S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT & RESEARCH, NAGPUR

Practical 02

Aim: To understand and demonstrate the use of basic commands in different operating systems (Windows, Linux, and UNIX) for managing files, directories, permissions, and user interactions through a terminal or command-line interface.

Name: Aaditya S Gudhniye

USN:CM24062

Semester / Year:

Academic Session:

Date of Performance:

Date of Submission:

❖ **Aim:** To understand and demonstrate the use of basic commands in different operating systems (Windows, Linux, and UNIX) for managing files, directories, permissions, and user interactions through a terminal or command-line interface.

❖ Objectives:

1. To learn and practice fundamental command-line operations for file and directory management.
2. To explore and utilize user and permission management commands effectively.
3. To enhance system administration skills by working with commands across different operating systems.

❖ Requirements:

Hardware Requirements:

- **Processor**: Multi-core CPU, Intel Core i3 (3.0 GHz) or higher
- **RAM**: Minimum 4 GB (8 GB recommended for optimal performance)
- **Storage**: 100 GB HDD or SSD (Solid State Drive) for faster access
- **Network Interface**: Ethernet or Wi-Fi adapter for connectivity

Software Requirements:

- **Operating System**: Windows 10/11, Linux (Ubuntu 20.04/CentOS 8), UNIX-based OS
- **Command-line Interface**: PowerShell or Command Prompt (Windows), Terminal (Linux/UNIX)
- **Text Editor**: Nano, Vim, or Visual Studio Code for file editing
- **Administrative Privileges**: Superuser (Linux/UNIX) or Administrator (Windows) access

❖ Theory:

In system administration, command-line interfaces (CLI) are essential tools for managing and interacting with operating systems like Windows, Linux, and UNIX. Commands allow users to perform various tasks such as navigating directories, managing files, controlling permissions, and monitoring system performance. Each operating system provides a set of built-in commands, such as ‘man’, ‘ls’, ‘cd’, ‘mkdir’, and ‘chmod’, to facilitate efficient system management. Understanding these commands and their syntax is crucial for automating tasks, enhancing security, and ensuring optimal system functionality. This practical aims to develop foundational skills in executing and applying basic commands across different platforms.

❖ Commands:

1. **Display User Manual of a Command**
   - Functionality: Shows the manual page with details about a command's usage, options, and arguments.
   - Syntax: man <command>
   - Example: man ls

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> man ls
NAME
    Get-ChildItem

SYNTAX
    Get-ChildItem [[-Path] <string[]>] [[-Filter] <string>] [-Include <string[]>] [-Exclude <string[]>] [-Recurse]
    [-Depth <uint32>] [-Force] [-Name] [-UseTransaction] [-Attributes {ReadOnly | Hidden | System | Directory |
    Archive | Device | Normal | Temporary | SparseFile | ReparsePoint | Compressed | Offline | NotContentIndexed |
    Encrypted | IntegrityStream | NoScrubData}] [-FollowSymlink] [-Directory] [-File] [-Hidden] [-ReadOnly] [-System]
    [<CommonParameters>]

    Get-ChildItem [[-Filter] <string>] -LiteralPath <string[]> [-Include <string[]>] [-Exclude <string[]>] [-Recurse]
    [-Depth <uint32>] [-Force] [-Name] [-UseTransaction] [-Attributes {ReadOnly | Hidden | System | Directory |
    Archive | Device | Normal | Temporary | SparseFile | ReparsePoint | Compressed | Offline | NotContentIndexed |
    Encrypted | IntegrityStream | NoScrubData}] [-FollowSymlink] [-Directory] [-File] [-Hidden] [-ReadOnly] [-System]
    [<CommonParameters>]


ALIASES
    gci
    ls
    dir


REMARKS
    Get-Help cannot find the Help files for this cmdlet on this computer. It is displaying only partial help.
        -- To download and install Help files for the module that includes this cmdlet, use Update-Help.
        -- To view the Help topic for this cmdlet online, type: "Get-Help Get-ChildItem -Online" or
           go to https://go.microsoft.com/fwlink/?LinkID=113308.

-- More  --
```

2. **Change Current Working Directory**.
   - Functionality: Changes the terminal's current working directory.
   - Syntax: cd <directory-path>
   - Example: cd /home/user/Documents.

```
PS C:\Users\ASUS> cd newdir
PS C:\Users\ASUS\newdir>
```

3. **List Contents of the Current Directory.**
   - Functionality: Lists all files and directories in the current location.
   - Syntax: ls
   - Example: ls

```
PS C:\Users\ASUS\newdir> ls


    Directory: C:\Users\ASUS\newdir


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         03-02-2026     01:35             36 combine.txt
-a----         03-02-2026     01:29             28 file.txt
-a----         03-02-2026     01:31             10 file1.txt


PS C:\Users\ASUS\newdir>
```

4. **Read/Modify/Concatenate Text Files.**
   - Functionality: Displays or manipulates file content.
   - Syntax:
     - Read: cat <filename>

```
PS C:\Users\ASUS\newdir> "hello world" | Out-File -FilePath "file.txt"
PS C:\Users\ASUS\newdir> cat file.txt
hello world
PS C:\Users\ASUS\newdir>
```

     - Modify: ‘nano <filename>
     - Concatenate: cat <file1> <file2> > <outputfile>

```
PS C:\Users\ASUS\newdir> cat file.txt ,file1.txt >> combine.txt
PS C:\Users\ASUS\newdir> cat combine.txt
hello world
hi
PS C:\Users\ASUS\newdir>
```

5. **Create a New Directory**.
   - Functionality: Creates a new directory at the specified path.
   - Syntax: mkdir <directory-name
   - Example: mkdir newdir

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ASUS> mkdir newdir


    Directory: C:\Users\ASUS


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        03-02-2026     01:23                newdir


PS C:\Users\ASUS>
```

6. **Display Current Working Directory.**
   - Functionality: Prints the current directory path.
   - Syntax: pwd
   - Example: pwd

```
PS C:\Users\ASUS\newdir> pwd

Path
----
C:\Users\ASUS\newdir


PS C:\Users\ASUS\newdir>
```

7. **Write Arguments to Standard Output.**
   - Functionality: Prints the provided string or variables.
   - Syntax: echo <arguments>
   - Example: echo Hello World

```
PS C:\Users\ASUS\newdir> echo Hello World
Hello
World
PS C:\Users\ASUS\newdir>
```

8. **Remove a File**.
   - Functionality: Deletes a specified file.
   - Syntax: rm <filename>
   - Example: rm file.txt

```
PS C:\Users\ASUS\newdir> rm file.txt
PS C:\Users\ASUS\newdir> ls


    Directory: C:\Users\ASUS\newdir


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        03-02-2026     01:35             36 combine.txt
-a----        03-02-2026     01:31             10 file1.txt


PS C:\Users\ASUS\newdir>
```

9. **Delete a Directory.**
   - Functionality: Removes an empty directory.
   - Syntax: rmdir <directory-name>
   - Example: rmdir olddir

```
PS C:\Users\ASUS\newdir> cd..
PS C:\Users\ASUS> rmdir newdir

Confirm
The item at C:\Users\ASUS\newdir has children and the Recurse parameter was not specified. If you continue, all
children will be removed with the item. Are you sure you want to continue?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
PS C:\Users\ASUS> cd newdir
cd : Cannot find path 'C:\Users\ASUS\newdir' because it does not exist.
At line:1 char:1
+ cd newdir
+ ~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Users\ASUS\newdir:String) [Set-Location], ItemNotFoundException
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLocationCommand

PS C:\Users\ASUS>
```

10. **Copy a File or Directory.**
    - Functionality: Copies a file or directory to a destination.
    - Syntax: cp <source> <destination>
    - Example: cp file.txt backup/

```
PS C:\Users\ASUS\newdir> cp file.txt backup.txt
PS C:\Users\ASUS\newdir> ls


    Directory: C:\Users\ASUS\newdir


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        03-02-2026     01:39             28 backup.txt
-a----        03-02-2026     01:35             36 combine.txt
-a----        03-02-2026     01:39             28 file.txt
-a----        03-02-2026     01:31             10 file1.txt


PS C:\Users\ASUS\newdir> cat backup.txt
hello world
PS C:\Users\ASUS\newdir>
```

11. **Switch to Root User.**
    - Functionality: Gains root privileges temporarily.
    - Syntax: sudo su
    - Example: sudo s

```
PS C:\Users\ASUS\newdir> sudo s
Sudo is disabled on this machine. To enable it, go to the Developer Settings page in the Settings app
PS C:\Users\ASUS\newdir> sudo s
Sudo is disabled on this machine. To enable it, go to the Developer Settings page in the Settings app
PS C:\Users\ASUS\newdir> sudo su
Sudo is disabled on this machine. To enable it, go to the Developer Settings page in the Settings app
PS C:\Users\ASUS\newdir>
```

12. **Move Files or Directories.**
    - Functionality: Moves or renames files and directories.
    -Syntax: mv <source> <destination>
    - Example: mv file.txt newdir/

```
PS C:\Users\ASUS\newdir> mv file.txt new1/
```

```
PS C:\Users\ASUS> cd new1/
PS C:\Users\ASUS\new1> ls
PS C:\Users\ASUS\new1> "hello world" | Out-File -FilePath file.txt
PS C:\Users\ASUS\new1> ls


    Directory: C:\Users\ASUS\new1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        03-02-2026     01:48             28 file.txt


PS C:\Users\ASUS\new1>
```

```
PS C:\Users\ASUS\newdir> cd..
PS C:\Users\ASUS> mkdir new1


    Directory: C:\Users\ASUS


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        03-02-2026     01:41                new1


PS C:\Users\ASUS> cd newdir
PS C:\Users\ASUS\newdir> mv file.txt new1/
PS C:\Users\ASUS\newdir> ls


    Directory: C:\Users\ASUS\newdir


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        03-02-2026     01:39             28 backup.txt
-a----        03-02-2026     01:35             36 combine.txt
-a----        03-02-2026     01:31             10 file1.txt
-a----        03-02-2026     01:39             28 new1


PS C:\Users\ASUS\newdir>
```

13. **Search for a String in a File.**
    - Functionality: Searches for a specific word or pattern in a file.
    - Syntax: grep "<string>" <file>
    - Example: grep "error" log.txt

```
PS C:\Users\ASUS\newdir> sls "hello world" file.txt

file.txt:1:hello world


PS C:\Users\ASUS\newdir>
```

14. **Print Top N Lines of a File.**
    - Functionality: Displays the first N lines of a file.
    - Syntax: head -n <N> <file>
    - Example: ‘head -n 10 file.txt’

```
PS C:\Users\ASUS\newdir> $lines = Get-Content file.txt
PS C:\Users\ASUS\newdir> $lines = $lines[1], "ola world"
PS C:\Users\ASUS\newdir> $lines | Set-Content file.txt
PS C:\Users\ASUS\newdir> cat file.txt
hello world
ola world
PS C:\Users\ASUS\newdir>
```

```
PS C:\Users\ASUS\newdir> Get-Content file.txt -TotalCount 2
hello world
ola world
PS C:\Users\ASUS\newdir>
```

15. **Print Last N Lines of a File.**
    - Functionality: Displays the last N lines of a file.
    - Syntax: tail -n <N> <file>
    - Example: ‘tail -n 10 file.txt

```
PS C:\Users\ASUS\newdir> Get-Content file.txt -Tail 1
ola world
PS C:\Users\ASUS\newdir>
```

16. **Remove Read Permission from Owner.**
    - Functionality: Revokes the owner’s read permission for a file.
    - Syntax: chmod u-r <filename>
    - Example: chmod u-r file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell equivalent - Remove read permission for current user
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $rule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "Read", "Allow"
>> )
PS C:\Users\ASUS\newdir> $acl.RemoveAccessRule($rule)
True
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

17. **Change Specific Permissions.**
    - Functionality: Sets or removes specific file permissions.
    -Syntax: chmod u+r,w-x,g+w <filename>
    - Example: chmod u+r,w-x,g+w file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Set specific permissions for user, group, others
PS C:\Users\ASUS\newdir> # Example: u+r,w-x,g+w becomes complex in Windows
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $userRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "Read, Write", "Allow"
>> )
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($userRule)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

18. **Add Write Permission to Owner, None to Others**.
    -Functionality: Allows write access for the owner only.
    - Syntax: chmod u+w,o-rwx <filename>
    - Example: chmod u+w,o-rwx file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Grant write to owner, remove all from others
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $ownerRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "Write", "Allow"
>> )
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($ownerRule)
PS C:\Users\ASUS\newdir> # Remove inheritance and clear all other rules
PS C:\Users\ASUS\newdir> $acl.SetAccessRuleProtection($true, $false)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

19. **Assign Permissions to Users.**
    - Functionality: Modifies file access for users, groups, and others.
    - Syntax: chmod u+rwx,g+rx,o+r <filename>
    - Example: ‘chmod u+rwx,g+rx,o+r file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Set different permissions for different users/groups
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> # User: FullControl
PS C:\Users\ASUS\newdir> $userRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "FullControl", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> # Group: ReadAndExecute
PS C:\Users\ASUS\newdir> $groupRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Users", "ReadAndExecute", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> # Others: Read
PS C:\Users\ASUS\newdir> $everyoneRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Everyone", "Read", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($userRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($groupRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($everyoneRule)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

20. **Assign R/W/X to Others.**
    - Functionality: Gives read, write, and execute permissions to others.
    - Syntax: chmod o+rwx <filename>
    - Example: chmod o+rwx file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Grant FullControl to Everyone
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $rule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Everyone", "FullControl", "Allow"
>> )
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($rule)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

21. **Remove All Permissions from All Users.**
    - Functionality: Clears all permissions on a file.
    - Syntax: ‘chmod a-rwx <filename>
    - Example: ‘chmod a-rwx file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Remove all permissions (take ownership first may be needed)
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $acl.SetAccessRuleProtection($true, $false)  # Break inheritance, remove all inherited rules
PS C:\Users\ASUS\newdir> $acl.Access | ForEach-Object { $acl.RemoveAccessRule($_) }  # Remove explicit rules
True
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

22. **Remove Read Permission Using Absolute Mode.**
    - Functionality: Uses numeric mode to restrict read access.
    - Syntax: chmod 700 <filename>
    - Example: chmod 700 file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Set permissions to 700 equivalent (Owner: FullControl, Others: None)
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $ownerRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "FullControl", "Allow"
>> )
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($ownerRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRuleProtection($true, $false)  # Remove inheritance
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

23. **Set R/W for Owner, None for Group/Other.**
    - Functionality: Assigns permissions in numeric mode.
    - Syntax: chmod 600 <filename>
    - Example: chmod 600 file.txt'

```
PS C:\Users\ASUS\newdir> # PowerShell - Owner: Execute, Group/Others: Read
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $ownerRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "ExecuteFile", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $groupRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Users", "Read", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $everyoneRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Everyone", "Read", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($ownerRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($groupRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($everyoneRule)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

24. **Add Execute for Owner, Read for Group/Others.**
    - Functionality: Adds execution and read access.
    - Syntax: chmod u+x,g+r,o+r <filename>
    - Example: chmod u+x,g+r,o+r file.txt

```
PS C:\Users\ASUS\newdir> # PowerShell - Owner: Execute, Group/Others: Read
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $ownerRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "$env:USERNAME", "ExecuteFile", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $groupRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Users", "Read", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $everyoneRule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Everyone", "Read", "Allow"
>> )
PS C:\Users\ASUS\newdir>
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($ownerRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($groupRule)
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($everyoneRule)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

25. **Add Execute Permission to All Users.**
    - Functionality: Enables execution by everyone.
    - Syntax: chmod a+x <filename>
    - Example: chmod a+x script.sh

```
PS C:\Users\ASUS\newdir> # PowerShell - Add execute permission for all users
PS C:\Users\ASUS\newdir> $acl = Get-Acl file.txt
PS C:\Users\ASUS\newdir> $rule = New-Object System.Security.AccessControl.FileSystemAccessRule(
>>     "Everyone", "ExecuteFile", "Allow"
>> )
PS C:\Users\ASUS\newdir> $acl.SetAccessRule($rule)
PS C:\Users\ASUS\newdir> Set-Acl file.txt $acl
```

- **Conclusion:** In conclusion, understanding and using essential operating system commands like **'ls', 'cd', 'cp', 'mv', and 'chmod'** enables efficient file management, navigation, and permission control. Tools like **'grep', 'head', and 'tail'** enhance data processing. Mastery of these commands improves system administration, task automation, and overall system security and performance.

- **Discussion Questions:**
  1. **What is the significance of the pwd command in a Linux environment?**
  2. **Explain the function of the cp command and its common options.**
  3. **How does chmod 700 affect file permissions, and what does each digit represent?**
  4. **Describe the difference between head and tail commands in Linux.**
  5. **What is the purpose of the grep command, and how is it used with regular expressions?**

- **References**:

https://ubuntu.com/tutorials/command-line-for-beginners#1-overview
https://www.geeksforgeeks.org/25-basic-ubuntu-commands/

Date: / /2026

Signature
Course Coordinator
B.Tech CSE(AIML)
Sem: 4 / 2025-26

-