as following>save with filename with .bat extension>Run in cmd.

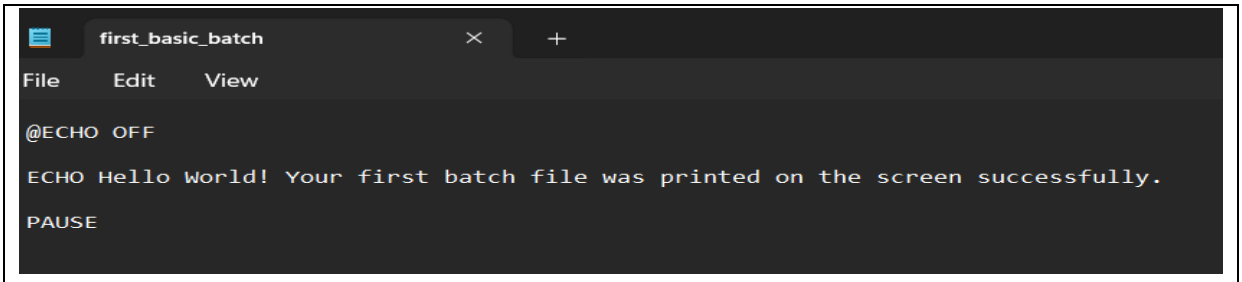


Fig.7.1

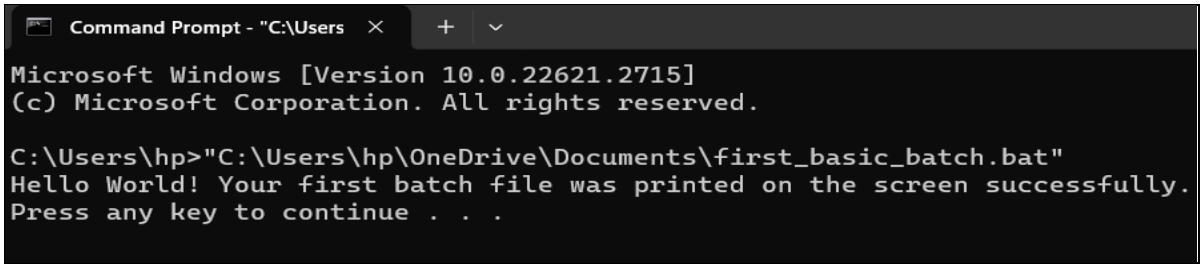


Fig.7.2

This is a basic example, and batch scripts can be as simple or complex as needed for the task at hand. They are powerful for automating repetitive tasks, such as file operations, system configurations, or any sequence of commands you want to execute together.

7.2 Emphasis on creativity, efficiency, and practical utility

Efficiency in Command Prompt Mastery:

- Keyboard Shortcuts and Aliases:

  Creativity: Craft personalized keyboard shortcuts or aliases for frequently used commands.

  Efficiency: Reduce typing effort and execute commands swiftly.

  Practical Utility: Time-saving measures for power users.
- Command Chaining and Piping:

Creativity: Combine multiple commands in a single line for creative workflows.
Efficiency: Execute complex operations with a concise command sequence.
Practical Utility: Perform advanced tasks with minimal effort.

Practical Utility in Everyday Tasks:

- File and Folder Management:

Creativity: Devise naming conventions and organizational structures for files and folders.
Efficiency: Execute file operations quickly using commands like copy and move.
Practical Utility: Maintain a well-organized file system.

- Network Troubleshooting:

Creativity: Develop innovative approaches to diagnose and solve network issues.
Efficiency: Utilize commands like ping and tracert for quick problem identification.
Practical Utility: Ensure seamless connectivity for efficient workflows.

Embracing Creativity in Command Prompt:

- Scripting for Automation:

Creativity: Develop scripts (batch files) to automate repetitive tasks.
Efficiency: Create imaginative solutions to streamline workflows.
Practical Utility: Automation allows for hands-free execution of complex sequences.

- Customizing the Command Prompt:

Creativity: Personalize the Command Prompt appearance with color schemes and fonts.
Efficiency: Easily distinguish directories and files for quicker navigation.
Practical Utility: A visually appealing Command Prompt enhances user experience.

"**Command Line Mastery"**

Learning Outcomes:

- Develop adaptability and learning agility by exploring new command prompt functionalities and adapting to evolving project requirements.
- Learn advanced command prompt commands for tasks like system diagnostics, network troubleshooting, and file backups.
- Develop an understanding of file systems and how to manage them via the command line.
- Cultivate critical thinking skills by analyzing problems, executing commands, and evaluating outcomes to achieve desired results.
- Acquire skills in system diagnostics, network troubleshooting, and basic scripting.
- Enhance problem-solving abilities in a command-line environment.