

HND CYBER SECURITY & NETWORKING

PROGRAMME	COMPUTING PORTFOLIO
MODULE	COMPUTER ARCHITECTURE AND OPERATING SYSTEMS
MODULE CODE	ACCA4022
LECTURER	JAMES WILLIAMS & MARK HUNTLY
STUDENT	ORIN MCGINN
STUDENT NUMBER	2304621
DATE DUE	19/04/24



TABLE OF CONTENTS

INTRODUCTION	3
KEY FUNCTIONS OF GENERAL OPERATING SYSTEMS.....	4
USE OF WINDOWS OS CLI COMMANDS.....	8
USE OF LINUX OS CLI COMMANDS	18
USE OF WINDOWS OS SCRIPTS	28
USE OF LINUS OS SCRIPTS.....	33
REFERENCE LIST	36
APPENDICES	37



INTRODUCTION

This project asked us to look at various operating systems and commands and key functions provided by some systems.

I also have provided examples of commands within these operating systems.

As part of the project I completed online courses, Cisco – Linux Unhatched.

KEY FUNCTIONS OF GENERAL OPERATING SYSTEMS

General-purpose operating systems (OS) are designed to manage computer hardware and software resources, providing a platform for users to execute programs efficiently and conveniently. The three primary functions of an OS are managing computer resources, establishing a user interface, and implementing and providing services for applications software.

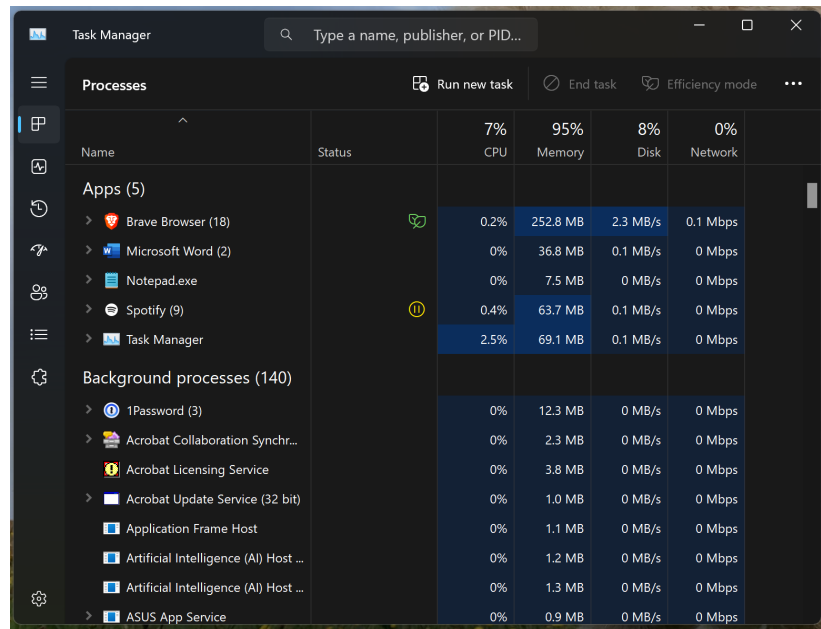
Popular general-purpose operating systems include Microsoft Windows, macOS, and Linux. Each OS has its unique features and capabilities, but they all share standard vital functions:

1. **Resource Management:** The OS manages and allocates hardware resources, such as the central processing unit (CPU), memory, disk drives, and printers. This ensures that each process receives enough resources to function correctly and prevents user programs from interfering with the proper working of the system.
2. **User Interface:** The OS provides a user-friendly interface, allowing users to interact with the computer system through commands or a graphical user interface (GUI). This interface enables users to communicate with applications and hardware.
3. **Process Management:** In a multi-programming environment, the OS decides the order in which processes can access the processor and how much processing time each process has. This function is called Process Scheduling. The OS keeps track of the status of processes and allocates the CPU to a process, deallocating it when the process is no longer required.
4. **Memory Management:** The OS allocates and deallocates memory to various processes, ensuring that one method does not consume the memory allocated to another.
5. **Device Management:** The OS manages device communication via its drivers. It keeps track of information regarding the creation, deletion, transfer, copy, and storage of files in an organised way.



Here are some screenshots showing where some of these functions can be seen, monitored, or managed in popular operating systems:

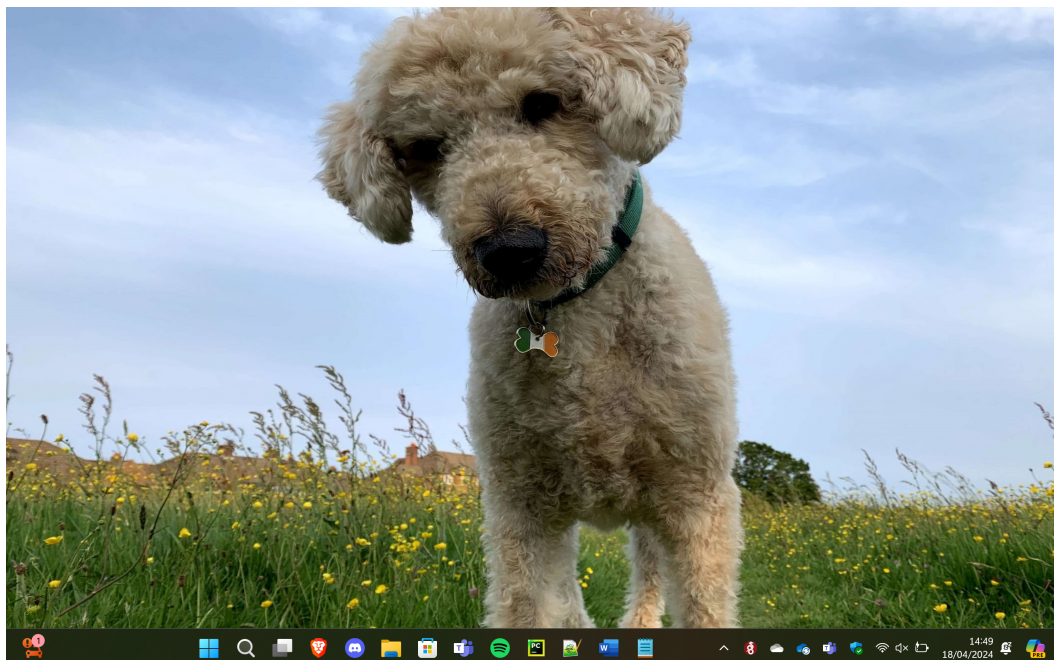
1. Resource Management: In Windows, you can monitor resource usage in the Task Manager (Ctrl+Shift+Esc). You can use the Activity Monitor on MacOS. In Linux, you can use the System Monitor.



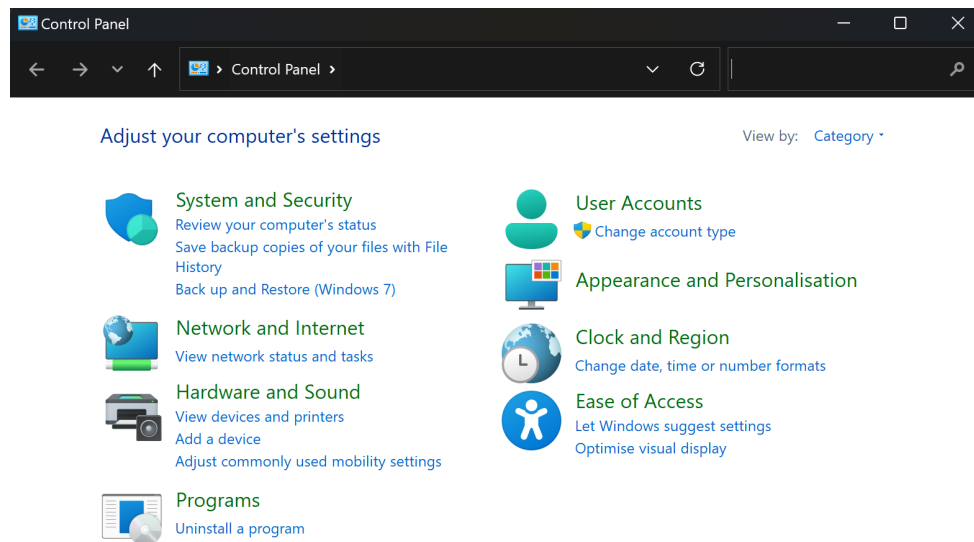
The screenshot shows the Windows Task Manager window with the 'Processes' tab selected. The window displays a list of running applications and background processes, along with their resource usage (CPU, Memory, Disk, and Network). The 'Apps' section lists 5 processes, and the 'Background processes' section lists 140 processes. The 'Task Manager' process is highlighted in blue.

Name	Status	7% CPU	95% Memory	8% Disk	0% Network
Apps (5)					
Brave Browser (18)		0.2%	252.8 MB	2.3 MB/s	0.1 Mbps
Microsoft Word (2)		0%	36.8 MB	0.1 MB/s	0 Mbps
Notepad.exe		0%	7.5 MB	0 MB/s	0 Mbps
Spotify (9)		0.4%	63.7 MB	0.1 MB/s	0 Mbps
Task Manager		2.5%	69.1 MB	0.1 MB/s	0 Mbps
Background processes (140)					
1Password (3)		0%	12.3 MB	0 MB/s	0 Mbps
Acrobat Collaboration Synchr...		0%	2.3 MB	0 MB/s	0 Mbps
Acrobat Licensing Service		0%	3.8 MB	0 MB/s	0 Mbps
Acrobat Update Service (32 bit)		0%	1.0 MB	0 MB/s	0 Mbps
Application Frame Host		0%	1.1 MB	0 MB/s	0 Mbps
Artificial Intelligence (AI) Host ...		0%	1.2 MB	0 MB/s	0 Mbps
Artificial Intelligence (AI) Host ...		0%	1.3 MB	0 MB/s	0 Mbps
ASUS App Service		0%	0.9 MB	0 MB/s	0 Mbps

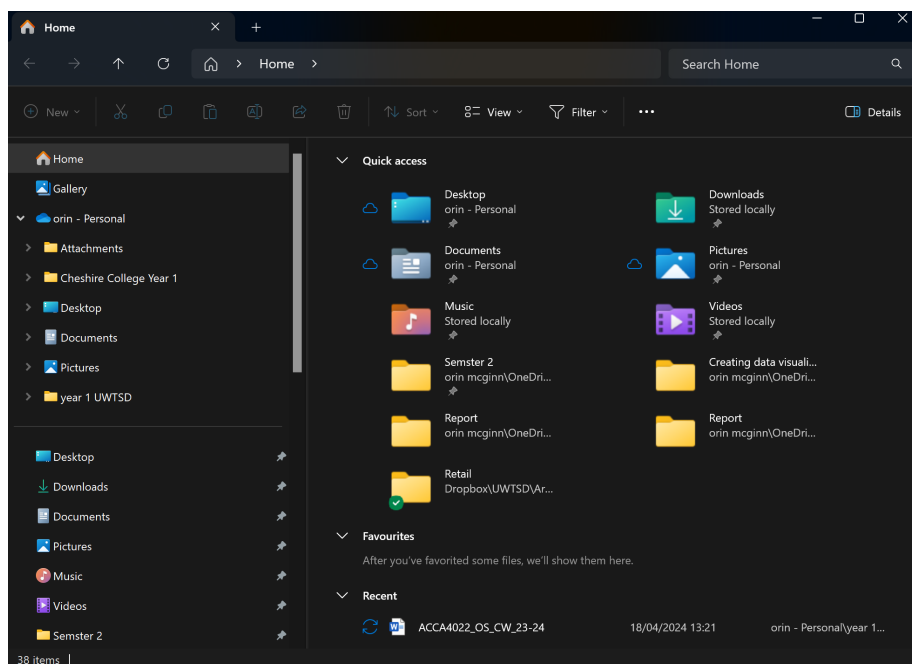
2. User Interface: The desktop and start menu in Windows, the Dock and Finder in macOS, and the desktop environment in Linux are examples of user interfaces.



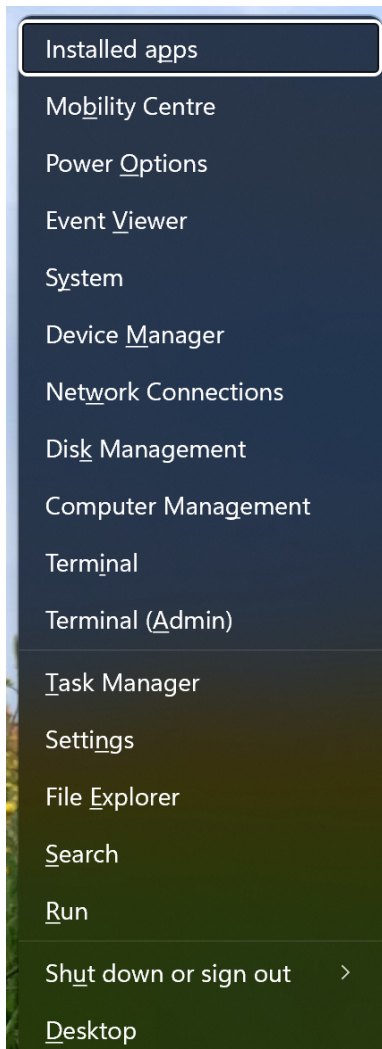
3. Security: In Windows, you can manage security settings in the Control Panel or Settings app. In macOS, you can use the System Preferences. You can use the command line or a graphical tool like the Users and Groups application in Linux.



4. File Management: In Windows, you can manage files in File Explorer. In macOS, you can use the Finder. You can use the file manager for your desktop environment or the command line in Linux.



5. Device Management: In Windows, you can manage devices in the Device Manager (Windows+X, then M). You can use the System Report (Apple menu > About This Mac > System Report) in macOS. You can use the command line or a graphical tool like the Disks application in Linux.

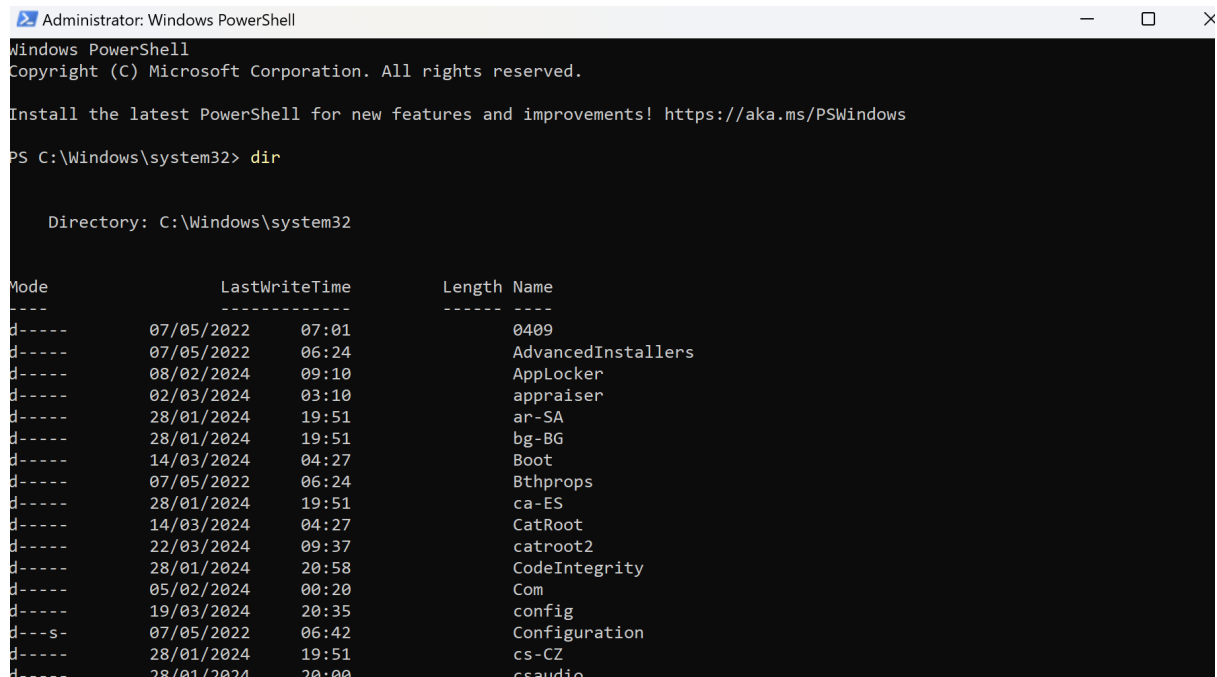


USE OF WINDOWS OS CLI COMMANDS

EXAMPLE 1

COMMAND: dir

OUTCOME: With this bash command, I am displaying the list of contents in the current directory for every single one. DLL file within system32.



```

Administrator: Windows PowerShell
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> dir

    Directory: C:\Windows\system32

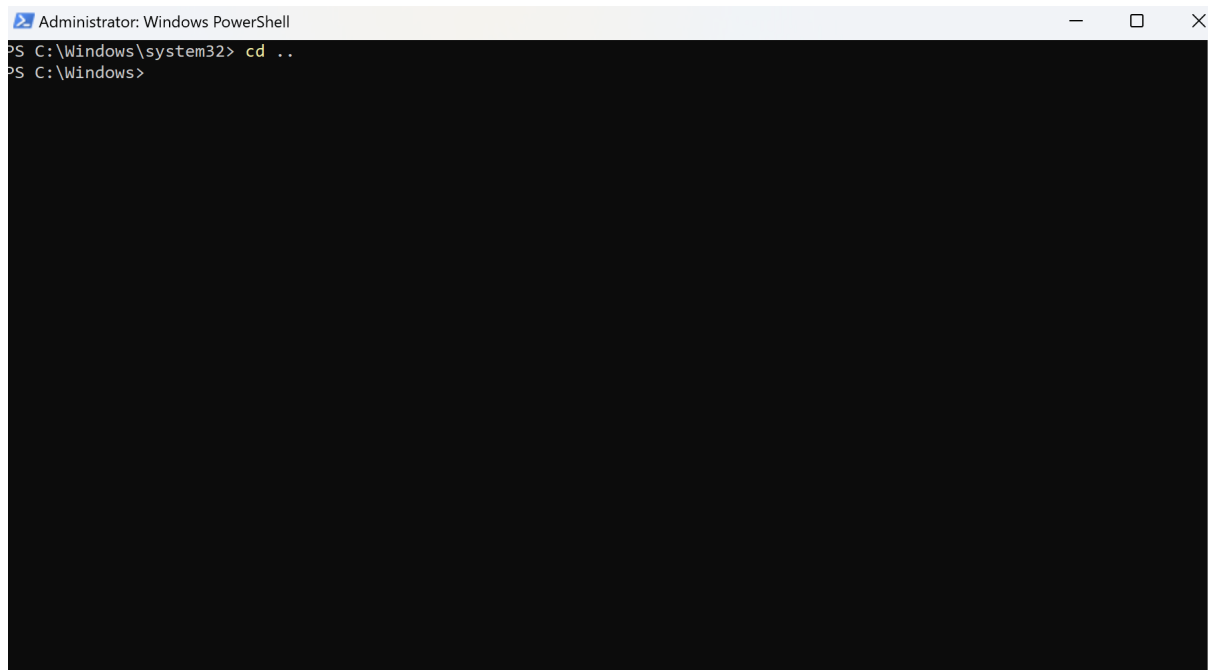
Mode                LastWriteTime         Length Name
----                -
d-----         07/05/2022         07:01          0409
d-----         07/05/2022         06:24      AdvancedInstallers
d-----         08/02/2024         09:10      AppLocker
d-----         02/03/2024         03:10      appraiser
d-----         28/01/2024         19:51      ar-SA
d-----         28/01/2024         19:51      bg-BG
d-----         14/03/2024         04:27      Boot
d-----         07/05/2022         06:24      Bthprops
d-----         28/01/2024         19:51      ca-ES
d-----         14/03/2024         04:27      CatRoot
d-----         22/03/2024         09:37      catroot2
d-----         28/01/2024         20:58      CodeIntegrity
d-----         05/02/2024         00:20      Com
d-----         19/03/2024         20:35      config
d---s-         07/05/2022         06:42      Configuration
d-----         28/01/2024         19:51      cs-CZ
d-----         28/01/2024         20:00      csaudio
  
```



EXAMPLE 2

COMMAND: cd ..

OUTCOME: Using CD space with .. I have gone back to a directory from C:\Windows\system32> to C:\Windows, meaning if I don't need to access any files in the system, I can go back to accessing the Windows files.



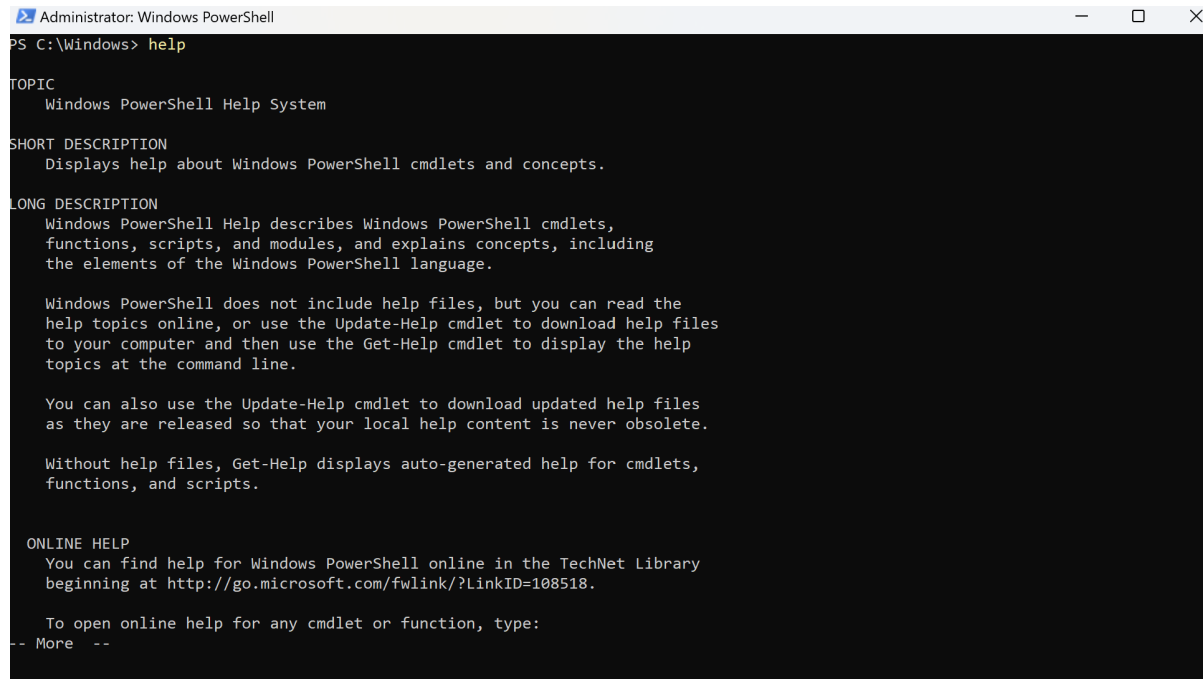
```
Administrator: Windows PowerShell
PS C:\Windows\system32> cd ..
PS C:\Windows>
```



EXAMPLE 3

COMMAND: help

OUTCOME: The help command is straightforward and helpful to use in any terminal. Different OS terminals will have different help messages giving you information.

A screenshot of a Windows PowerShell window titled 'Administrator: Windows PowerShell'. The command 'PS C:\Windows> help' has been entered. The output is as follows:

```
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.

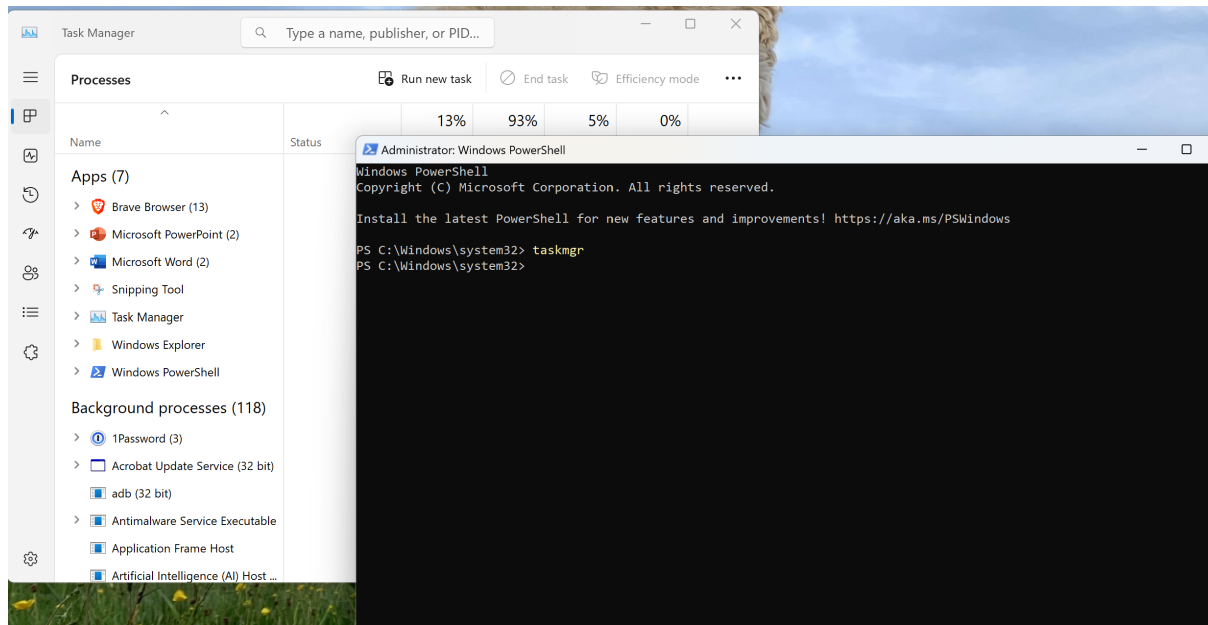
    To open online help for any cmdlet or function, type:
-- More --
```



EXAMPLE 4

COMMAND: taskmgr

OUTCOME: when you enter It's into PowerShell it pulls up the task manager Application



EXAMPLE 5

COMMAND: tasklist

OUTCOME: Displays a list of currently running processes, including the process ID, name, and memory usage.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>tasklist
```

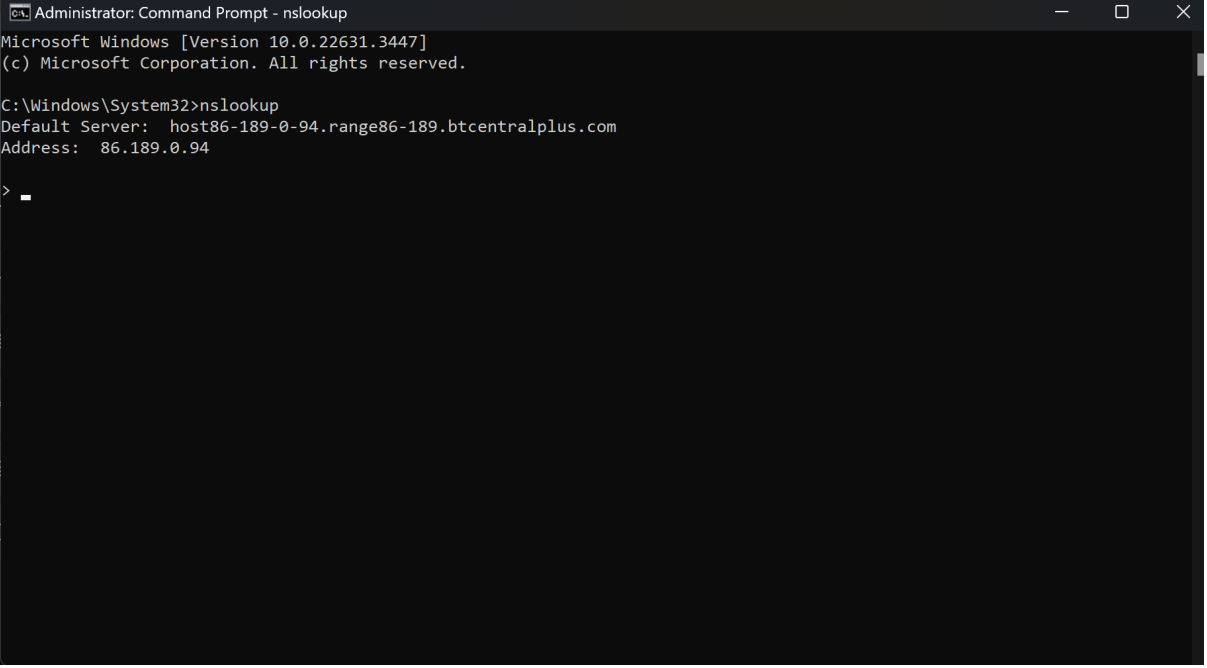
Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	1,544 K
Secure System	172	Services	0	32,884 K
Registry	236	Services	0	15,272 K
smss.exe	1224	Services	0	24 K
csrss.exe	1620	Services	0	2,100 K
wininit.exe	1716	Services	0	4 K
csrss.exe	1740	Console	1	2,464 K
services.exe	1796	Services	0	6,276 K
winlogon.exe	1828	Console	1	3,316 K
LsaIso.exe	1884	Services	0	4 K
lsass.exe	1900	Services	0	16,976 K
svchost.exe	2012	Services	0	16,700 K
fontdrvhost.exe	2040	Services	0	68 K
fontdrvhost.exe	1280	Console	1	1,808 K
WUDFHost.exe	1312	Services	0	1,812 K
svchost.exe	2052	Services	0	15,636 K
svchost.exe	2100	Services	0	3,356 K
dwm.exe	2196	Console	1	147,320 K
svchost.exe	2232	Services	0	984 K
svchost.exe	2248	Services	0	1,476 K
svchost.exe	2276	Services	0	2,784 K
svchost.exe	2452	Services	0	1,076 K



EXAMPLE 6

COMMAND: nslookup

OUTCOME: Queries the DNS server for the IP address associated with a specified hostname, or the hostname associated with a specified IP address.



```
Administrator: Command Prompt - nslookup
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>nslookup
Default Server: host86-189-0-94.range86-189.btcentralplus.com
Address: 86.189.0.94

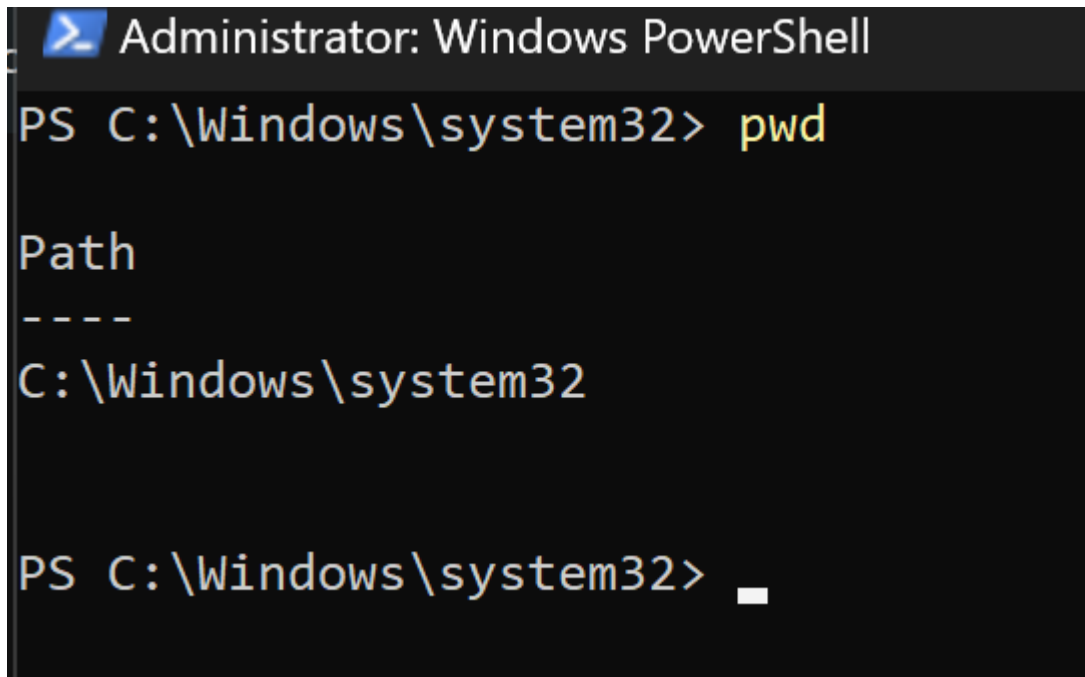
> -
```



EXAMPLE 7

COMMAND: pwd

OUTCOME: The pwd Command shows the current path of your command line if you were to do cd .. Followed by a pwd again It would show you a different path so in this picture here you can see what in C drive windows system 32 but if I was to use cd .. And then pwd it would only come up as C drive than windows out of the folder system 32.



```
Administrator: Windows PowerShell
PS C:\Windows\system32> pwd

Path
----
C:\Windows\system32

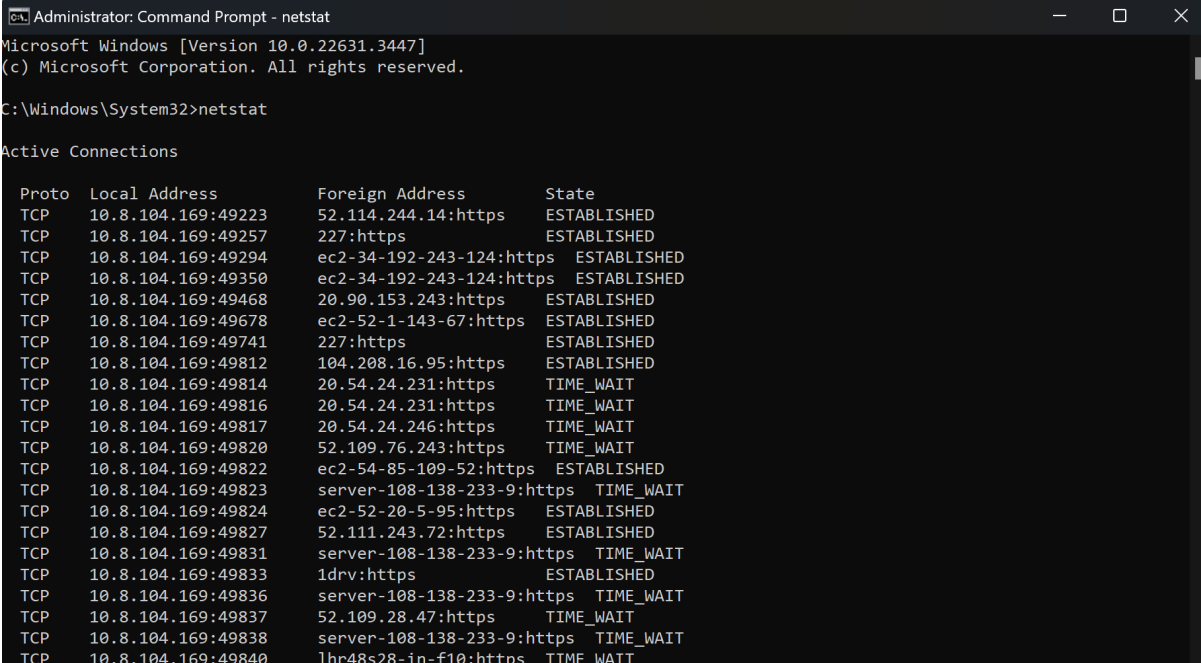
PS C:\Windows\system32> _
```



EXAMPLE 8

COMMAND: netstat

OUTCOME: Displays active network connections, including the protocol, local and foreign addresses, and connection state.



```
Administrator: Command Prompt - netstat
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>netstat

Active Connections

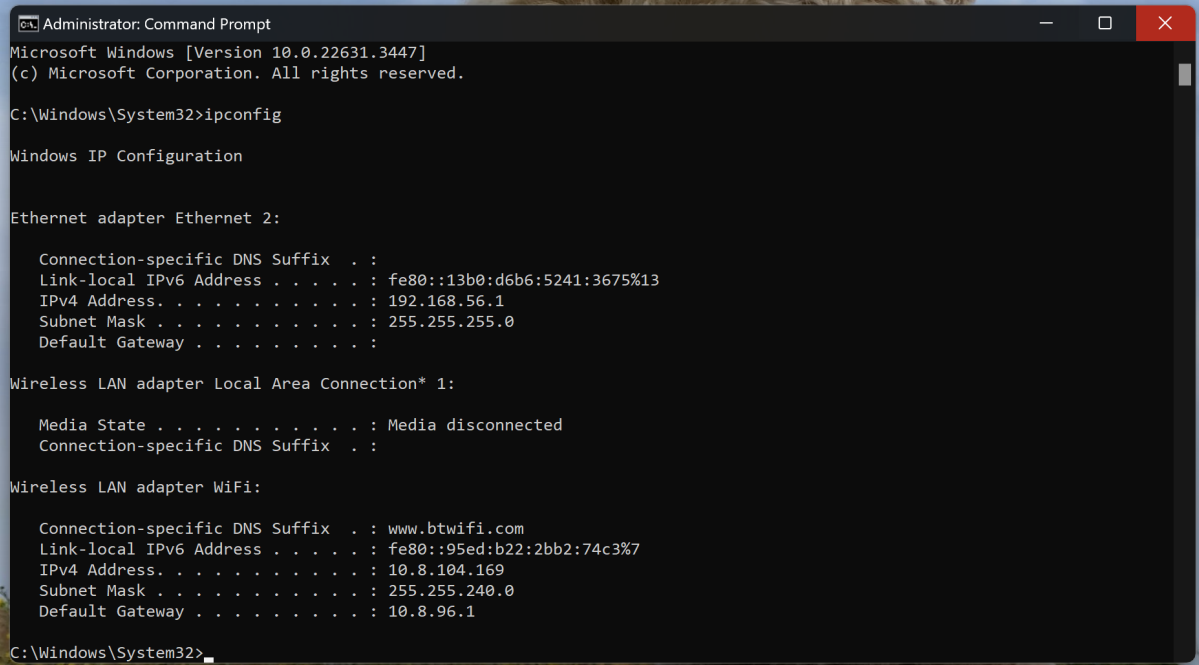
Proto Local Address           Foreign Address         State
TCP    10.8.104.169:49223       52.114.244.14:https     ESTABLISHED
TCP    10.8.104.169:49257       227:https               ESTABLISHED
TCP    10.8.104.169:49294       ec2-34-192-243-124:https ESTABLISHED
TCP    10.8.104.169:49350       ec2-34-192-243-124:https ESTABLISHED
TCP    10.8.104.169:49468       20.90.153.243:https     ESTABLISHED
TCP    10.8.104.169:49678       ec2-52-1-143-67:https   ESTABLISHED
TCP    10.8.104.169:49741       227:https               ESTABLISHED
TCP    10.8.104.169:49812       104.208.16.95:https     ESTABLISHED
TCP    10.8.104.169:49814       20.54.24.231:https      TIME_WAIT
TCP    10.8.104.169:49816       20.54.24.231:https      TIME_WAIT
TCP    10.8.104.169:49817       20.54.24.246:https      TIME_WAIT
TCP    10.8.104.169:49820       52.109.76.243:https     TIME_WAIT
TCP    10.8.104.169:49822       ec2-54-85-109-52:https   ESTABLISHED
TCP    10.8.104.169:49823       server-108-138-233-9:https TIME_WAIT
TCP    10.8.104.169:49824       ec2-52-20-5-95:https    ESTABLISHED
TCP    10.8.104.169:49827       52.111.243.72:https     ESTABLISHED
TCP    10.8.104.169:49831       server-108-138-233-9:https TIME_WAIT
TCP    10.8.104.169:49833       1drv:https              ESTABLISHED
TCP    10.8.104.169:49836       server-108-138-233-9:https TIME_WAIT
TCP    10.8.104.169:49837       52.109.28.47:https      TIME_WAIT
TCP    10.8.104.169:49838       server-108-138-233-9:https TIME_WAIT
TCP    10.8.104.169:49840       lhr48s28-in-f10:https   TIME_WAIT
```



EXAMPLE 9

COMMAND: ipconfig

OUTCOME: Displays the IP configuration for all network adapters, including IP address, subnet mask, and default gateway.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::13b0:d6b6:5241:3675%13
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Wireless LAN adapter Wifi:

    Connection-specific DNS Suffix  . : www.btwifi.com
    Link-local IPv6 Address . . . . . : fe80::95ed:b22:2bb2:74c3%7
    IPv4 Address. . . . . : 10.8.104.169
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 10.8.96.1

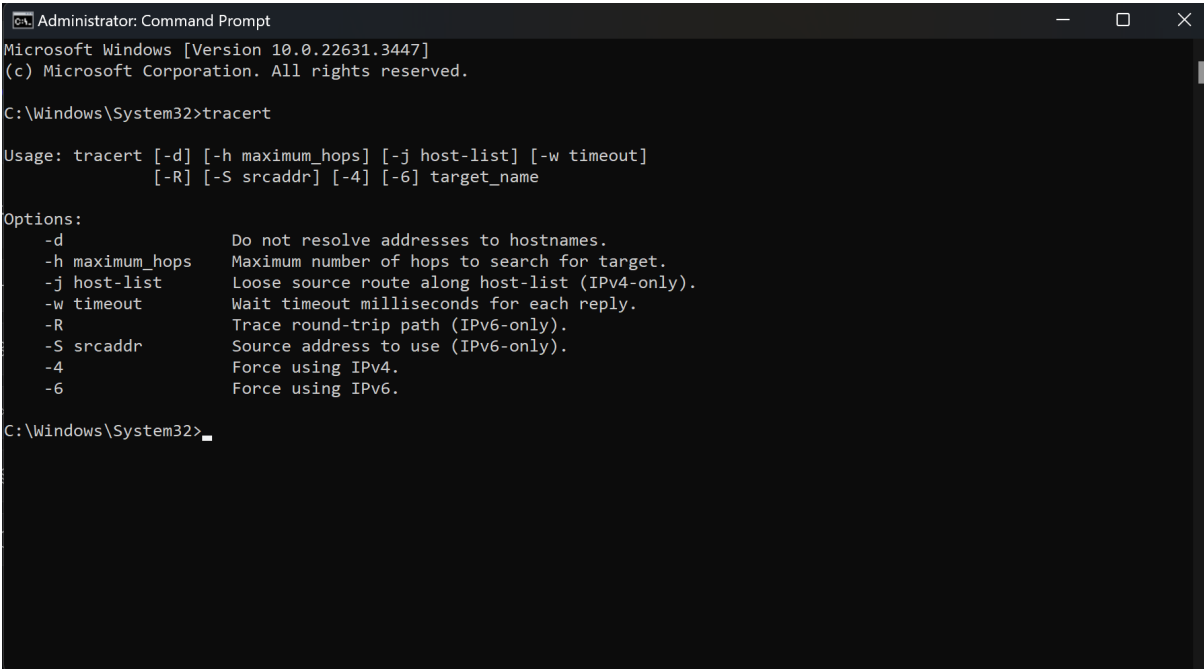
C:\Windows\System32>
```



EXAMPLE 10

COMMAND: tracert

OUTCOME: Traces the route packets take to reach a specified IP address or hostname, displaying each hop's IP address and response time.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
              [-R] [-S srcaddr] [-4] [-6] target_name

Options:
  -d                Do not resolve addresses to hostnames.
  -h maximum_hops   Maximum number of hops to search for target.
  -j host-list       Loose source route along host-list (IPv4-only).
  -w timeout         Wait timeout milliseconds for each reply.
  -R                Trace round-trip path (IPv6-only).
  -S srcaddr         Source address to use (IPv6-only).
  -4                Force using IPv4.
  -6                Force using IPv6.

C:\Windows\System32>
```

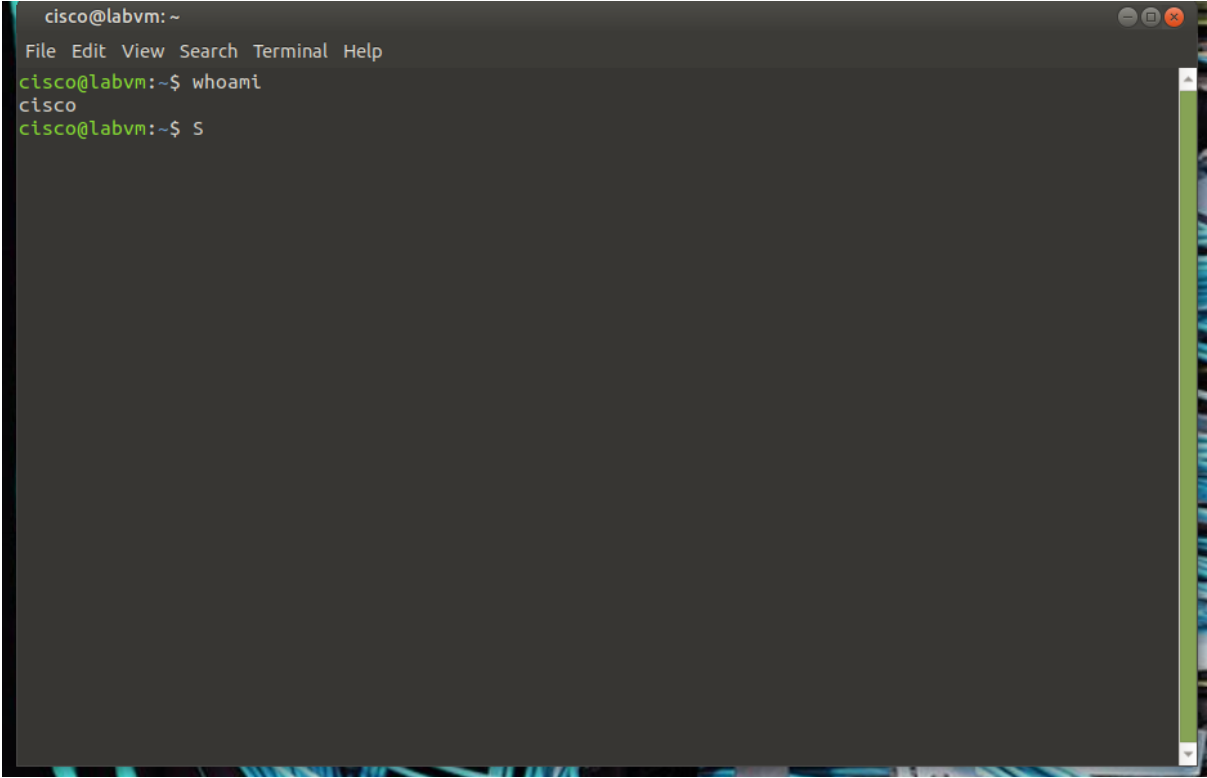


USE OF LINUX OS CLI COMMANDS

EXAMPLE 1

COMMAND: whoami

OUTCOME: This command allows the user to see the currently logged in user. It is also helpful in bash scripting to show who is running a script. The output of this command displays the username.

A screenshot of a Linux terminal window. The window title is 'cisco@labvm: ~'. The menu bar shows 'File Edit View Search Terminal Help'. The terminal shows the command 'whoami' being executed, with the output 'cisco' displayed on the next line. The prompt 'cisco@labvm:~\$' is visible at the bottom.

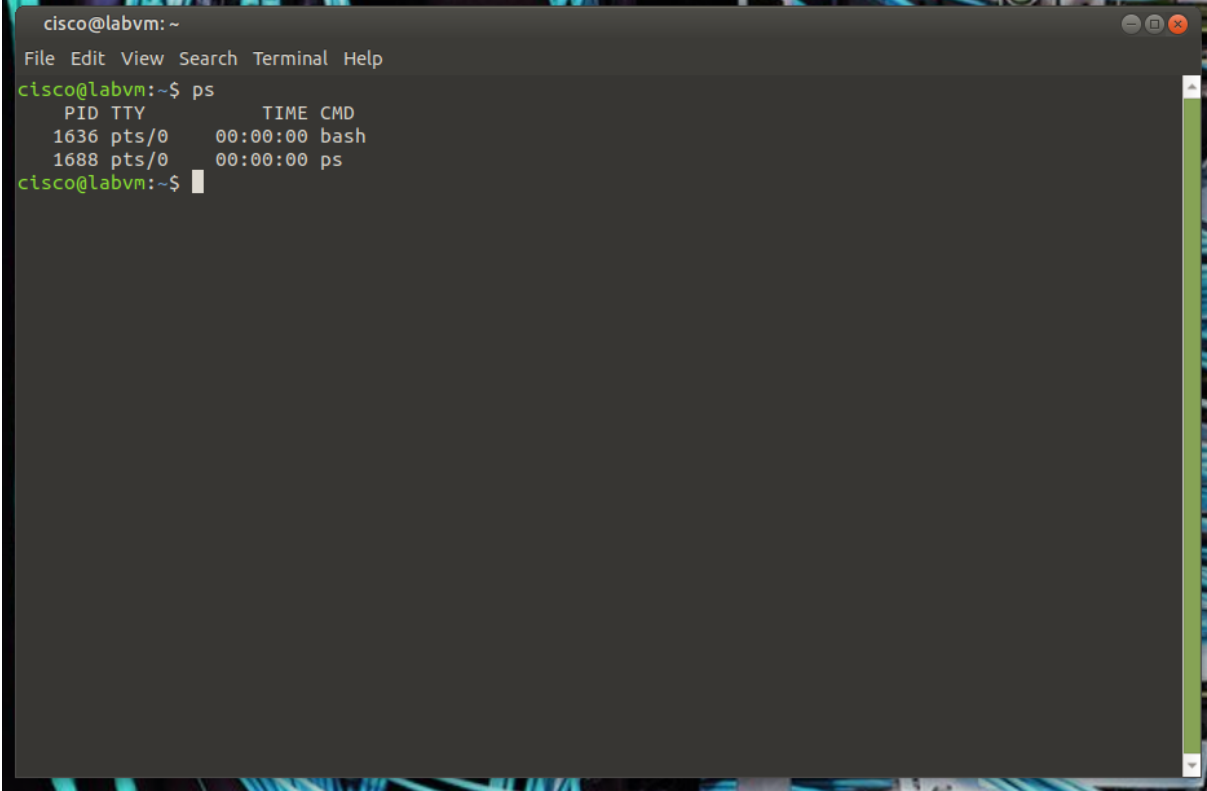
```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ whoami  
cisco  
cisco@labvm:~$ S
```



EXAMPLE 2

COMMAND: PS

OUTCOME: This command shows the user programs running within the system it also allows the user to manage and monitor memory I/O resources and processor times

A screenshot of a terminal window titled 'cisco@labvm: ~'. The terminal shows the command 'ps' being executed. The output is a table with columns: PID, TTY, TIME, and CMD. The first row shows PID 1636, TTY pts/0, TIME 00:00:00, and CMD bash. The second row shows PID 1688, TTY pts/0, TIME 00:00:00, and CMD ps. The prompt 'cisco@labvm:~\$' is visible at the bottom.

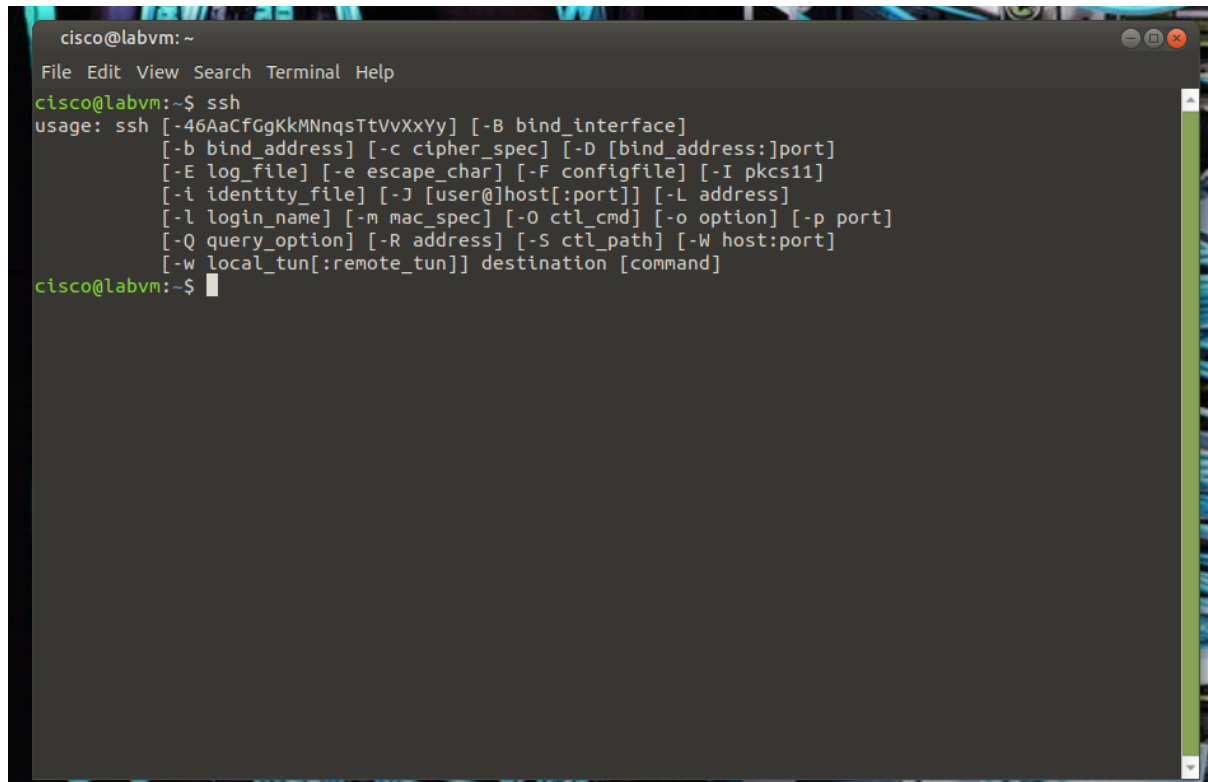
```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ ps  
  PID TTY          TIME CMD  
 1636 pts/0    00:00:00 bash  
 1688 pts/0    00:00:00 ps  
cisco@labvm:~$
```



EXAMPLE 3

COMMAND: SSH

OUTCOME: ssh also known as secure shell is a method for allowing two computers to communicate using this network communication protocol

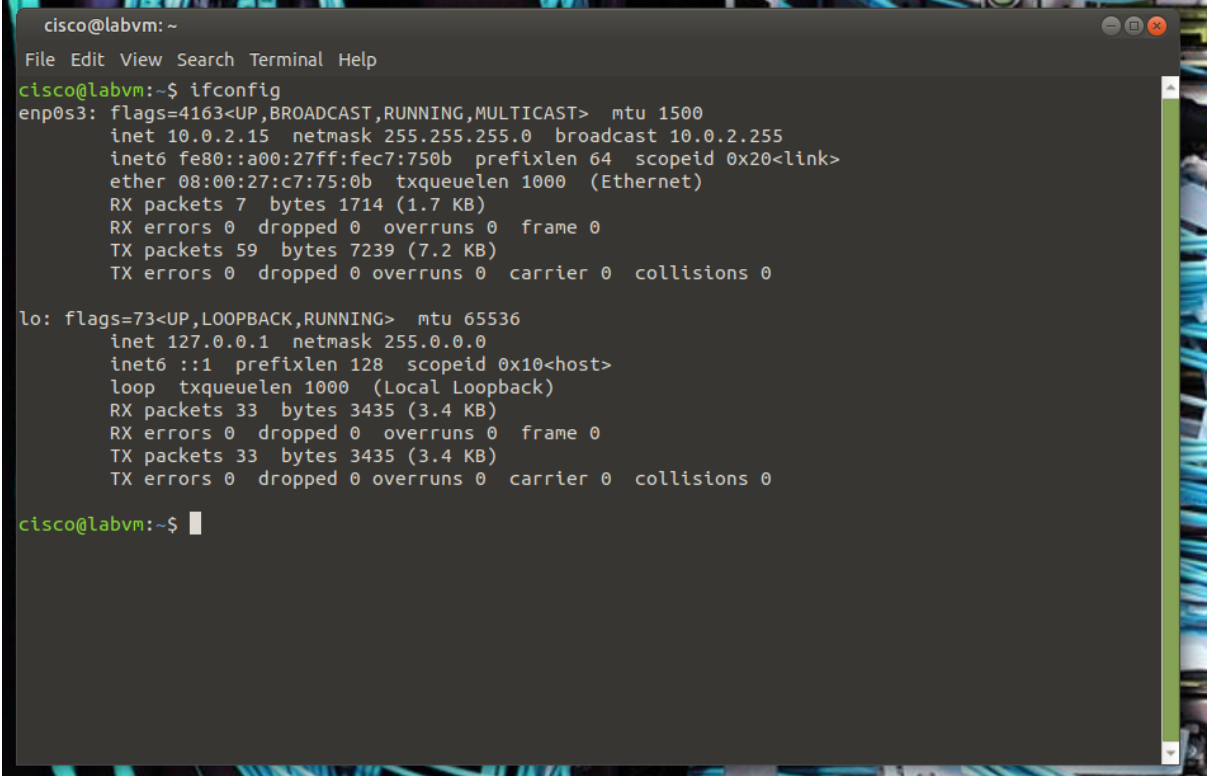
A screenshot of a terminal window titled 'cisco@labvm: ~'. The terminal shows the command 'ssh' being entered, followed by its usage information. The usage text lists various options for the ssh command, including bind interface, bind address, cipher spec, log file, escape char, configfile, pkcs11, identity file, user@host:port, address, login name, mac spec, ctl cmd, option, port, query option, address, ctl path, host:port, local_tun:remote_tun, destination, and command. The terminal prompt is 'cisco@labvm:~\$'.

EXAMPLE 4

COMMAND: ifconfig

OUTCOME: This is a system command that is used at system startup that defines the network address for each network interface

This command can also be used to change network settings for a network interface.

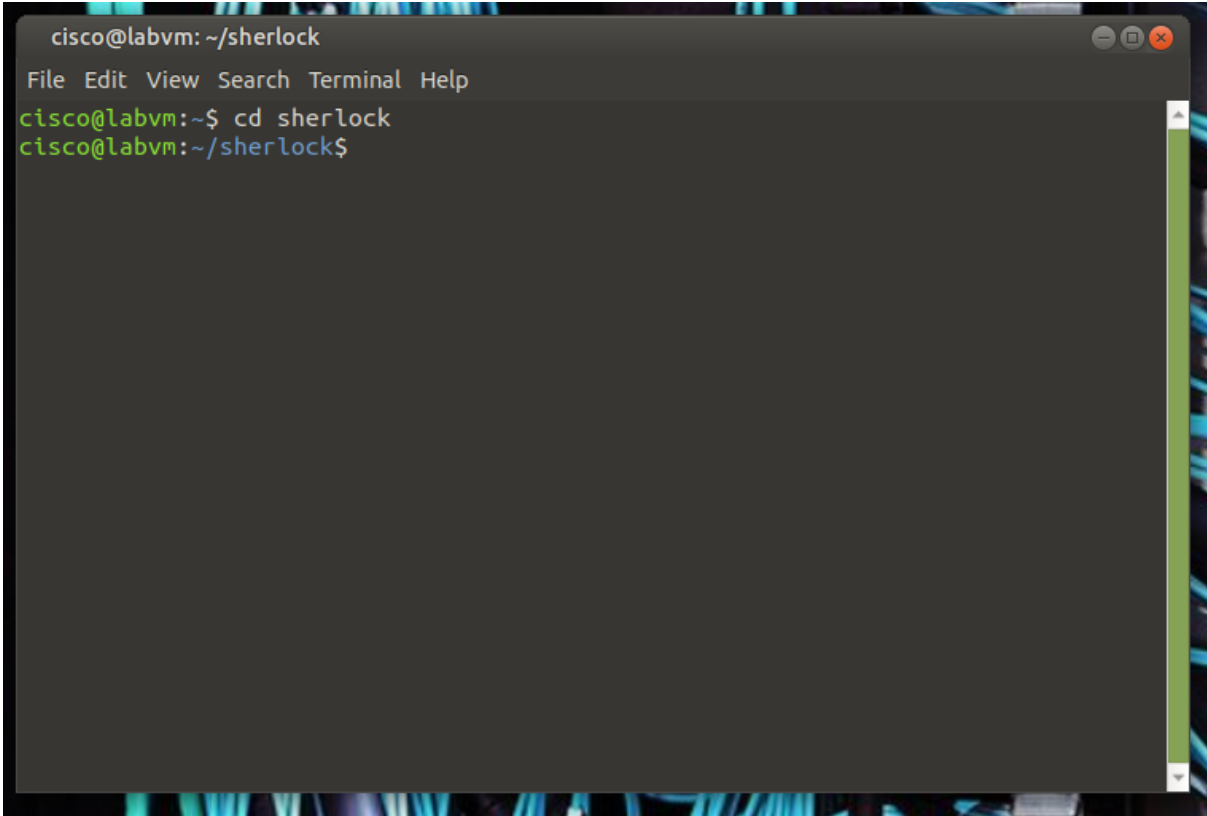


```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::a00:27ff:fec7:750b prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:c7:75:0b txqueuelen 1000 (Ethernet)  
    RX packets 7 bytes 1714 (1.7 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 59 bytes 7239 (7.2 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 33 bytes 3435 (3.4 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 33 bytes 3435 (3.4 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
cisco@labvm:~$
```

EXAMPLE 5

COMMAND: cd (chdir)

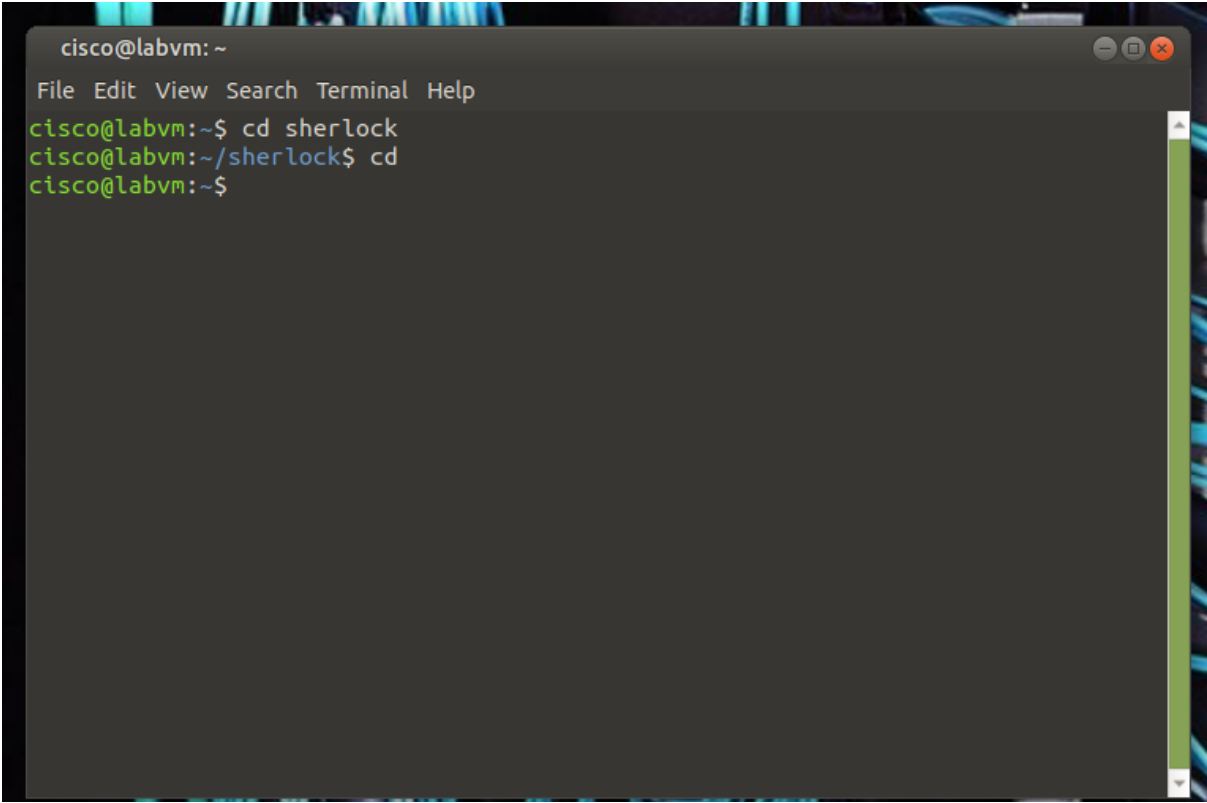
OUTCOME: This is one of the most basic Linux commands. It simply allows the user to change directories within the operating system

A screenshot of a Linux terminal window. The window title is 'cisco@labvm: ~/sherlock'. The menu bar shows 'File Edit View Search Terminal Help'. The terminal text shows the prompt 'cisco@labvm:~\$' followed by the command 'cd sherlock', and then the prompt 'cisco@labvm:~/sherlock\$' on the next line. The terminal has a dark background with green text. The window has standard Linux window controls (minimize, maximize, close) in the top right corner.

EXAMPLE 6

COMMAND: cd (chdir)

OUTCOME: The use of the cd command alone lets the user step one directory up.



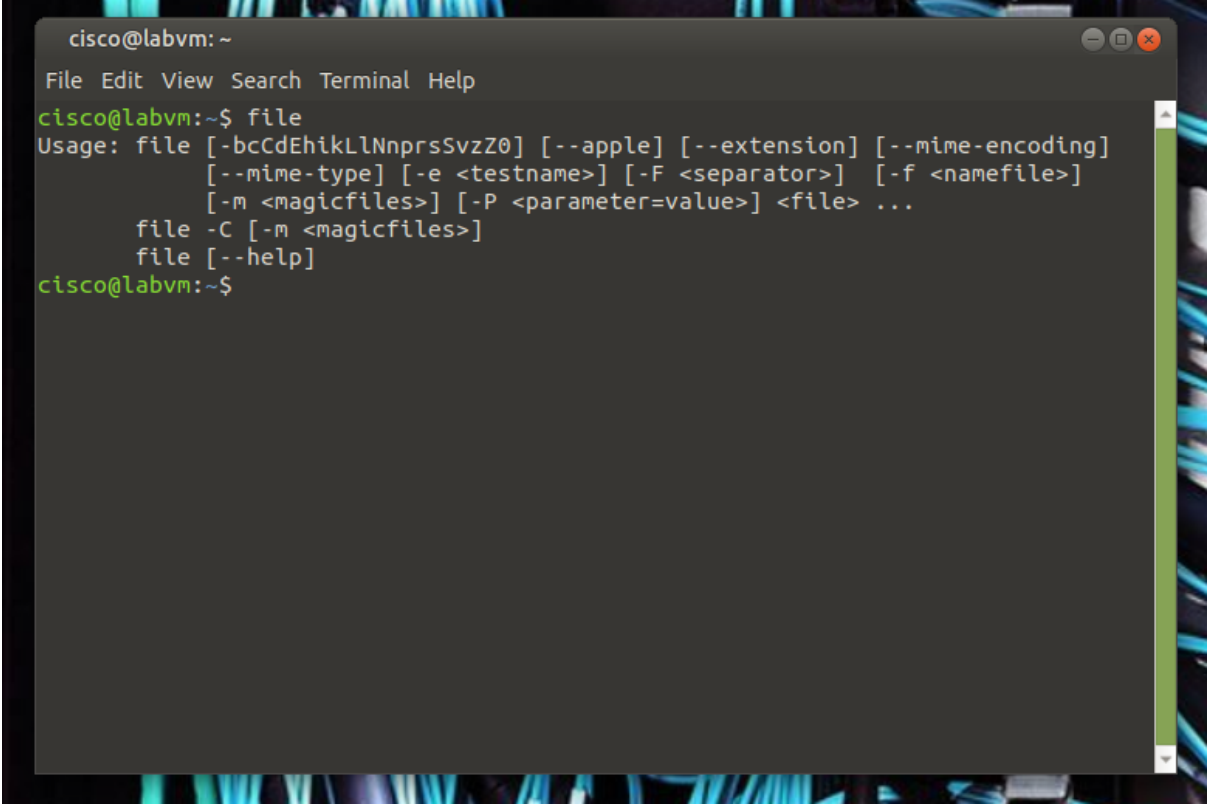
```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ cd sherlock  
cisco@labvm:~/sherlock$ cd  
cisco@labvm:~$
```



EXAMPLE 7

COMMAND: file

OUTCOME: This command allows the user to receive output from a file as required

A screenshot of a terminal window titled 'cisco@labvm: ~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command 'cisco@labvm:~\$ file' being entered. The output displays the usage for the 'file' command, including various options like '--apple', '--extension', '--mime-encoding', '--mime-type', '-e <testname>', '-F <separator>', '-f <namefile>', '-m <magicfiles>', '-P <parameter=value>', and '-C [-m <magicfiles>]'. It also shows 'file --help' at the bottom. The prompt 'cisco@labvm:~\$' is visible at the end of the output.

```
cisco@labvm: ~
File Edit View Search Terminal Help
cisco@labvm:~$ file
Usage: file [-bCdEhikLlNnprSsvzZ0] [--apple] [--extension] [--mime-encoding]
        [--mime-type] [-e <testname>] [-F <separator>] [-f <namefile>]
        [-m <magicfiles>] [-P <parameter=value>] <file> ...
        file -C [-m <magicfiles>]
        file [--help]
cisco@labvm:~$
```

EXAMPLE 8

COMMAND: apt

OUTCOME: This command is used to install and remove packages. Because this is an administrative task, it has to be used with the sudo command.

```
File Machine View Input Devices Help
cisco@labvm: ~
File Edit View Search Terminal Help
cisco@labvm:~$ locate

Command 'locate' not found, but can be installed with:

sudo apt install mlocate

cisco@labvm:~$ sudo apt install mlocate
[sudo] password for cisco:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  liblvm1
Use 'sudo apt autoremove' to remove it.
Suggested packages:
  ncache
The following NEW packages will be installed:
  mlocate
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 50.1 kB of archives.
After this operation, 258 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 mlocate amd64 0.26-3ubuntu3 [50.1 kB]
Fetched 50.1 kB in 0s (103 kB/s)
Selecting previously unselected package mlocate.
(Reading database ... 245578 files and directories currently installed.)
Preparing to unpack .../mlocate_0.26-3ubuntu3_amd64.deb ...
Unpacking mlocate (0.26-3ubuntu3) ...
Setting up mlocate (0.26-3ubuntu3) ...
update-alternatives: using /usr/bin/mlocate to provide /usr/bin/locate (locate)
in auto mode
Adding group 'mlocate' (GID 134) ...
Done.
Initializing mlocate database; this may take some time... done
Processing triggers for man-db (2.9.1-1) ...#####.....]
cisco@labvm:~$
```



EXAMPLE 9

COMMAND: netstat

OUTCOME: This gives an overview of all network activities and displays network connections, open ports, and other essential network data.

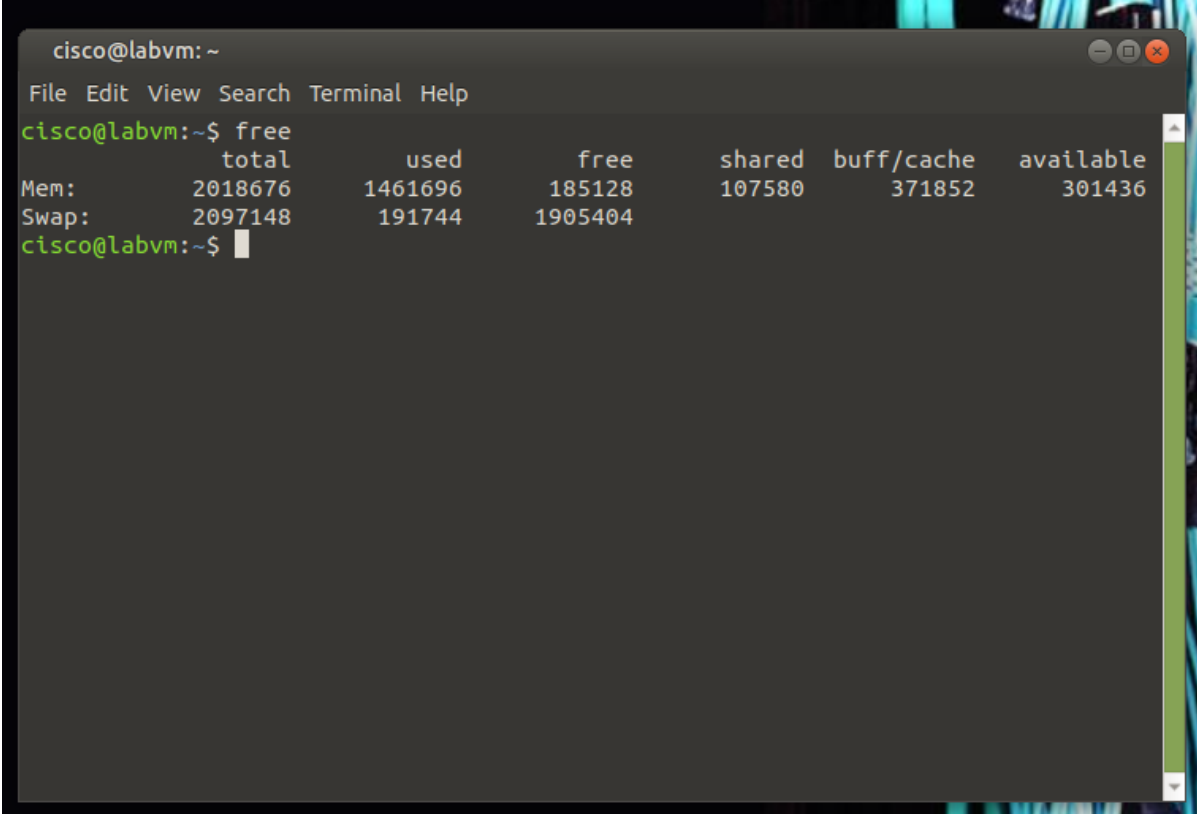
```
File Machine View Input Devices Help
cisco@labvm: ~
File Edit View Search Terminal Help
cisco@labvm:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags               Type                   State                  I-Node    Path
unix  2      [ ] DGRAM                24703                   /run/user/900/systemd/notify
unix  2      [ ] DGRAM                14704                   /run/systemd/journal/syslog
unix 16      [ ] DGRAM                14714                   /run/systemd/journal/dev-log
unix  8      [ ] DGRAM                14718                   /run/systemd/journal/socket
unix  3      [ ] DGRAM                14690                   /run/systemd/notify
unix  3      [ ] STREAM              CONNECTED               22100                   /run/dbus/system_bus_socket
unix  3      [ ] STREAM              CONNECTED               25324                   /run/systemd/journal/stdout
unix  3      [ ] STREAM              CONNECTED               25274
unix  3      [ ] STREAM              CONNECTED               25230                   @/tmp/.X11-unix/X0
unix  3      [ ] STREAM              CONNECTED               22102
unix  3      [ ] STREAM              CONNECTED               18306
unix  3      [ ] STREAM              CONNECTED               26344
unix  3      [ ] STREAM              CONNECTED               21216
unix  3      [ ] STREAM              CONNECTED               23447                   /run/systemd/journal/stdout
unix  3      [ ] STREAM              CONNECTED               21341                   /run/dbus/system_bus_socket
unix  3      [ ] STREAM              CONNECTED               24501                   @/tmp/.ICE-unix/983
unix  3      [ ] STREAM              CONNECTED               21709                   /run/dbus/system_bus_socket
unix  2      [ ] DGRAM                17785
unix  3      [ ] STREAM              CONNECTED               27355                   /run/user/900/pulse/native
unix  3      [ ] STREAM              CONNECTED               25987                   /run/dbus/system_bus_socket
unix  3      [ ] STREAM              CONNECTED               25557                   /run/user/900/bus
unix  3      [ ] STREAM              CONNECTED               24403
unix  3      [ ] STREAM              CONNECTED               24232                   @/tmp/.X11-unix/X0
unix  3      [ ] STREAM              CONNECTED               27262
unix  3      [ ] STREAM              CONNECTED               26019
unix  3      [ ] STREAM              CONNECTED               25225
unix  3      [ ] STREAM              CONNECTED               24816
unix  3      [ ] STREAM              CONNECTED               21110                   /run/systemd/journal/stdout
unix  3      [ ] STREAM              CONNECTED               16381                   /run/systemd/journal/stdout
unix  2      [ ] DGRAM                25039
unix  3      [ ] STREAM              CONNECTED               26956                   /run/user/900/bus
unix  2      [ ] DGRAM                23807
unix  3      [ ] STREAM              CONNECTED               29409
unix  3      [ ] STREAM              CONNECTED               26689                   /run/user/900/pulse/native
unix  3      [ ] STREAM              CONNECTED               25277
unix  3      [ ] STREAM              CONNECTED               25218                   /run/systemd/journal/stdout
unix  3      [ ] STREAM              CONNECTED               26857                   @/tmp/.X11-unix/X0
unix  3      [ ] STREAM              CONNECTED               25894
unix  2      [ ] DGRAM                24440
unix  3      [ ] STREAM              CONNECTED               24344
unix  3      [ ] STREAM              CONNECTED               21118
unix  3      [ ] STREAM              CONNECTED               20903
unix  3      [ ] STREAM              CONNECTED               24586                   /run/dbus/system_bus_socket
unix  3      [ ] STREAM              CONNECTED               24222                   @/tmp/.X11-unix/X0
unix  3      [ ] STREAM              CONNECTED               18563
unix  3      [ ] STREAM              CONNECTED               27325
unix  3      [ ] STREAM              CONNECTED               24402
unix  3      [ ] STREAM              CONNECTED               24345                   /run/user/900/bus
unix  3      [ ] STREAM              CONNECTED               21906                   /run/systemd/journal/stdout
unix  3      [ ] STREAM              CONNECTED               26418                   /run/systemd/journal/stdout
unix  3      [ ] STREAM              CONNECTED               26942                   /run/user/900/bus
unix  3      [ ] STREAM              CONNECTED               25098
unix  3      [ ] STREAM              CONNECTED               24350                   /run/systemd/journal/stdout
unix  2      [ ] DGRAM                18281
unix  3      [ ] STREAM              CONNECTED               25841
unix  3      [ ] STREAM              CONNECTED               22105
unix  2      [ ] DGRAM                15543
unix  3      [ ] STREAM              CONNECTED               27068                   /run/user/900/bus
unix  3      [ ] STREAM              CONNECTED               26364
unix  3      [ ] STREAM              CONNECTED               25414                   @/tmp/dbus-jvH0zPRvyH
unix  3      [ ] STREAM              CONNECTED               21217                   /run/dbus/system_bus_socket
unix  3      [ ] STREAM              CONNECTED               25270
unix  3      [ ] STREAM              CONNECTED               25229                   /run/user/900/bus
unix  3      [ ] STREAM              CONNECTED               25672
unix  3      [ ] STREAM              CONNECTED               21297
unix  3      [ ] STREAM              CONNECTED               80517                   @/tmp/dbus-jvH0zPRvyH
unix  2      [ ] DGRAM                62968
```



EXAMPLE 10

COMMAND: free

OUTCOME: This allows you to check for memory RAM on your system or the memory statics of the Linux operating system.



The screenshot shows a terminal window titled 'cisco@labvm: ~'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The user has entered the command 'free', and the output is displayed as follows:

	total	used	free	shared	buff/cache	available
Mem:	2018676	1461696	185128	107580	371852	301436
Swap:	2097148	191744	1905404			

The prompt 'cisco@labvm:~\$' is shown at the bottom of the terminal window.

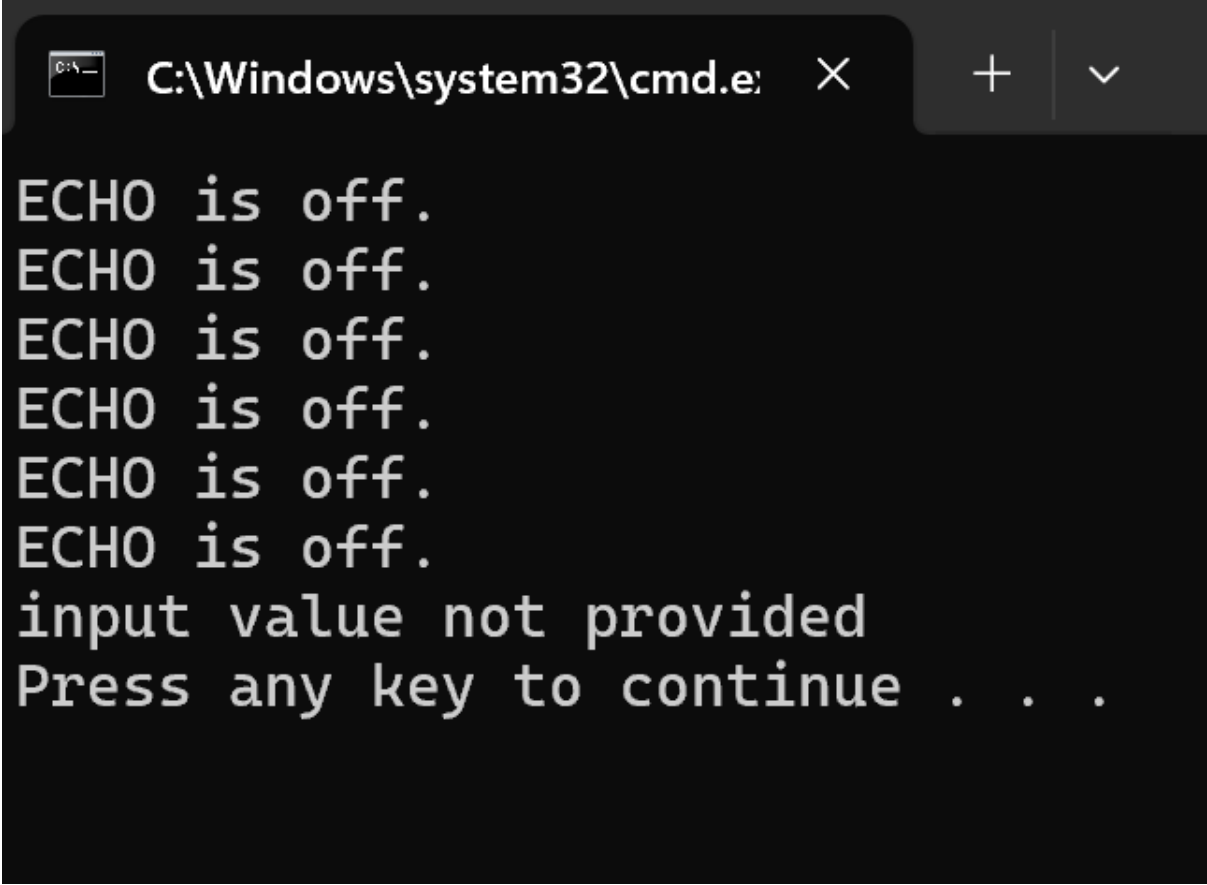
USE OF WINDOWS OS SCRIPTS

Batch Scripting

Batch scripting in Windows is a powerful tool that allows you to automate tasks and run a series of commands in a script file. Here's an example of a batch script that demonstrates arrays and decision-making:

```
1  @echo off
2  setlocal enabledelayedexpansion
3  set topic[0] = comments
4  set topic[1] = variables
5  set topic[2] = arrays
6  set topic[3] = decision making
7  set topic[4] = time and date
8  set topic[5] = operators
9
10 for /l %%n in (0,1,5) do (
11     echo !topic[%%n]!
12 )
13
14 if [%1] == [] (
15     echo input value not provided
16     goto stop
17 ) else (
18     echo "Valid value"
19 )
20
21 :stop
22 pause
```





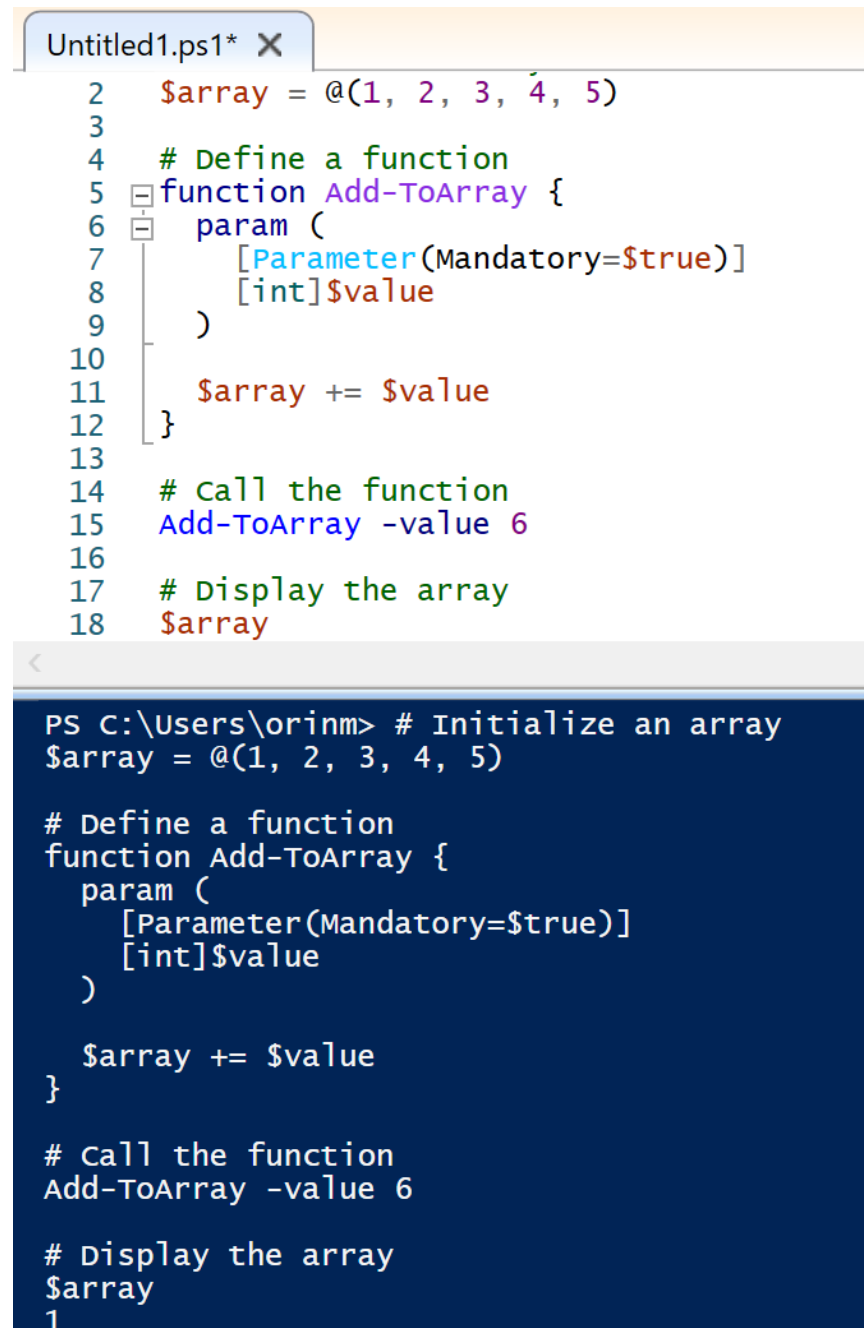
```
C:\Windows\system32\cmd.e: X + v  
ECHO is off.  
ECHO is off.  
ECHO is off.  
ECHO is off.  
ECHO is off.  
ECHO is off.  
input value not provided  
Press any key to continue . . .
```

In this script, we define an array topic with six elements and then iterate through the array to display each element. We also include a decision-making structure using the if statement to check if a value has been provided as input.



PowerShell Scripting

PowerShell scripting in Windows is a more modern and powerful approach to automation. It allows you to work with .NET objects, use advanced functions, and handle errors more effectively. Here's an example of a PowerShell script that demonstrates arrays and functions:



The image shows a PowerShell script in a text editor window titled 'Untitled1.ps1*'. The script defines an array, a function to add values to it, and then calls the function and displays the array. Below the editor is a screenshot of the PowerShell console output, which shows the script being executed line by line.

```
2 $array = @(1, 2, 3, 4, 5)
3
4 # Define a function
5 function Add-ToArray {
6     param (
7         [Parameter(Mandatory=$true)]
8         [int]$value
9     )
10
11     $array += $value
12 }
13
14 # call the function
15 Add-ToArray -value 6
16
17 # Display the array
18 $array
```

```
PS C:\Users\orinm> # Initialize an array
$array = @(1, 2, 3, 4, 5)

# Define a function
function Add-ToArray {
    param (
        [Parameter(Mandatory=$true)]
        [int]$value
    )

    $array += $value
}

# call the function
Add-ToArray -value 6

# Display the array
$array
1
```

In this script, we define an array \$array and a function Add-to Array that takes a mandatory integer parameter \$value and adds it to the array. We then call the function and display the updated array.

Logging

Logging is another essential aspect of scripting. In Windows, you can use the > and >> operators to redirect output to a file. Here's an example of a batch script that demonstrates logging:

```
@echo off
tasklist > process.txt
```

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	5,388 K
Secure System	140	Services	0	40,556 K
Registry	184	Services	0	37,812 K
smss.exe	1048	Services	0	1,132 K
csrss.exe	1484	Services	0	5,056 K
wininit.exe	1576	Services	0	6,160 K
csrss.exe	1600	Console	1	5,988 K
services.exe	1652	Services	0	13,848 K
LsaIso.exe	1664	Services	0	3,352 K
lsass.exe	1688	Services	0	26,976 K
winlogon.exe	1796	Console	1	11,452 K
svchost.exe	1868	Services	0	27,020 K
fontdrvhost.exe	1892	Services	0	2,588 K
fontdrvhost.exe	1944	Console	1	9,556 K
WUDFHost.exe	1952	Services	0	4,728 K
svchost.exe	2036	Services	0	17,220 K
svchost.exe	1232	Services	0	8,096 K
dwm.exe	1568	Console	1	101,740 K
svchost.exe	2108	Services	0	4,904 K
svchost.exe	2120	Services	0	4,648 K
svchost.exe	2212	Services	0	8,804 K
svchost.exe	2220	Services	0	9,352 K
svchost.exe	2312	Services	0	6,576 K
svchost.exe	2328	Services	0	15,512 K
svchost.exe	2348	Services	0	5,512 K
svchost.exe	2400	Services	0	15,516 K
svchost.exe	2508	Services	0	9,096 K
bdservicehost.exe	2520	Services	0	445,836 K
bdservicehost.exe	2532	Services	0	22,012 K
svchost.exe	2616	Services	0	5,712 K
svchost.exe	2688	Services	0	14,300 K
WUDFHost.exe	2752	Services	0	6,800 K
svchost.exe	2780	Services	0	7,164 K
svchost.exe	2944	Services	0	8,940 K
svchost.exe	3080	Services	0	7,644 K
svchost.exe	3204	Services	0	5,676 K
svchost.exe	3272	Services	0	7,140 K

In this script, we use the > operator to redirect the output of the tasklist command to a file named process.txt.

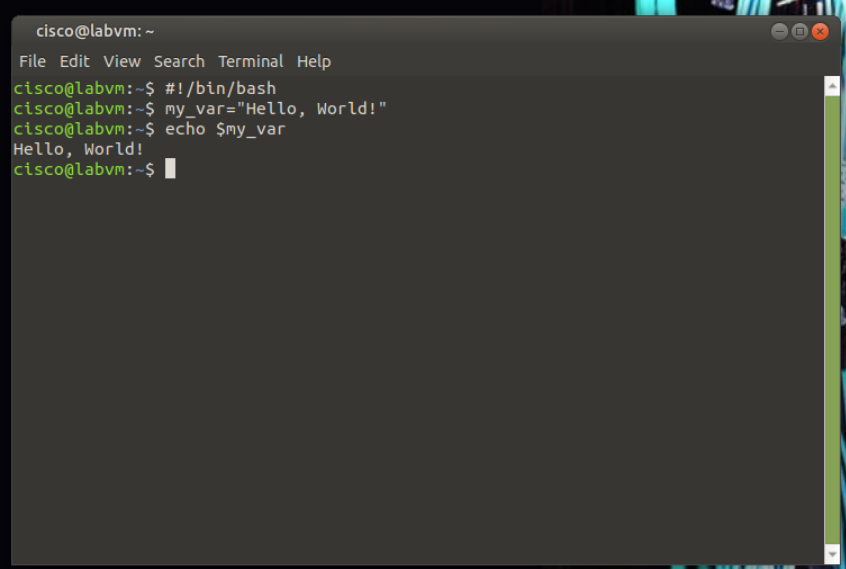


Conclusion

Batch and PowerShell scripting are powerful tools that allow you to automate tasks and work more efficiently in Windows. By understanding key features such as arrays, decision making, functions, error handling, and logging, you can create more effective and reliable scripts.

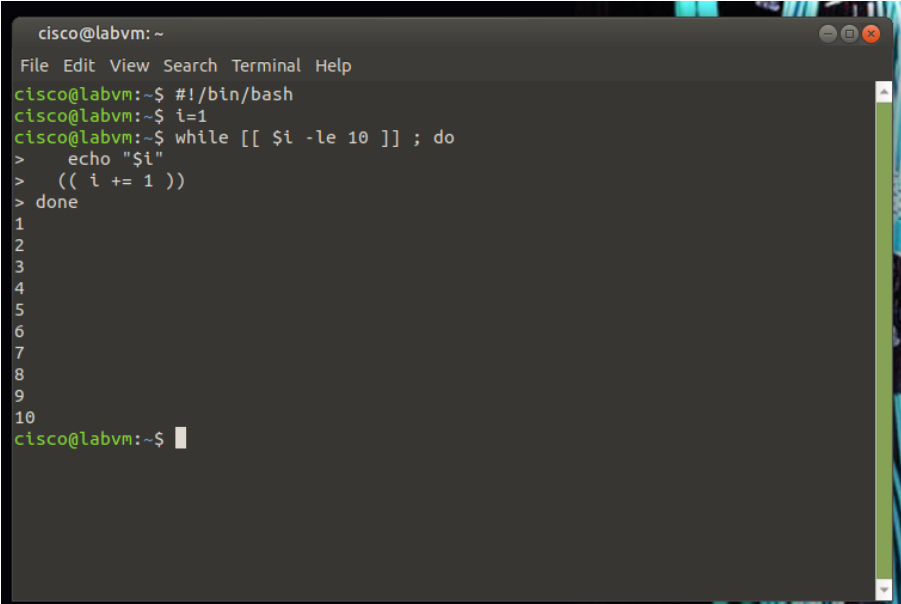
USE OF LINUX OS SCRIPTS

Variables: Variables are used to store data in a script. Here's an example of declaring and using a variable in Bash:



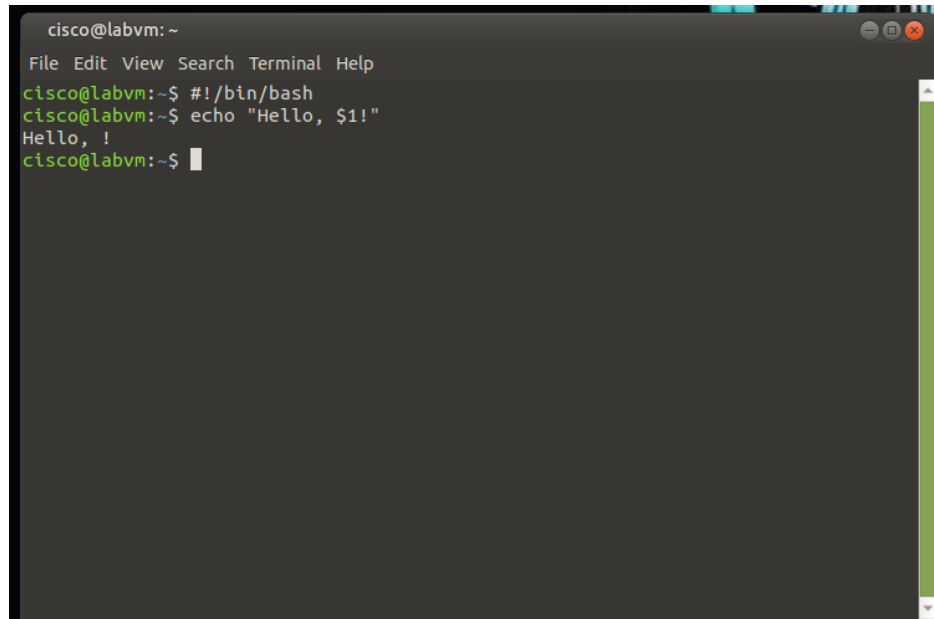
```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ #!/bin/bash  
cisco@labvm:~$ my_var="Hello, World!"  
cisco@labvm:~$ echo $my_var  
Hello, World!  
cisco@labvm:~$
```

Control Structures: Bash provides several control structures, such as if, else, while, until, and for loops. Here's an example of a while loop:



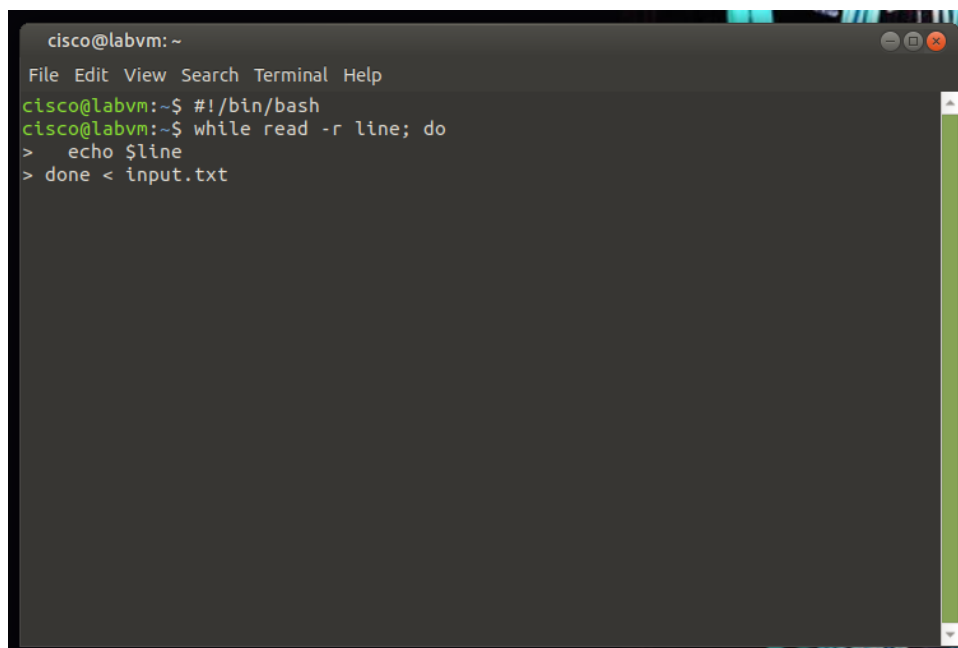
```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ #!/bin/bash  
cisco@labvm:~$ i=1  
cisco@labvm:~$ while [[ $i -le 10 ]] ; do  
>   echo "$i"  
>   (( i += 1 ))  
> done  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
cisco@labvm:~$
```

Command Line Arguments: Bash scripts can accept command line arguments. Here's a script that takes a name as an argument and prints a personalised greeting:

A terminal window titled 'cisco@labvm: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'cisco@labvm:~\$'. The user enters '#!/bin/bash' and 'echo "Hello, \$1!"'. The output is 'Hello, !'. The prompt returns to 'cisco@labvm:~\$' with a cursor.

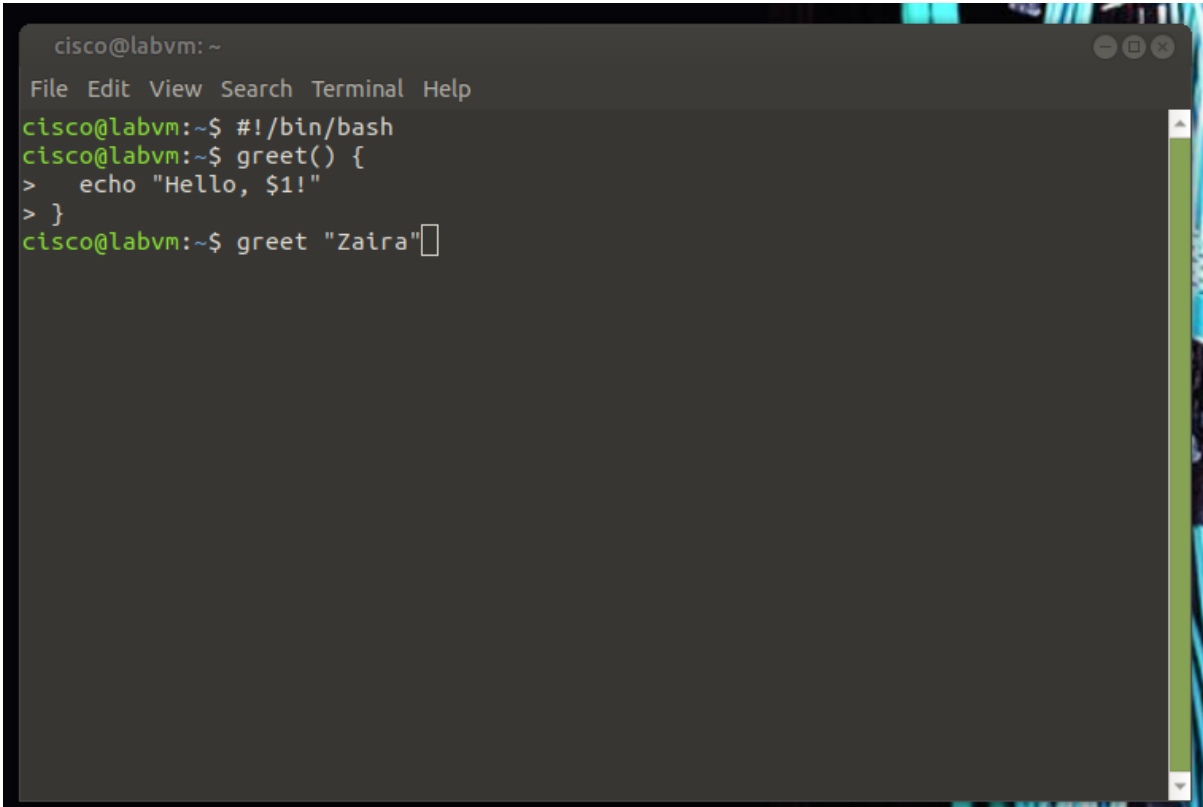
```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ #!/bin/bash  
cisco@labvm:~$ echo "Hello, $1!"  
Hello, !  
cisco@labvm:~$
```

Input/Output Redirection: Bash scripts can read input from and write output to files. Here's an example that reads lines from a file and prints them to the console:

A terminal window titled 'cisco@labvm: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'cisco@labvm:~\$'. The user enters '#!/bin/bash', 'while read -r line; do', 'echo \$line', and 'done < input.txt'.

```
cisco@labvm: ~  
File Edit View Search Terminal Help  
cisco@labvm:~$ #!/bin/bash  
cisco@labvm:~$ while read -r line; do  
> echo $line  
> done < input.txt
```

Functions: Bash scripts can define and use functions. Here's an example:

A screenshot of a terminal window titled 'cisco@labvm: ~'. The terminal has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The prompt is 'cisco@labvm:~\$'. The user enters '#!/bin/bash'. The prompt changes to 'cisco@labvm:~\$'. The user enters 'greet() {'. The prompt changes to '>'. The user enters 'echo "Hello, \$1!"'. The prompt changes to '>'. The user enters '}'. The prompt changes to 'cisco@labvm:~\$'. The user enters 'greet "Zaira"'. The terminal shows the function definition and its execution.

```
cisco@labvm: ~
File Edit View Search Terminal Help
cisco@labvm:~$ #!/bin/bash
cisco@labvm:~$ greet() {
>   echo "Hello, $1!"
> }
cisco@labvm:~$ greet "Zaira"
```



REFERENCE LIST

[1]

0xBEN, “Intro to Object-Oriented Programming with PowerShell,” *Benheater*, Jan. 13, 2022. <https://benheater.com/intro-object-oriented-programming-powershell/> (accessed Apr. 11, 2024).

[2]

stackify, “PowerShell Commands Every Developer Should Know,” *Stackify*, Sep. 21, 2017. <https://stackify.com/powershell-commands-every-developer-should-know/> (accessed Apr. 11, 2024).

[3]

local.host, “Unlocking the Secrets: Is PowerShell Built on the .NET Framework?” *local.host*, May 04, 2023. <https://local.host/is-powershell-built-on-net-framework/> (accessed Apr. 11, 2024).

[4]

poshcode, “Error Handling | PowerShell Practice and Style,” *Poshcode Gitbook.io*, Feb. 24, 2021. <https://poshcode.gitbook.io/powershell-practice-and-style/best-practices/error-handling> (accessed Apr. 11, 2024).

[5]

Gitbook, “Remoting Basics | Secrets of PowerShell Remoting,” *Gitbook.io*, Apr. 15, 2018. <https://devops-collective-inc.gitbook.io/secrets-of-powershell-remoting/remoting-basics> (accessed Apr. 11, 2024).



APPENDICES

LINUX UNHATCHED

I completed the Cisco Networking Academy course, NDG Linux Unhatched. The competition certificate is attached.

Also attached are screenshots of the course from when I was working on it.



Dear Orin McGinn,

Congratulations on completing the NDG Linux Unhatched course in the Cisco Networking Academy. This letter documents you have successfully completed the NDG Linux Unhatched course, which provides an introduction to the Linux command line. Linux is everywhere! As the reach of Linux continues to grow, knowledge of Linux is a core skill for all IT professionals. By completing this course, you have gained a better understanding of Linux.

If you decide to pursue additional knowledge of Linux consider:

NDG LINUX ESSENTIALS

This course is the perfect next step for beginners looking to expand their skills and knowledge of Linux. This full-semester course can be delivered as instructor-led training or as a selfpaced learning experience. The NDG Linux Essentials

*Sincerely,
The NDG Team*

course is designed to prepare you for the Linux Professional Institute Linux Essentials Professional Development Certificate.

Again, congratulations and we wish you continued success!

NDG LINUX SERIES

This certification level series offers beginners more rigorous indepth coverage. The NDG Introduction to Linux I and NDG Introduction to Linux II courses focus on the basic Linux system administration skills needed in preparation for the Linux Professional Institute LPIC-1 certification.

Date 5 Mar 2024



NDG Linux Unhatched - NDG Linux Unhatched

Share

Content

Objectives

11.1

12

13

14

14.1

14.2

15

16

17

18

19

20

21

22

22.1

essential to diagnosing and fixing most Linux based systems. Since Linux has now become so ubiquitous, even those who plan on working primarily with systems not utilizing the Linux kernel can benefit from having a basic understanding of the terminal.

What is a command? A command is a software program that when executed on the CLI (command line interface), performs an action on the computer. When you type in a command, a process is run by the operating system that can read input, manipulate data and produce output. A command runs a process on the operating system, which then causes the computer to perform a job.

To execute a command, the first step is to type the name of the command. Click in the terminal on the right. Type **ls** (lowercase letters **L** and **S**) and hit **Enter**. The result should resemble the example below:

```
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
```

The name of the command is often based on what it does or what the developer who created the command thinks will best describe the command's function. For example, the **ls** command displays a *listing* of information about files. Associating the name of the command with something mnemonic for what it does may help you to remember

>_ Ubuntu PC

```
sysadmin@localhost:~$ ls ~/documents
ls: cannot access '/home/sysadmin/documents': No such file or directory
sysadmin@localhost:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
sysadmin@localhost:~$
```

Toggle onscreen keyboard

Content

Objectives

drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Desktop

←

Ultimately, commands can use many combinations of options and arguments. The possibilities for each command will be unique. Remember the **aptitude** easter egg?

```
sysadmin@localhost:~$ aptitude moo
There are no Easter Eggs in this program.
```

It is possible to alter the behavior of this command using options. See what happens when the **-v** (verbose) option is added:

```
sysadmin@localhost:~$ aptitude -v moo
There really are no Easter Eggs in this program.
```

By combining multiple **-v** options, we can get a variety of responses:

```
sysadmin@localhost:~$ aptitude -vv moo
Didn't I already tell you that there are no Easter Eggs in
sysadmin@localhost:~$ aptitude -vvv moo
Stop it!
```

>_ Ubuntu PC

```
sysadmin@localhost:~$ ls -l
total 4
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Desktop
drwx----- 4 sysadmin sysadmin 4096 Dec 20 2017 Documents
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Downloads
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Music
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Pictures
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Public
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Templates
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Videos
sysadmin@localhost:~$ ls -r
Videos Templates Public Pictures Music Downloads Documents Desktop
sysadmin@localhost:~$ ls -rl
total 4
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Videos
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Templates
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Public
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Pictures
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Music
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Downloads
drwx----- 4 sysadmin sysadmin 4096 Dec 20 2017 Documents
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Desktop
sysadmin@localhost:~$ aptitude moo
There are no Easter Eggs in this program.
```

Toggle onscreen keyboard

Content

Objectives

listing. The result of using both options is a long listing given in reverse order:

```
sysadmin@localhost:~$ ls -l -r
total 32
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Videos
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Templates
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Public
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Pictures
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Music
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Downloads
drwx----- 4 sysadmin sysadmin 4096 Dec 20 2017 Documents
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Desktop
sysadmin@localhost:~$ ls -rl
total 32
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Videos
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Templates
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Public
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Pictures
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Music
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Downloads
drwx----- 4 sysadmin sysadmin 4096 Dec 20 2017 Documents
drwx----- 2 sysadmin sysadmin 4096 Dec 20 2017 Desktop
```

>_ Ubuntu PC

```
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Music
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Pictures
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Public
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Templates
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Videos
sysadmin@localhost:~$ ls -r
Videos Templates Public Pictures Music Downloads Documents Desktop
sysadmin@localhost:~$ ls -rl
total 4
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Videos
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Templates
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Public
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Pictures
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Music
drwx----- 4 sysadmin sysadmin 4096 Dec 20 2017 Downloads
drwx----- 2 sysadmin sysadmin 6 Dec 20 2017 Desktop
sysadmin@localhost:~$ aptitude moo
There are no Easter Eggs in this program.
sysadmin@localhost:~$ aptitude -v moo
There really are no Easter Eggs in this program.
sysadmin@localhost:~$ aptitude -vv moo
Didn't I already tell you that there are no Easter Eggs in this program?
sysadmin@localhost:~$ aptitude -vvv moo
```

Toggle onscreen keyboard



Content

Objectives

directory.

```

sysadmin@localhost:~/Documents/School/Art$ cd ..
sysadmin@localhost:~/Documents/School$

```

The . Character

Regardless of which directory you are in, the . character always represents your current directory. For the cd shortcut is not very useful, but it will come in handy for commands covered in subsequent sections.

The ~ Character

The home directory of the current user is represented by the ~ character. As stated above, you always begin as the sysadmin user, whose home is located at /home/sysadmin. To return to your home directory at any time execute the following command:

```

sysadmin@localhost:~/Documents/School$ cd ~
sysadmin@localhost:~$

```

> Ubuntu PC

```

sysadmin@localhost:~$ pwd
/home/sysadmin
sysadmin@localhost:~$ cd documents
-bash: cd: documents: No such file or directory
sysadmin@localhost:~$ cd /
sysadmin@localhost:/$ cd /home/sysadmin
sysadmin@localhost:~$ pwd
/home/sysadmin
sysadmin@localhost:~$ cd Documents
sysadmin@localhost:~/Documents$ cd school/arts
-bash: cd: school/arts: No such file or directory
sysadmin@localhost:~/Documents$ cd School/Art
sysadmin@localhost:~/Documents/School/Art$ pwd
/home/sysadmin/Documents/School/Art
sysadmin@localhost:~/Documents/School/Art$ cd ..
-bash: cd:.. command not found
sysadmin@localhost:~/Documents/School/Art$ cd ..
sysadmin@localhost:~/Documents/School$ cd ~
sysadmin@localhost:~$

```

NDG Linux Unhatched - NDG Linux Unhatched

Content

Objectives

The numbers in file size field switch from descending to ascending.

Used alone the `-r` option will list the files in reverse alphabetical order:

```

sysadmin@localhost:~$ ls -r /var/log
wtmp      lastlog  faillog  cron.log  auth.log  alte
upstart   kern.log dpkg.log  btmp      apt
syslog    fsck     dmesg    bootstrap.log  apache2

```

Did you know?

You can find Linux in:

< Previous

Next >

>_ Ubuntu PC

```

dnxcr-xx-x 2 root root 6 Apr 11 2014 upstart
sysadmin@localhost:~$ ls -l -5 /var/log
total 824
-rw-r--r-- 1 root root 325238 Dec 20 2017 dpkg.log
-rw-r--r-- 1 root utmp 292584 Mar 5 10:35 lastlog
-rw-r--r-- 1 root adm 85083 Dec 20 2017 dmesg
-rw-r--r-- 1 root root 47816 Dec 7 2017 bootstrap.log
-rw-r--r-- 1 root root 32064 Dec 20 2017 faillog
-rw-r--r-- 1 root root 18047 Dec 20 2017 alternatives.log
-rw-r--r-- 1 syslog adm 13624 Mar 5 11:17 syslog
-rw-r--r-- 1 syslog adm 703 Mar 5 11:17 auth.log
-rw-r--r-- 1 root utmp 384 Mar 5 10:35 wtmp
-rw-r--r-- 1 syslog adm 334 Mar 5 11:17 cron.log
-rw-r--r-- 1 syslog adm 108 Mar 5 10:35 kern.log
dnxcr-xx-x 2 root adm 72 Dec 20 2017 apache2
dnxcr-xx-x 1 root root 41 Dec 20 2017 apt
dnxcr-xx-x 2 root root 38 Dec 7 2017 fsck
dnxcr-xx-x 2 root root 6 Apr 11 2014 upstart
-rw-r--r-- 1 root utmp 0 Dec 7 2017 btmp
sysadmin@localhost:~$ ls -r /var/log
wtmp      lastlog  faillog  cron.log  auth.log  alternatives.log
upstart   kern.log dpkg.log  btmp      apt
syslog    fsck     dmesg    bootstrap.log  apache2
sysadmin@localhost:~$

```

Share ↗

NDG Linux Unhatched - NDG Linux Unhatched

Content	Objectives
 Once the command has completed, notice the prompt has <i>not</i> changed, you are still logged in as <code>syzadmin</code>. The <code>sudo</code> command only provides administrative access for the execution of the specified command. This is an advantage as it reduces the risk that a user accidentally executes a command as root. The intention to execute a command is clear; the command is executed as root if prefixed with the <code>sudo</code> command. Otherwise, the command is executed as a regular user.	> Ubuntu PC ▾ <pre>root@localhost:~# exit logout syzadmin@localhost:~\$ sudo usage: sudo -h -K -k -V usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user] usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command] usage: sudo [-AbEfKnPss] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-u user] [VAR=value] [-i -s] [<i>command</i>] usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-u user] file ... syzadmin@localhost:~\$ sudo sh [sudo] password for syzadmin: syzadmin@localhost:~\$ █</pre>

[< Previous](#)
[Next >](#)

Toggle onscreen keyboard

NDG Linux Unhatched - NDG Linux Unhatched

Content

Objectives

execute (x)

Allows for a file to be run as a process, although script files require read permission, as well.

Allows a user to change to the directory if parent directories have execute permission as well.

Consider This

Understanding which permissions apply is an important skill set in Linux. For example, consider the following set of permissions:

```
-r--rw-rwx. 1 sysadmin staff 999 Apr 10 2013 /home/s
```

In this scenario, the user `sysadmin` ends up having less access to this file than members of the `staff` group or everyone else. The user `sysadmin` only has the permissions of `r--`. It doesn't matter if `sysadmin` is a member of the `staff` group; once user ownership has been established, only the user owner's permissions apply.

Previous Next

Ubuntu PC

```
sysadmin@localhost:~$ cd ~/Documents
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rw-r--r-- 1 sysadmin sysadmin 647 Dec 28 2017 hello.sh
sysadmin@localhost:~/Documents$
```

Toggle onscreen keyboard

NDG Linux Unhatched - NDG Linux Unhatched

Content

Objectives

./hello.sh

```
sysadmin@localhost:~/Documents$ ./hello.sh
( Hello World! )
-----
      \
      <(*)
      ( )
```

Consider This

Notice that to execute the script in the previous example, a `./` character combination was placed before the script name.

`./hello.sh`

This indicates the "command" should be run from the current directory.

Previous Next

Ubuntu PC

```
sysadmin@localhost:~$ cd ~/Documents
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rw-r--r-- 1 sysadmin sysadmin 647 Dec 28 2017 hello.sh
sysadmin@localhost:~/Documents$ ./hello.sh
-bash: ./hello.sh: Permission denied
sysadmin@localhost:~/Documents$ chmod u+x hello.sh
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rwxr--r-- 1 sysadmin sysadmin 647 Dec 28 2017 hello.sh
sysadmin@localhost:~/Documents$ ./hello.sh
-bash: ./hello.sh: No such file or directory
sysadmin@localhost:~/Documents$ ./hello.sh
( Hello World! )
-----
      \
      <(*)
      ( )
sysadmin@localhost:~/Documents$
```

Toggle onscreen keyboard

NDG Linux Unhatched - NDG Linux Unhatched

Content

Objectives

```
sysadmin@localhost:~/Documents$ ./hello.sh
-bash: ./hello.sh: Permission denied
```

Only the user owner has the execute permission, and now the `root` user is the user owner. This file now requires administrative access to execute. Use the `sudo` command to execute the script as the `root` user.

```
sysadmin@localhost:~/Documents$ sudo ./hello.sh
[sudo] password for sysadmin:
( Hello World! )
-----
      \
      <(*)
      ( )
```

Previous Next

Ubuntu PC

```
-rw-r--r-- 1 sysadmin sysadmin 18 Dec 28 2017 numbers.txt
-rw-r--r-- 1 sysadmin sysadmin 77 Dec 28 2017 os.csv
-rw-r--r-- 1 sysadmin sysadmin 59 Dec 28 2017 people.csv
-rw-r--r-- 1 sysadmin sysadmin 118 Dec 28 2017 profile.txt
-rw-r--r-- 1 sysadmin sysadmin 51 Dec 28 2017 red.txt
sysadmin@localhost:~/Documents$ sudo chown root hello.sh
sysadmin@localhost:~/Documents$ sudo chown root hello.sh
-bash: root: command not found
sysadmin@localhost:~/Documents$ sudo root hello.sh
sudo: root: command not found
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rwxr--r-- 1 root sysadmin 647 Dec 28 2017 hello.sh
sysadmin@localhost:~/Documents$ ./hello.sh
-bash: ./hello.sh: Permission denied
sysadmin@localhost:~/Documents$ sudo ./hello.sh
( Hello World! )
-----
      \
      <(*)
      ( )
sysadmin@localhost:~/Documents$
```

Toggle onscreen keyboard



NDG Linux Unhatched - NDG Linux Unhatched

Content Objectives

For example, to change the output of the `head` command to view the first five lines of the `alpha.txt` file:

```
sysadmin@localhost:~/Documents$ head -n 5 alpha.txt
A is for Apple
B is for Bear
C is for Cat
D is for Dog
E is for Elephant
```

View the last five lines of the `alpha.txt` file:

```
sysadmin@localhost:~/Documents$ tail -n 5 alpha.txt
V is for Velvet
W is for Walrus
X is for Xenon
Y is for Yellow
Z is for Zebra
sysadmin@localhost:~/Documents$
```

Previous Next

Toggle onscreen keyboard

Ubuntu PC

```
R is for Rat
S is for Sloth
T is for Turnip
U is for Up
V is for Velvet
W is for Walrus
X is for Xenon
Y is for Yellow
Z is for Zebra
sysadmin@localhost:~/Documents$ head -n 5 alpha.txt
A is for Apple
B is for Bear
C is for Cat
D is for Dog
E is for Elephant
sysadmin@localhost:~/Documents$ tail -n alpha.txt
tail: alpha.txt: invalid number of lines
sysadmin@localhost:~/Documents$ tail -n 5 alpha.txt
V is for Velvet
W is for Walrus
X is for Xenon
Y is for Yellow
Z is for Zebra
sysadmin@localhost:~/Documents$
```

NDG Linux Unhatched - NDG Linux Unhatched

Content Objectives

```
sysadmin@localhost:~/Documents$ ls
School      alpha-third.txt  hidden.txt      numbers.txt
Work        alpha.txt        letters.txt     os.csv
adjectives.txt  animals.txt      linux.txt       password
alpha-first.txt  food.txt        longfile.txt    people.csv
alpha-second.txt  hello.sh        newhome.txt     profile.txt
```

Consider This

Permissions can have an impact on file management commands, such as the `cp` command. In order to copy a file, it is necessary to have execute permission to access the directory where the file is located and the read permission for the file being copied.

It is also necessary to have write and execute permission on the directory the file is being copied to. Typically, there are two places where you should always have write and execute permission on the directory: your home directory and the `/tmp` directory.

Previous Next

Toggle onscreen keyboard

Ubuntu PC

```
sysadmin@localhost:~/Documents$ cd ~/Documents
sysadmin@localhost:~/Documents$ cp /etc/passwd
cp: missing destination file operand after '/etc/passwd'
Try 'cp --help' for more information.
sysadmin@localhost:~/Documents$ cp /etc/passwd .
cp: missing destination file operand after '/etc/passwd.'
Try 'cp --help' for more information.
sysadmin@localhost:~/Documents$ cp /etc/passwd .
cp: cannot stat '/etc/passwd': No such file or directory
sysadmin@localhost:~/Documents$ ls
School      alpha-second.txt  food.txt        linux.txt      os.csv
Work        alpha-third.txt   hello.sh        longfile.txt   people.csv
adjectives.txt  alpha.txt        hidden.txt      newhome.txt    profile.txt
alpha-first.txt  animals.txt      letters.txt     numbers.txt    red.txt
sysadmin@localhost:~/Documents$
```

NDG Linux Unhatched - NDG Linux Unhatched

Content Objectives

The example uses a block size of one megabyte.

count: Count: The number of blocks to be read from the input file.

```
dd if=/dev/zero of=/tmp/swapek bs=1M count=50
```

The example command reads 50 blocks.

Consider This

No block size or count needs to be specified when copying over entire devices. For example, to clone from one hard drive (`/dev/sda`) to another (`/dev/sdb`) execute the following command:

```
dd if=/dev/sda of=/dev/sdb
```

Previous Next

Toggle onscreen keyboard

Ubuntu PC

```
sysadmin@localhost:~/Documents$ cd ~
sysadmin@localhost:~$ dd if=/dev/zero of=/tmp/swapek bs=1M count=50
dd: invalid number '1M'
sysadmin@localhost:~$ dd if=/dev/zero of=/tmp/swapek bs=1M count=50

58+0 records in
58+0 records out
52428800 bytes (52 MB) copied, 0.942432 s, 55.6 MB/s
sysadmin@localhost:~$
```

