TASK- 1 **Shell Scripting**

* Shell scripting is a method of automating tasks by writing a series of commands in a file, which can then be executed as a program.
* PowerShell is a powerful command-line shell and scripting language developed by Microsoft, designed for system administration, automation, and configuration management.

**Commands:**

* **Get-command:** Give the commands present in the power shell.
* PS C:\Users\shobi> get-command
* CommandType Name Version Source
* ----------- ---- ------- ------
* Alias Add-AppPackage 2.0.1.0 Appx
* Alias Add-AppPackageVolume 2.0.1.0 Appx
* Alias Add-AppProvisionedPackage 3.0 Dism
* Alias Add-MsixPackage 2.0.1.0 Appx
* Alias Add-MsixPackageVolume 2.0.1.0 Appx
* Alias Add-MsixVolume 2.0.1.0 Appx
* Alias Add-ProvisionedAppPackage 3.0 Dism
* Alias Add-ProvisionedAppSharedPackageContainer 3.0 Dism
* Alias Add-ProvisionedAppxPackage 3.0 Dism
* Alias Add-ProvisioningPackage 3.0 Provisioning
* Alias Add-TrustedProvisioningCertificate 3.0 Provisioning
* Alias Apply-WindowsUnattend 3.0 Dism
* Alias Disable-PhysicalDiskIndication 2.0.0.0 Storage
* Alias Disable-PhysicalDiskIndication 1.0.0.0 VMDirectStorage
* Alias Disable-StorageDiagnosticLog 2.0.0.0 Storage
* Alias Disable-StorageDiagnosticLog 1.0.0.0 VMDirectStorage
* Alias Dismount-AppPackageVolume 2.0.1.0 Appx
* Alias Dismount-MsixPackageVolume 2.0.1.0 Appx
* Alias Dismount-MsixVolume 2.0.1.0 Appx
* Alias Enable-PhysicalDiskIndication 2.0.0.0 Storage
* Alias Enable-PhysicalDiskIndication 1.0.0.0 VMDirectStorage
* Alias Enable-StorageDiagnosticLog 2.0.0.0 Storage
* Alias Enable-StorageDiagnosticLog 1.0.0.0 VMDirectStorage
* Alias Flush-Volume 2.0.0.0 Storage
* Alias Flush-Volume 1.0.0.0 VMDirectStorage
* Alias Get-AppPackage 2.0.1.0 Appx
* Alias Get-AppPackageAutoUpdateSettings 2.0.1.0 Appx

**Get-help:** Provides more in-depth information about the cmdlet, including parameter descriptions.

* Cmdlets are the native commands used within PowerShell. They are fundamental to how PowerShell operates.
* PS C:\Users\shobi> get-help
* TOPIC
* Windows PowerShell Help System
* SHORT DESCRIPTION
* Displays help about Windows PowerShell cmdlets and concepts.
* LONG DESCRIPTION
* Windows PowerShell Help describes Windows PowerShell cmdlets,
* functions, scripts, and modules, and explains concepts, including
* the elements of the Windows PowerShell language.
* Windows PowerShell does not include help files, but you can read the
* help topics online, or use the Update-Help cmdlet to download help files
* to your computer and then use the Get-Help cmdlet to display the help
* topics at the command line.
* You can also use the Update-Help cmdlet to download updated help files
* as they are released so that your local help content is never obsolete.
* Without help files, Get-Help displays auto-generated help for cmdlets,
* functions, and scripts.
* ONLINE HELP
* You can find help for Windows PowerShell online in the TechNet Library
* beginning at http://go.microsoft.com/fwlink/?LinkID=108518.
* To open online help for any cmdlet or function, type:
* Get-Help <cmdlet-name> -Online
* UPDATE-HELP
* To download and install help files on your computer:
* 1. Start Windows PowerShell with the "Run as administrator" option.
* 2. Type:
* **Get-process**: list running process.
* PS C:\Users\shobi> Get-Process
* Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
* ------- ------ ----- ----- ------ -- -- -----------
* 365 27 53772 99648 11.95 9176 16 AdskAccessCore
* 233 17 16376 31696 0.67 22956 16 AdskAccessService
* 330 27 10672 19216 27388 0 AdskAccessServiceHost
* 390 26 45844 67284 0.88 7368 16 AdskAccessUIHost
* 907 42 71616 80504 15.72 13940 16 AdskAccessUIHost
* 575 23 29560 49760 0.44 18256 16 AdskAccessUIHost
* 355 18 14424 49376 0.70 27404 16 AdskAccessUIHost
* 380 25 6628 22512 108.33 26272 16 AdskIdentityManager
* 162 13 21632 10228 5672 0 AdskLicensingService
* 147 10 3796 5676 9956 0 AggregatorHost
* 232 18 23084 21652 0.59 16068 16 ai
* 159 11 2820 2044 3228 0 amdfendrsr
* 167 11 2232 2200 0.13 7044 16 amdow
* 551 22 6868 160120 7.00 18532 16 AMDRSServ
* 453 42 111748 59048 1.05 11080 16 AMDRSSrcExt
* 611 22 12604 16736 11092 0 AppHelperCap
* 371 17 4628 18836 20596 16 atieclxx
* 242 15 2956 2584 3236 0 atiesrxx
* 178 14 3284 10768 0.06 18756 16 browser\_assistant
* **Stop-process:** stop the running process.
* **Cls**: To clear the commands in power shell.
* **Get-service**: Lists services
* PS C:\Users\shobi> Get-Service
* Status Name DisplayName
* ------ ---- -----------
* Stopped AarSvc\_2d1ce004 Agent Activation Runtime\_2d1ce004
* Running AdskLicensingSe... Autodesk Desktop Licensing Service
* Stopped ALG Application Layer Gateway Service
* Running AMD Crash Defen... AMD Crash Defender Service
* Running AMD External Ev... AMD External Events Utility
* Stopped AppIDSvc Application Identity
* Running Appinfo Application Information
* Stopped AppReadiness App Readiness
* Running AppXSvc AppX Deployment Service (AppXSVC)
* Stopped ApxSvc Windows Virtual Audio Device Proxy ...
* Stopped aspnet\_state ASP.NET State Service
* Running AudioEndpointBu... Windows Audio Endpoint Builder
* Running Audiosrv Windows Audio
* **Start-Service** <service name> -- Starts a service.
* **Stop-Service** <service name> -- Stops a service.
* **Get-ChildItem**  -- Lists files and directories.

**To create the folder:**

* PS C:\Users\shobi> New-Item -Itemtype Directory -Path "D:\MyFolder3"
* Directory: D:\
* Mode LastWriteTime Length Name
* ---- ------------- ------ ----
* d----- 11-03-2025 04:56 PM MyFolder3

**To create the file:**

* PS C:\Users\shobi> New-Item -ItemType file -Path "D:\MyFolder1\pstest.txt"
* Directory: D:\MyFolder1
* Mode LastWriteTime Length Name
* ---- ------------- ------ ----
* -a---- 11-03-2025 05:04 PM 0 pstest.txt

**To set the content in file:**

* PS C:\Users\shobi> set-content -Path "D:\MyFolder1\pstest.txt" -value "shobithapentakota mca"

**To display the content in file:**

**PS C:\Users\shobi> get-content -Path "D:\MyFolder1\pstest.txt"**

**shobithapentakota mca**

**To know the how many files in the folder:**

* PS C:\Users\shobi> Get-ChildItem -Path "D:\MyFolder1"
* Directory: D:\MyFolder1
* Mode LastWriteTime Length Name
* ---- ------------- ------ ----
* -a---- 11-03-2025 05:12 PM 23 pstest.txt

**To remove the files from the folder:**

* PS C:\Users\shobi> Remove-Item -Path "D:\MyFolder1"

**To remove the folder :**

* PS C:\Users\shobi> Remove-item -path "D:\MyFolder3"
* $variable = "value" -- Assigns a value to a variable.
* Write-Host $variable -- Displays the value of a variable.
* PS C:\Users\shobi> $number=10
* Write-Host $number
* 10

if statement:

* $number = 10  
   if ($number -gt 5)  
   {   
  Write-Host "Number is greater than 5"  
   }  
   elseif ($number -eq 5)   
  {   
  Write-Host "Number is equal to 5"   
  }  
   else {   
  Write-Host "Number is less than 5"   
  }
* **for loop:**

Write-Host "Eg 1: Printing 1 to 10"

for ($i = 1; $i -le 10; $i++) {

Write-Host $i

}

* Eg 1: Printing 1 to 10
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

**Program:**

* **# Define the folder name and path**

$folderName = "MyTestFolder"

$folderPath = Join-Path -Path "D:\AIWorkLoads" -ChildPath $folderName #Adjust C:\temp as needed

**# Create the folder if it doesn't exist**

if (-not (Test-Path -Path $folderPath)) {

New-Item -ItemType Directory -Path $folderPath | Out-Null

Write-Host "Created folder: $folderPath"

} else {

Write-Host "Folder already exists: $folderPath"

}

**# Create 5 files within the folder with date and time in the filename and add content**

for ($i = 1; $i -le 5; $i++) {

$dateTime = Get-Date -Format "yyyyMMdd"

$fileName = "file\_$($dateTime)\_$($i).txt"

$filePath = Join-Path -Path $folderPath -ChildPath $fileName

**# Create the file**

New-Item -ItemType File -Path $filePath -Force | Out-Null

**# Set content for the file**

$fileContent = "Softwaregurukulam\_$fileName"

Set-Content -Path $filePath -Value $fileContent

Write-Host "Created file: $filePath with content: $fileContent"}

**Task -2:**

**Linux Commands**

Linux is a powerful, open-source operating system that serves as a foundation for a wide range of devices, from servers to personal computers and mobile devices. Developed initially by Linus Torvalds in the early 1990s, Linux is based on the Unix operating system and has become one of the most widely used operating systems globally, particularly in server environments and supercomputing.

**Overview of Linux :**

**Kernel and Distributions**  
At the core of Linux is the Linux kernel, which manages hardware resources and software applications. The modular design of Linux allows for various distributions (distros), each tailored to specific needs or use cases. Popular distributions include Ubuntu, Red Hat, Fedora, CentOS, and Debian. These distros can be customized extensively due to their open-source nature.

**Applications :**  
Linux is utilized in numerous applications, including:

* **Server Management**: It is the predominant OS for servers due to its stability and security.
* **Cloud Computing**: Many cloud service providers favor Linux for its flexibility and cost-effectiveness.
* **Embedded Systems**: Linux is widely used in embedded devices, including automotive systems through projects like Automotive Grade Linux (AGL)[3](https://www.techtarget.com/searchdatacenter/definition/Linux-operating-system)[4](https://www.linux.com/what-is-linux/).

**Benefits of Using Linux:**

The advantages of Linux stem from its open-source model:

* **Customization**: Users can modify the source code to suit their specific needs.
* **Security**: Linux is known for its robust security features, often requiring fewer antivirus solutions compared to other operating systems.
* **Stability and Performance**: It offers high performance with minimal resource consumption, making it suitable for both high-end servers and low-spec devices.

**Learning and Using Linux :**

For those interested in learning Linux, it can be installed on most computers using virtual machines like Oracle VirtualBox. This allows users to experiment with different distributions without affecting their primary operating system. New users often start with Ubuntu due to its user-friendly interface.

In summary, Linux stands out as a versatile operating system that caters to a diverse range of applications while providing significant benefits in terms of customization, security, and performance. Its community-driven development continues to foster innovation across various sectors.

**Linux commands:**

**Pwd:** present working directory.

$ pwd

/c/Users/shobi

**Ls**: listing the files and folders.

**Mkdir**: making a directory.

**Cd**: To change the directory.

**Clear:** clear the screen.

**Touch**: To create a file.

**To modify the file**: vi filename>>>>>click I>>>>>enter some data>>>>press ESC button>>>:qw!

**TO see the Data in file**: cat filename

**Cd ..** : to get one step back from the pwd.

**Uname**: it shows the kernal and its version.

**Whoami**: It gives the user name.

**Date**: It gives the current date.

**Touch file{1..10}**>>>>>>create the files

**Mkdir dir{1..10}**>>>>>create the dir.

**Mv :** move and rename the files.

**Rm-rf** : removing,forcibily& recursively.

**History:** To see the history of commands.

**Cal**: To see calender.

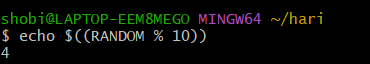
**To print the random numbers:**

Echo $RANDOM

Echo $((RANDOM % 10))

Echo $(( 10+ RANDOM % 10))



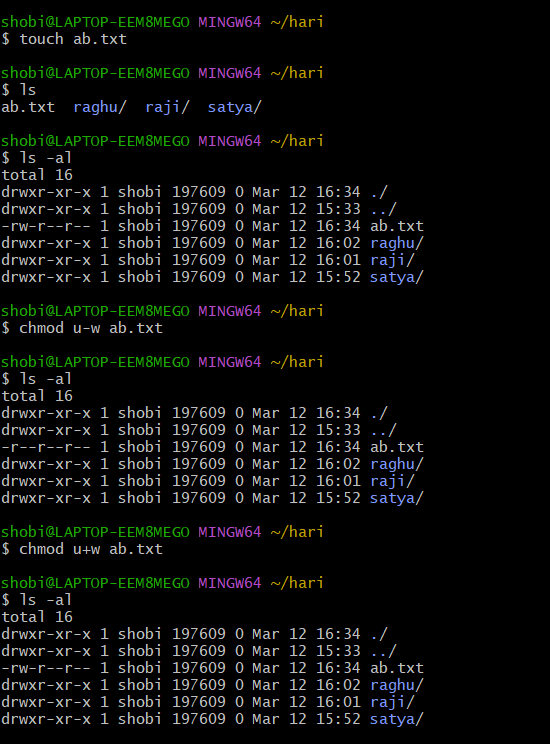


**To change the permissions of a particular file:**

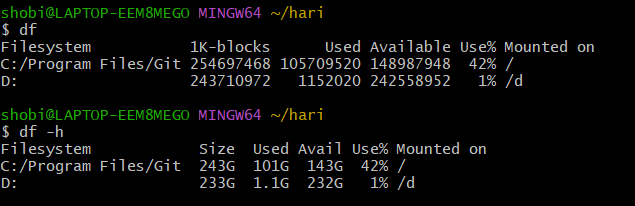
Chmod u-w filename

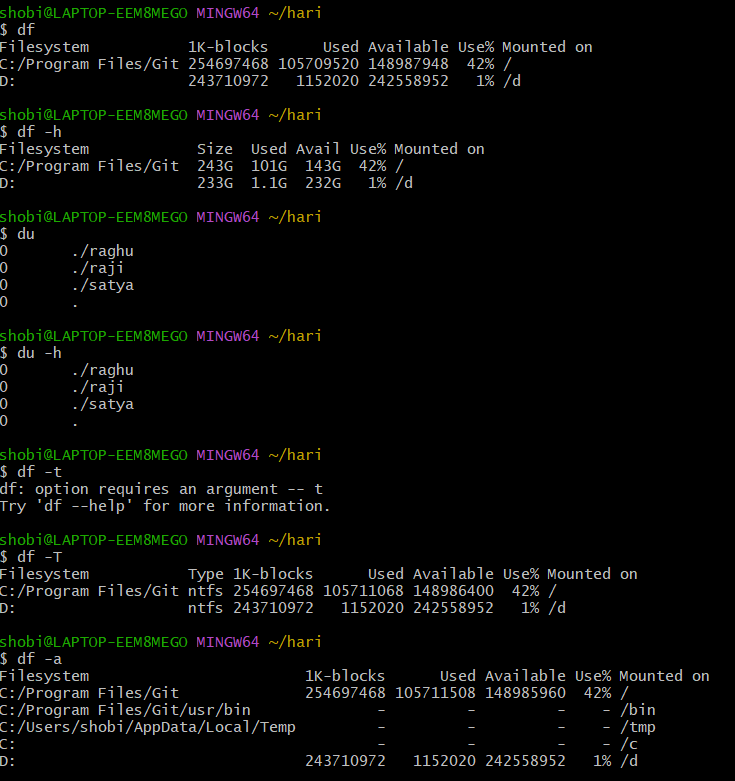
Chmod u+r filename

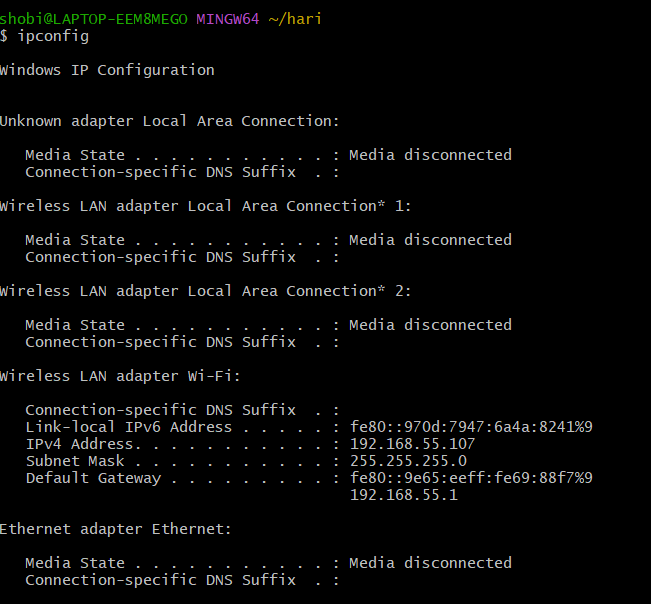
Chmod g\_r filename



* df >>>space in our files
* df -h>>>>>it gives in human readable format



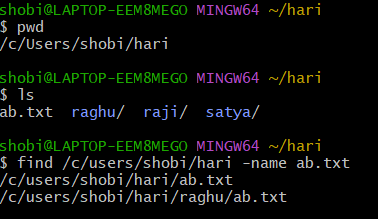




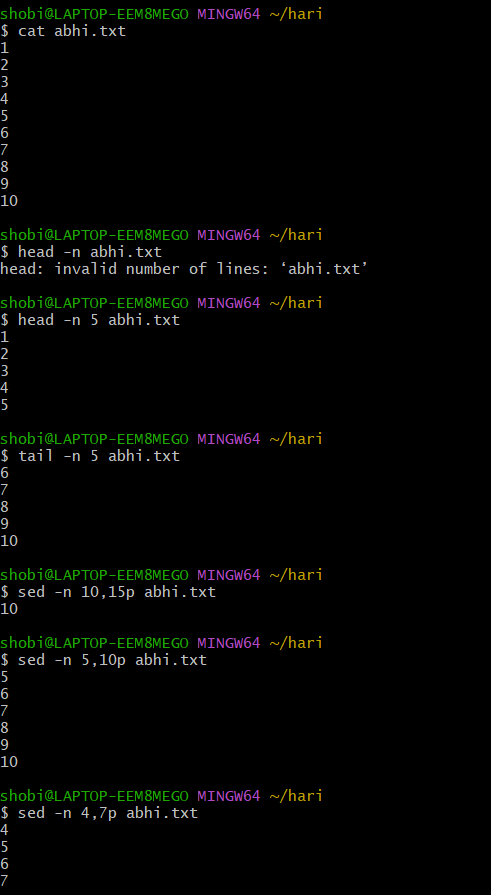
**Information about network:** ipconfig

**TO check the file is exist or not:**

Find path -name filename



**To print data from the files** :



**Scripts:**

#! /bin/bash

Echo “hi”

Sleep 2

Echo “hello”

Sleep 3

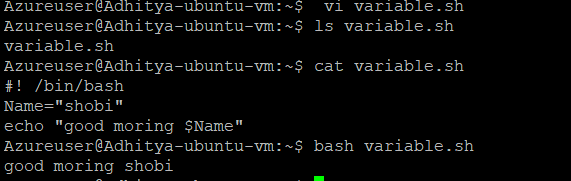
Echo “bye”

O/p : hi

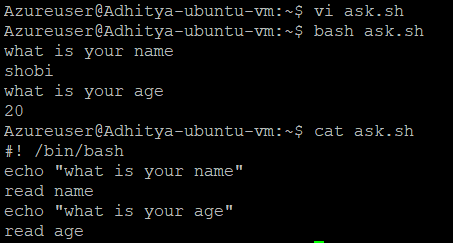
hello

bye

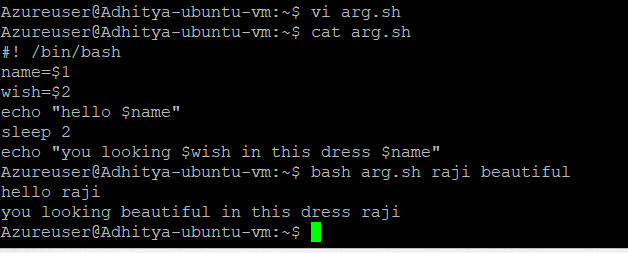
2) **variables:**



3)**Asking while executing**:



**Aruguments and parameters**:



**Predefined variables:**

