



ACCA4022

Computer Architecture and Operating Systems

Operating Systems Assignment

Student Name:AKRAM AHMAD FAKHREDIN

Student ID:2511138

Table of Contents

AI Acknowledgement	4
Introduction.....	5
Task 1 – Key Functions of Operating Systems.....	6
Task 2 – Windows CLI Commands.....	9
Task 3 – Linux CLI Commands.....	16
Task 4 – Windows Batch Scripts	25
Task 5 – Linux Bash Scripts	26
Conclusion	29
References.....	30
Appendix.....	31

List of Figures

Figure 1: Windows Task Manager.....	6
Figure 2: Disk Management.....	7
Figure 3: Control Panel	7
Figure 4: tasklist.....	8
Figure 5: taskkill /PID 160.....	9
Figure 6: dir.....	10
Figure 7: mkdir test.....	11
Figure 8: copy file.txt folder	12
Figure 9: ping google.com	12
Figure 10:ipconfig.....	13
Figure 11: netstat.....	13
Figure 12: net user	14
Figure 13: net user testuser 123456 /add	14
Figure 14: schtasks.....	15
Figure 15: ps	16

Figure 16: top.....	17
Figure 17: kill 3893.....	17
Figure 18:lsmkdir.....	18
Figure 19:rm file.txt.....	19
Figure 20: ping google.com	20
Figure 21: ifconfig	21
Figure 22: netstatclear.....	22
Figure 23: passwd username.....	23
Figure 24: crontab -l.....	23
Figure 25: Script Code	25
Figure 26: Script Output	25
Figure 27: Greeting Script	26
Figure 28: Script Output	26
Figure 29: Script	27
Figure 30: Output Screen	27

AI Acknowledgement

The assignment utilized the Google AI to assist in the research process and increase grammar and coherence. Data generated by the AI tool was verified and reformulated in my own words, with referring to other academic resources to ensure the precision and maintain the academic quality.

Introduction

System software that manages hardware and software of a computer is required which is an Operating System (OS). It is also used as a source of communication between the user and the computer hardware through which the user can easily execute the application and perform other computing functions. Both Microsoft windows and Linux are popular operating systems with both the command-line interface and graphical user interface system administration. OSs are also important as they control the memory, processes, files and devices such that the computer operates comfortably. This report discusses some of the greatest operating system functionalities and it explains how to use Windows and Linux command line interfaces. It also talks of the application of scripting to automate handling the system administration.

Task 1 – Key Functions of Operating Systems

The operating systems have turned out to be highly essential in handling the computer resources and also in the due manner of interaction between the hardware and the software components [1]. With the assistance of a few significant functions, an operating system can perform a number of these functions.

Process Management

Process management involves control of running of computer programs. The operating system is also committed to process usage and enables allocation of more than one program to be run simultaneously without distraction to the rest. This is the so-called multitasking. The operating system decides the timing of the tasks to be accomplished depending on the priority and the need of that system. There is also a Task Manager of windows in which users can monitor the on-going processes of computer as well as shows the CPU usage, memory usage and running processes.

Memory Management

Memory management is the process of inside allocation and administration of primary memory of a computer (RAM). The operating system determines the allocated memory size which the individual process is being given, and which the programs are not sharing memory with the rest of the processes. The modern operating systems also use virtual memory and the memory is merely increased by storing the data at least on the hard disk in an interim format. This helps the systems in the implementation of large scale applications despite a limited physical memory situation.

File System Management

The operating system organizes the data within files and directories to help users to access and store data easily. File system management helps users to delete, create, move, and modify files and control access authority. There is a File Explorer interface in the Windows systems and it allows exploration of the folders and with ease to manipulate files.

Device Management

Operating systems deal with hardware devices such as keyboards, printers, storage devices and monitors [2]. This is achieved through the use of device drivers that take the role of an interpreter between the operating system and the hardware devices. To give an example, the

operating system will interact with the printer driver when a user is printing a document to carry out the act.

Security and User Management

The other crucial functions of operating system is security. It restricts the access to resources in the system through authentication systems such as usernames and passwords. Administrators are allowed to control the access of files, applications and systems through user accounts. The permission of access control assists in ensuring that one can only perform a specific action which is authorized by the user.

Networking

The operating systems also have the ability to perform networking in which computers are linked either to the internet or other computers on a network [3]. The capabilities of networking include IP configuration, tracking network and communication protocols. These functions are applicable in accessing online resources, in exchanging files besides in collaboration with other systems.

Name	Status	CPU	Memory	Disk	Network
Apps (1)					
Task Manager		1.0%	6.1 MB	0.1 MB/s	0 Mbps
Background processes (8)					
Google Crash Handler		0%	0.3 MB	0 MB/s	0 Mbps
Google Crash Handler (32 bit)		0%	0.3 MB	0 MB/s	0 Mbps
Google Installer (32 bit)		0%	0.5 MB	0 MB/s	0 Mbps
Host Process for Windows Tasks		0%	1.4 MB	0 MB/s	0 Mbps
Microsoft Windows Search Filte...		0%	0.8 MB	0 MB/s	0 Mbps
Microsoft Windows Search Inde...		0%	5.8 MB	0 MB/s	0 Mbps
Microsoft Windows Search Prot...		0%	1.2 MB	0 MB/s	0 Mbps
Spooler SubSystem App		0%	0.5 MB	0 MB/s	0 Mbps
Windows processes (22)					
Antimalware Service Executable		0%	17.3 MB	0 MB/s	0 Mbps

Figure 1: Windows Task Manager

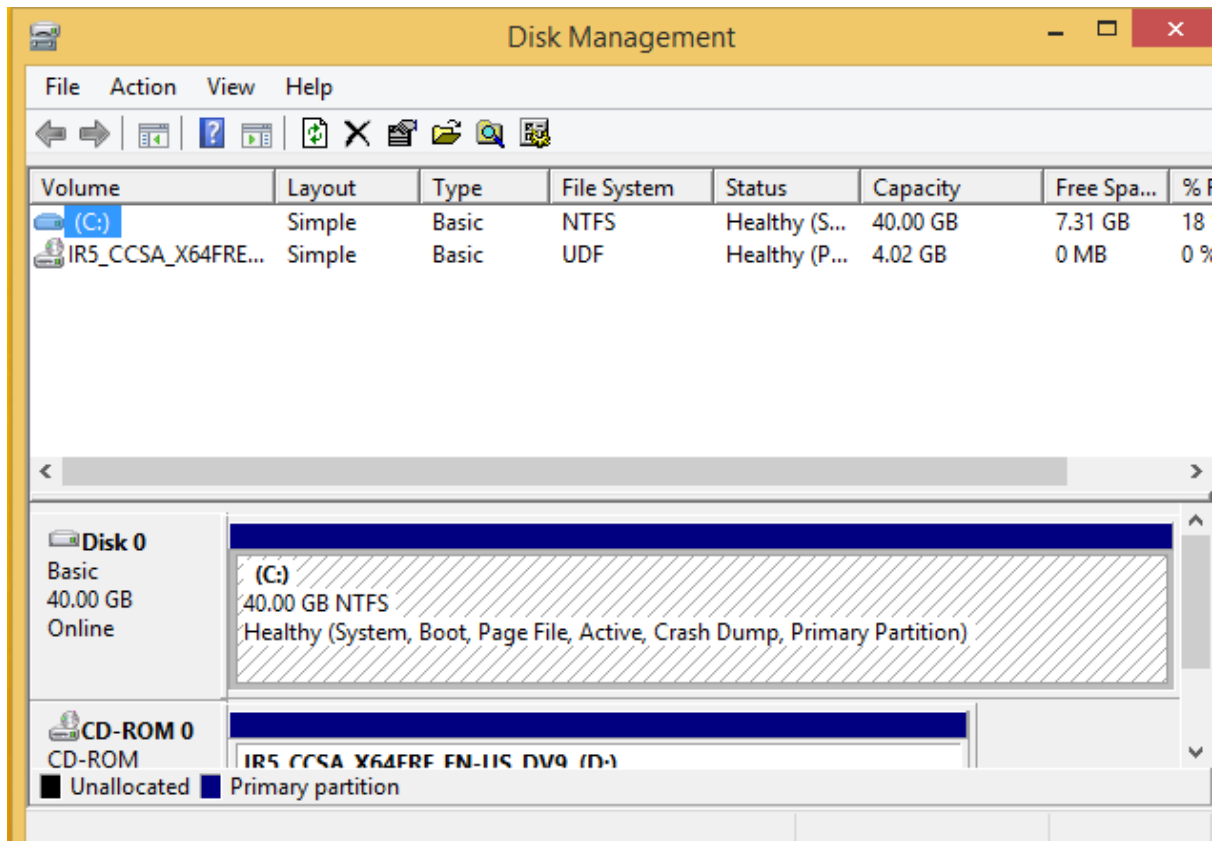


Figure 2: Disk Management

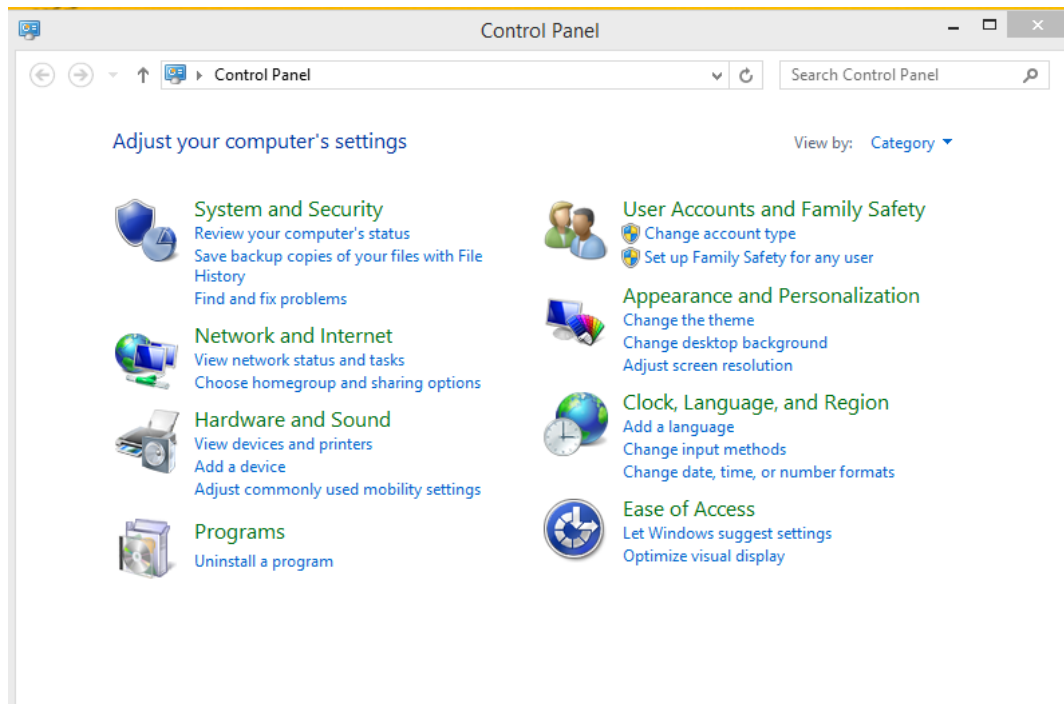


Figure 3: Control Panel

These screenshots demonstrate how operating systems manage system resources and provide tools for monitoring system performance.

Task 2 – Windows CLI Commands

Windows operating systems Command Interfaces Command Prompt and Power Shell are command interfaces that allow a user to interact with a system through the use of textual commands [4]. The tools have the capability to have the user performing many administrative and management tasks devoid of graphical interface. Through command-line commands, it would help users monitor performance of the system, handle files and directories, configure the network perimeter, handle the running processes, and handle user accounts too. PowerShell has greater levels of scripting and automation as compared to Command Prompt because standard batch commands are used [5]. These command-line tools are especially convenient to system administrators and IT workers, as they allow them to troubleshoot quicker, as well as administer the system and script as well as schedule routine jobs.

Process Management Commands

Tasklist

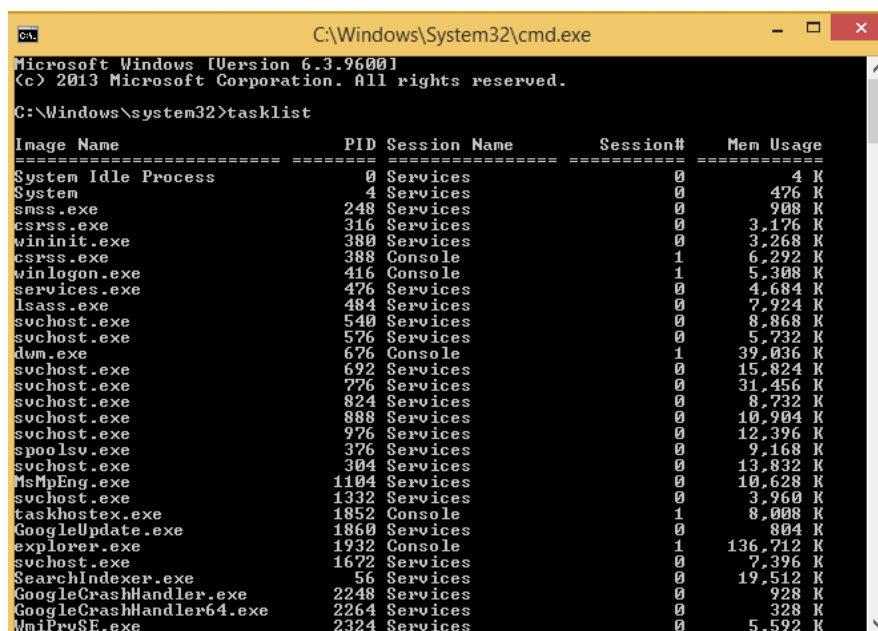
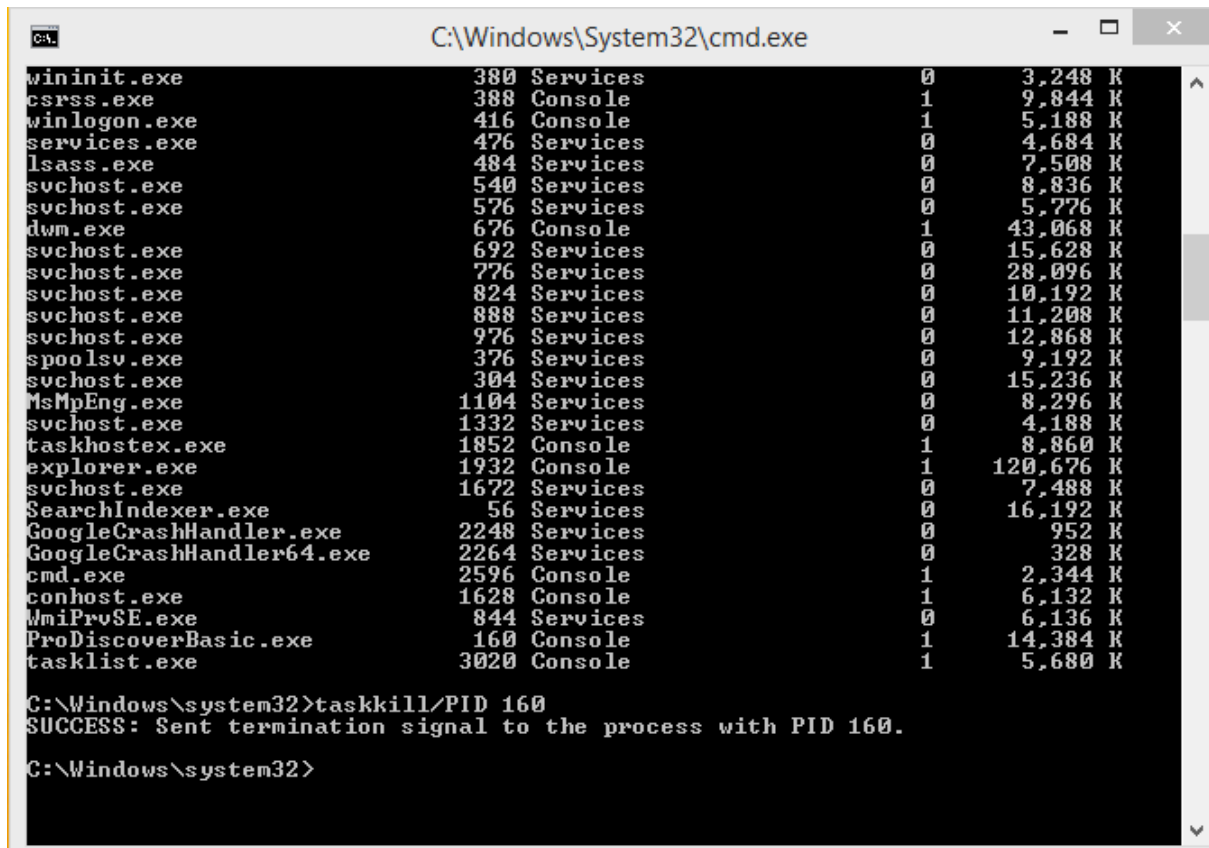


Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	4 K
System	4	Services	0	476 K
smss.exe	248	Services	0	908 K
csrss.exe	316	Services	0	3,176 K
wininit.exe	380	Services	0	3,268 K
csrss.exe	388	Console	1	6,292 K
winlogon.exe	416	Console	1	5,308 K
services.exe	476	Services	0	4,684 K
lsass.exe	484	Services	0	7,924 K
svchost.exe	540	Services	0	8,868 K
svchost.exe	576	Services	0	5,732 K
dwm.exe	676	Console	1	39,036 K
svchost.exe	692	Services	0	15,824 K
svchost.exe	776	Services	0	31,456 K
svchost.exe	824	Services	0	8,732 K
svchost.exe	888	Services	0	10,904 K
svchost.exe	976	Services	0	12,396 K
spoolsv.exe	376	Services	0	9,168 K
svchost.exe	304	Services	0	13,832 K
MsMpEng.exe	1104	Services	0	10,628 K
svchost.exe	1332	Services	0	3,960 K
taskhost.exe	1852	Console	1	8,008 K
GoogleUpdate.exe	1860	Services	0	804 K
explorer.exe	1932	Console	1	136,712 K
svchost.exe	1672	Services	0	7,396 K
SearchIndexer.exe	56	Services	0	19,512 K
GoogleCrashHandler.exe	2248	Services	0	928 K
GoogleCrashHandler64.exe	2264	Services	0	328 K
WmiPrvSE.exe	2324	Services	0	5,592 K

Figure 4: tasklist

The list of all the running processes is presented by this command. It shows information regarding process name, process id(PID) and memory use.

Taskkill



```
C:\Windows\System32\cmd.exe

wininit.exe      380 Services      0      3,248 K
csrss.exe        388 Console       1      9,844 K
winlogon.exe     416 Console       1      5,188 K
services.exe    476 Services      0      4,684 K
lsass.exe        484 Services      0      7,508 K
svchost.exe     540 Services      0      8,836 K
svchost.exe     576 Services      0      5,776 K
dwm.exe          676 Console       1     43,068 K
svchost.exe     692 Services      0     15,628 K
svchost.exe     776 Services      0     28,096 K
svchost.exe     824 Services      0     10,192 K
svchost.exe     888 Services      0     11,208 K
svchost.exe     976 Services      0     12,868 K
spoolsv.exe     376 Services      0      9,192 K
svchost.exe     304 Services      0     15,236 K
MsMpEng.exe     1104 Services     0      8,296 K
svchost.exe     1332 Services     0      4,188 K
taskhostex.exe  1852 Console       1      8,860 K
explorer.exe    1932 Console       1    120,676 K
svchost.exe     1672 Services     0      7,488 K
SearchIndexer.exe  56 Services      0     16,192 K
GoogleCrashHandler.exe 2248 Services     0      952 K
GoogleCrashHandler64.exe 2264 Services     0      328 K
cmd.exe         2596 Console       1      2,344 K
conhost.exe     1628 Console       1      6,132 K
WmiProSE.exe    844 Services      0      6,136 K
ProDiscoverBasic.exe 160 Console       1     14,384 K
tasklist.exe    3020 Console       1      5,680 K

C:\Windows\system32>taskkill /PID 160
SUCCESS: Sent termination signal to the process with PID 160.

C:\Windows\system32>
```

Figure 5: taskkill /PID 160

The command is used to terminate a running process that has Process ID. It is also practical in case one of the applications comes to a halt.

File and Directory Management

Dir

```
C:\Windows\System32\cmd.exe

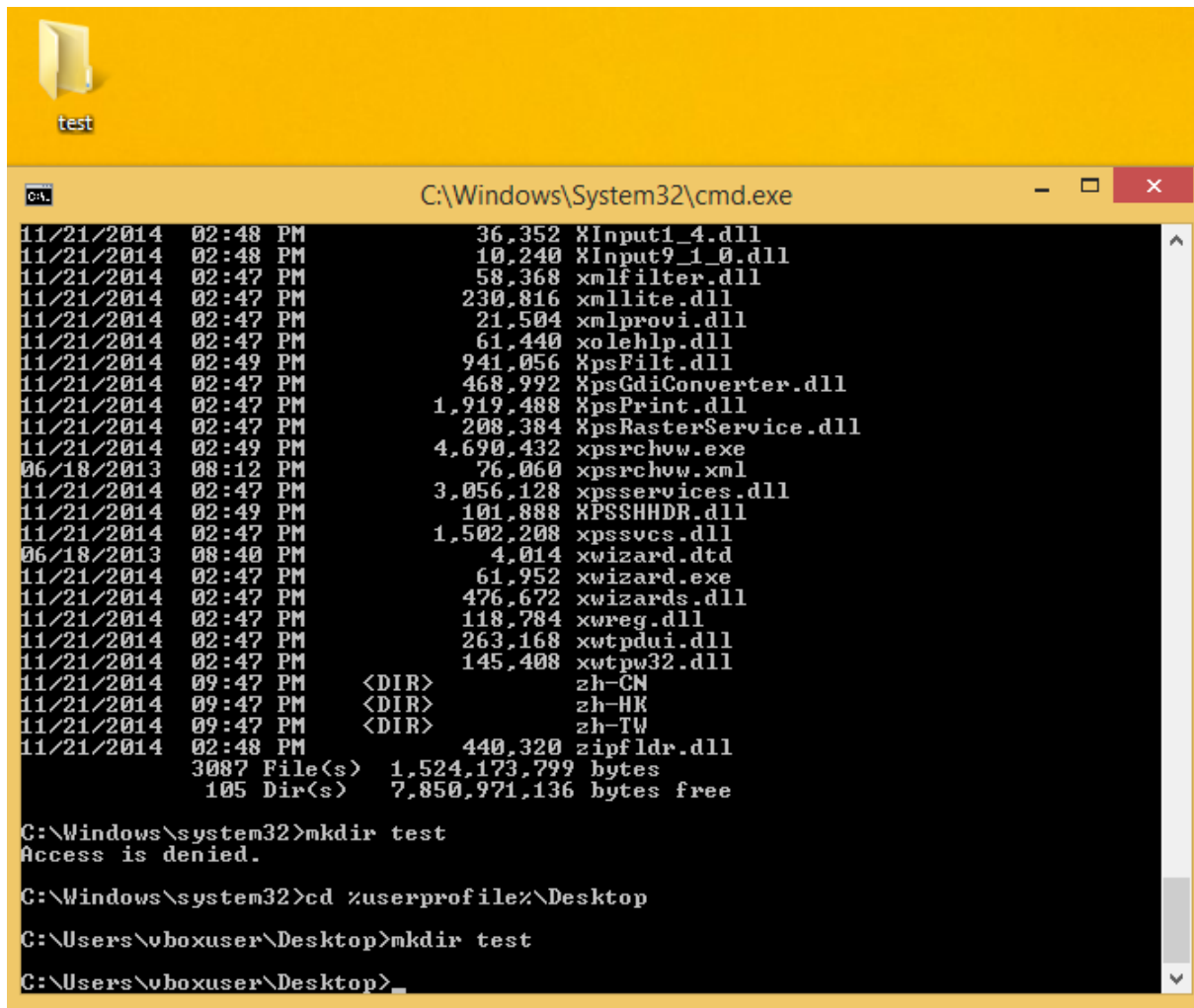
11/21/2014 02:47 PM          50,688 Wwanpref.dll
11/21/2014 02:47 PM          72,704 wwanprot.dll
11/21/2014 02:47 PM          62,976 WwanRadioManager.dll
11/21/2014 02:47 PM         513,536 wwanSvc.dll
11/21/2014 02:47 PM          63,528 wwanapi.dll
11/21/2014 02:47 PM         293,376 XAudio2_8.dll
11/21/2014 02:47 PM          48,640 xcopy.exe
11/21/2014 02:48 PM          36,352 XInput1_4.dll
11/21/2014 02:48 PM          10,240 XInput9_1_0.dll
11/21/2014 02:47 PM          58,368 xmlfilter.dll
11/21/2014 02:47 PM         230,816 xmlite.dll
11/21/2014 02:47 PM          21,504 xmlprovi.dll
11/21/2014 02:47 PM          61,440 xolehlp.dll
11/21/2014 02:49 PM          941,056 XpsFilt.dll
11/21/2014 02:47 PM          468,992 XpsGdiConverter.dll
11/21/2014 02:47 PM         1,919,488 XpsPrint.dll
11/21/2014 02:47 PM         208,384 XpsRasterService.dll
11/21/2014 02:49 PM         4,690,432 xpsrchvw.exe
06/18/2013 08:12 PM           76,060 xpsrchvw.xml
11/21/2014 02:47 PM         3,056,128 xpsServices.dll
11/21/2014 02:49 PM          101,888 XPSSHHDR.dll
11/21/2014 02:47 PM         1,502,208 xpssvcs.dll
06/18/2013 08:40 PM           4,014 xwizard.dtd
11/21/2014 02:47 PM           61,952 xwizard.exe
11/21/2014 02:47 PM          476,672 xwizards.dll
11/21/2014 02:47 PM          118,784 xwreg.dll
11/21/2014 02:47 PM          263,168 xwtpdui.dll
11/21/2014 02:47 PM          145,408 xwtpw32.dll
11/21/2014 09:47 PM          <DIR> zh-CN
11/21/2014 09:47 PM          <DIR> zh-HK
11/21/2014 09:47 PM          <DIR> zh-TW
11/21/2014 02:48 PM         440,320 zipfldr.dll
3087 File(s)  1,524,173,799 bytes
105 Dir(s)   7,850,971,136 bytes free

C:\Windows\system32>
```

Figure 6: dir

The dir will list files and directories of the working directory. It helps one to open the saved files and navigate that is in the directories.

Mkdir



```
C:\Windows\System32\cmd.exe
11/21/2014 02:48 PM          36,352 XInput1_4.dll
11/21/2014 02:48 PM          10,240 XInput9_1_0.dll
11/21/2014 02:47 PM          58,368 xmlfilter.dll
11/21/2014 02:47 PM        230,816 xmlite.dll
11/21/2014 02:47 PM          21,504 xmlprovi.dll
11/21/2014 02:47 PM          61,440 xolehlp.dll
11/21/2014 02:49 PM          941,056 XpsFilt.dll
11/21/2014 02:47 PM          468,992 XpsGdiConverter.dll
11/21/2014 02:47 PM        1,919,488 XpsPrint.dll
11/21/2014 02:47 PM          208,384 XpsRasterService.dll
11/21/2014 02:49 PM          4,690,432 xpsrchow.exe
06/18/2013 08:12 PM           76,060 xpsrchow.xml
11/21/2014 02:47 PM        3,056,128 xpsservices.dll
11/21/2014 02:49 PM          101,888 XPSSHHDR.dll
11/21/2014 02:47 PM        1,502,208 xpssvcs.dll
06/18/2013 08:40 PM           4,014 xwizard.dtd
11/21/2014 02:47 PM           61,952 xwizard.exe
11/21/2014 02:47 PM          476,672 xwizards.dll
11/21/2014 02:47 PM          118,784 xwreg.dll
11/21/2014 02:47 PM          263,168 xwtpdui.dll
11/21/2014 02:47 PM          145,408 xwtpw32.dll
11/21/2014 09:47 PM          <DIR>          zh-CN
11/21/2014 09:47 PM          <DIR>          zh-HK
11/21/2014 09:47 PM          <DIR>          zh-TW
11/21/2014 02:48 PM          440,320 zipfldr.dll
3087 File(s)  1,524,173,799 bytes
105 Dir(s)    7,850,971,136 bytes free

C:\Windows\system32>mkdir test
Access is denied.

C:\Windows\system32>cd %userprofile%\Desktop
C:\Users\vboxuser\Desktop>mkdir test
C:\Users\vboxuser\Desktop>
```

Figure 7: mkdir test

This is an instruction that creates a new folder termed test. It is generally used in file organization and folders.

Copy

```
C:\Windows\System32\cmd.exe
11/21/2014 02:49 PM 4,690,432 xpsrchvw.exe
06/18/2013 08:12 PM 76,060 xpsrchvw.xml
11/21/2014 02:47 PM 3,056,128 xpsservices.dll
11/21/2014 02:49 PM 101,888 XPSSHHDR.dll
11/21/2014 02:47 PM 1,502,208 xpssvcs.dll
06/18/2013 08:40 PM 4,014 xwizard.dtd
11/21/2014 02:47 PM 61,952 xwizard.exe
11/21/2014 02:47 PM 476,672 xwizards.dll
11/21/2014 02:47 PM 118,784 xwreg.dll
11/21/2014 02:47 PM 263,168 xwtpdnt.dll
11/21/2014 02:47 PM 145,408 xwtpw32.dll
11/21/2014 09:47 PM <DIR> zh-CN
11/21/2014 09:47 PM <DIR> zh-HK
11/21/2014 09:47 PM <DIR> zh-TW
11/21/2014 02:48 PM 440,320 zipfldr.dll
3087 File(s) 1,524,173,799 bytes
105 Dir(s) 7,850,971,136 bytes free

C:\Windows\system32>mkdir test
Access is denied.

C:\Windows\system32>cd %userprofile%\Desktop

C:\Users\vboxuser\Desktop>mkdir test

C:\Users\vboxuser\Desktop>copy file.txt folder
The system cannot find the file specified.

C:\Users\vboxuser\Desktop>echo Hello > file.txt

C:\Users\vboxuser\Desktop>mkdir folder

C:\Users\vboxuser\Desktop>copy file.txt folder
1 file(s) copied.

C:\Users\vboxuser\Desktop>
```

Figure 8: copy file.txt folder

Replication of files in the file system is done via the copy command to replicate files at one place to another.

Network Testing and Monitoring

Ping

```
C:\Windows\System32\cmd.exe

C:\Users\vboxuser\Desktop>ping google.com

Pinging google.com [142.250.193.78] with 32 bytes of data:
Reply from 142.250.193.78: bytes=32 time=13ms TTL=119
Reply from 142.250.193.78: bytes=32 time=15ms TTL=119
Reply from 142.250.193.78: bytes=32 time=13ms TTL=119
Reply from 142.250.193.78: bytes=32 time=10ms TTL=119

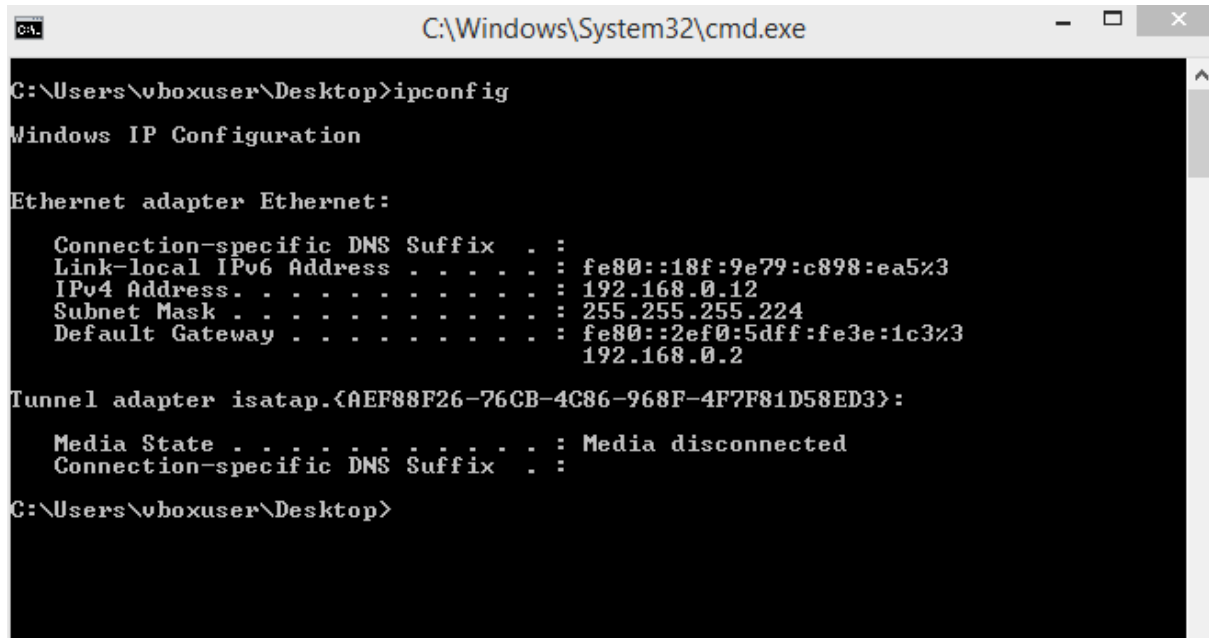
Ping statistics for 142.250.193.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 10ms, Maximum = 15ms, Average = 12ms

C:\Users\vboxuser\Desktop>
```

Figure 9: ping google.com

It is a command that is applied to test the connection of the network by transmitting packets to a remote server and timing the reply.

Ipconfig



```
C:\Windows\System32\cmd.exe

C:\Users\vboxuser\Desktop>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::18f:9e79:c898:ea5%3
    IPv4 Address. . . . . : 192.168.0.12
    Subnet Mask . . . . . : 255.255.255.224
    Default Gateway . . . . . : fe80::2ef0:5dff:fe3e:1c3%3
                                192.168.0.2

Tunnel adapter isatap.{A9F88F26-76CB-4C86-968F-4F7F81D58ED3}:

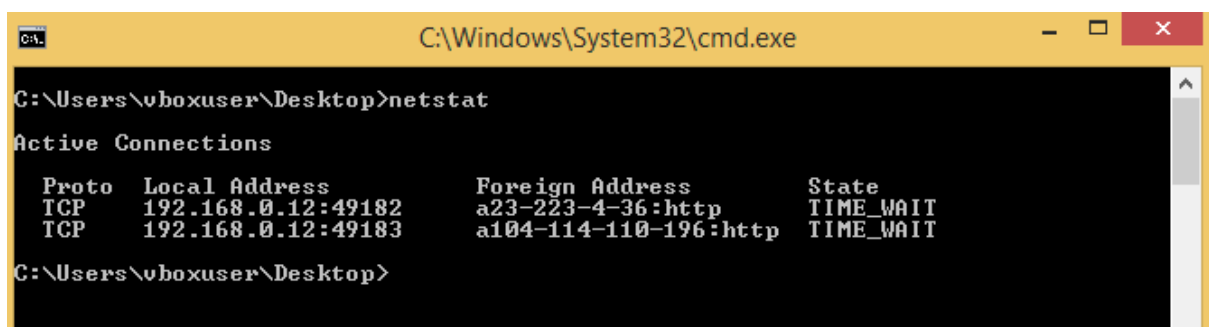
    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

C:\Users\vboxuser\Desktop>
```

Figure 10: ipconfig

This command is intended to test the connection in the network by sending packets to one of the remote servers and measuring the latency.

Netstat



```
C:\Windows\System32\cmd.exe

C:\Users\vboxuser\Desktop>netstat

Active Connections

 Proto Local Address           Foreign Address         State
 TCP    192.168.0.12:49182      a23-223-4-36:http      TIME_WAIT
 TCP    192.168.0.12:49183      a104-114-110-196:http  TIME_WAIT

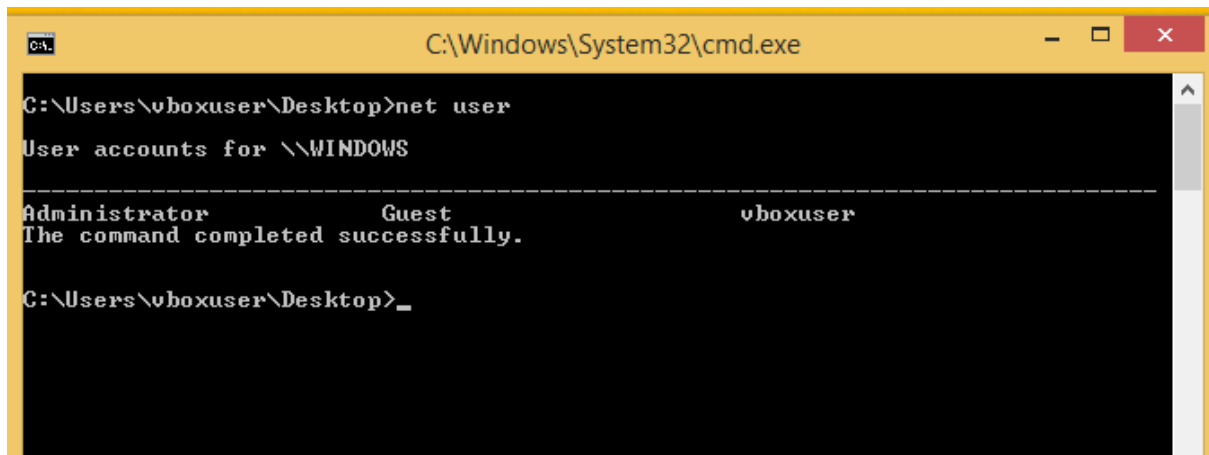
C:\Users\vboxuser\Desktop>
```

Figure 11: netstat

With this command, related networks and ports of the system that are listening will be shown.

User Management

net user



```
C:\Users\vboxuser\Desktop>net user
User accounts for \\WINDOWS

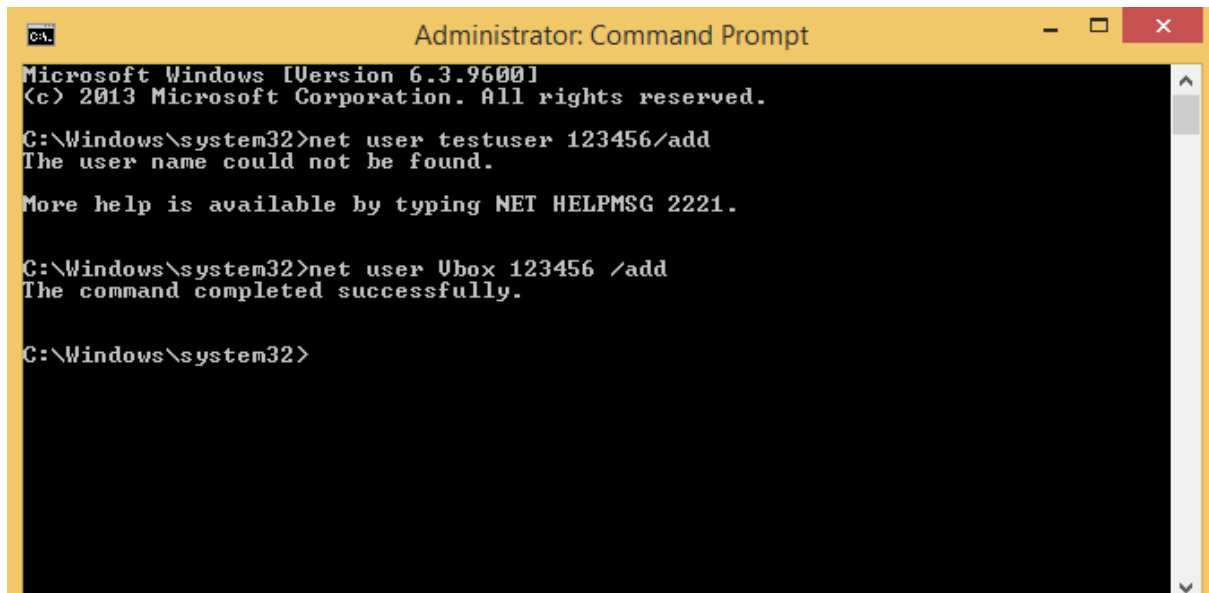
-----
Administrator      Guest      vboxuser
The command completed successfully.

C:\Users\vboxuser\Desktop>
```

Figure 12: net user

Displays the user accounts list available in the system.

net user username password /add



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user testuser 123456/add
The user name could not be found.

More help is available by typing NET HELPMSG 2221.

C:\Windows\system32>net user Ubox 123456 /add
The command completed successfully.

C:\Windows\system32>
```

Figure 13: net user testuser 123456 /add

New account with user name and password are introduced.

Task Scheduling

Schtasks

```

Administrator: Command Prompt

Scheduled Start With Network          3/14/2026 10:36:47 AM Ready
Folder: \Microsoft\Windows\Wininet
TaskName                             Next Run Time      Status
=====
CacheTask                             N/A                 Running

Folder: \Microsoft\Windows\WOF
TaskName                             Next Run Time      Status
=====
WIM-Hash-Management                  N/A                 Ready
WIM-Hash-Validation                  N/A                 Disabled

Folder: \Microsoft\Windows\Work Folders
TaskName                             Next Run Time      Status
=====
Work Folders Logon Synchronization    N/A                 Ready
Work Folders Maintenance Work         N/A                 Ready

Folder: \Microsoft\Windows\Workplace Join
TaskName                             Next Run Time      Status
=====
Automatic-Workplace-Join              N/A                 Disabled

Folder: \Microsoft\Windows\WS
TaskName                             Next Run Time      Status
=====
Badge Update                          N/A                 Ready
License Validation                    3/16/2026 10:38:27 AM Ready
Sync Licenses                        N/A                 Ready
WSRefreshBannedAppsListTask          3/14/2026 12:13:44 PM Ready
WSIask                               N/A                 Ready

C:\Windows\system32>

```

Figure 14: schtasks

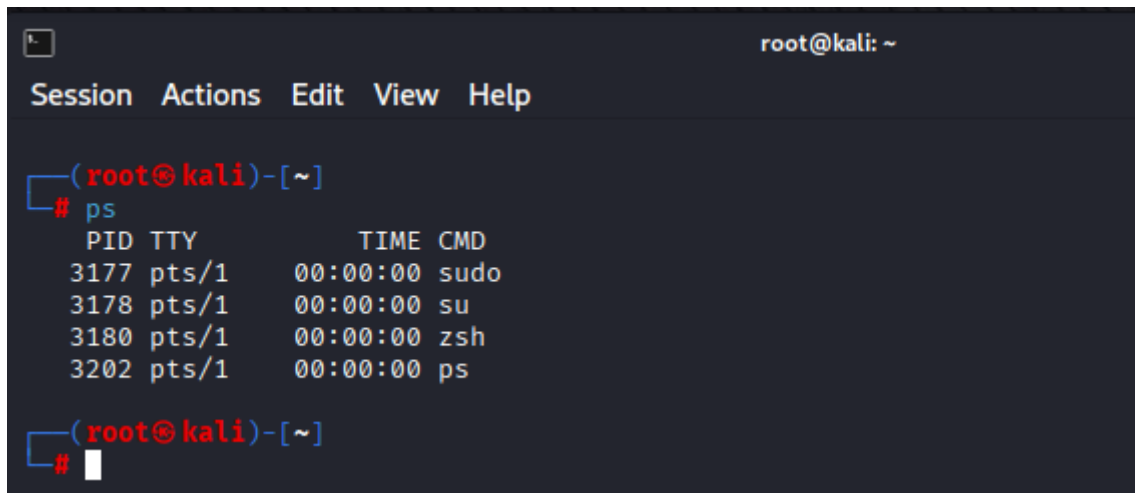
Schtasks command allows the users to schedule or to manage scheduled tasks that are executed automatically at some times.

Task 3 – Linux CLI Commands

Linux operating systems contain strong command-line interface (CLI) applications to enable one to interact with the system through the terminal [6]. These commands can enable the user to control and manipulate files and directory, view and control running processes, set up system setting and user accounts. Linux CLI is often used by system administrators since they can administer the system effectively and troubleshoot system failures. It also gives a preference to the automation with shell scripts that allow performing repetitive tasks and managing the actions of the system within a short time and efficiently.

Process Management

Ps



```
root@kali: ~  
Session Actions Edit View Help  
  
(root@kali)-[~]  
# ps  
  PID TTY          TIME CMD  
 3177 pts/1    00:00:00 sudo  
 3178 pts/1    00:00:00 su  
 3180 pts/1    00:00:00 zsh  
 3202 pts/1    00:00:00 ps  
  
(root@kali)-[~]  
#
```

Figure 15: ps

The ps command provides the information of the processes in the process of running.

Top

```
root@kali: ~
Session Actions Edit View Help

top - 22:45:32 up 48 min, 1 user, load average: 0.21, 0.44, 1.24
Tasks: 193 total, 1 running, 192 sleeping, 0 stopped, 0 zombie
%Cpu(s): 2.2 us, 1.7 sy, 0.0 ni, 96.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3919.3 total, 172.9 free, 1489.1 used, 2697.3 buff/cache
MiB Swap: 4094.0 total, 3828.5 free, 265.5 used, 2430.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 862 root        20   0 395456 116440 59188 S   7.3   2.9   1:28.36 Xorg
3125 ethical    20   0 656588 64048 49132 S   2.0   1.6   0:00.68 qterminal
 795 redis      20   0 74580 13700 7804 S   0.7   0.3   0:29.34 redis-server
1617 ethical    20   0 561384 92172 68540 S   0.7   2.3   0:14.27 xfwm4
 495 message+  20   0 10480 6588 4148 S   0.3   0.2   0:05.04 dbus-daemon
 643 root       20   0 353140 2756 2488 S   0.3   0.1   0:06.23 VBoxDRMClient
 811 root       20   0 1604676 18584 7604 S   0.3   0.5   0:08.67 containerd
 818 root       20   0 2212508 402892 15336 S   0.3  10.0  22:52.43 nessusd
    1 root       20   0 24932 15024 10516 S   0.0   0.4   0:01.72 systemd
    2 root       20   0      0      0      0 S   0.0   0.0   0:00.03 kthreadd
    3 root       20   0      0      0      0 S   0.0   0.0   0:00.00 pool_workqueue_release
    4 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-rcu_gp
    5 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-sync_wq
    6 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-kvfree_rcu_reclaim
    7 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-slub_flushwq
    8 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-netns
   12 root      20   0      0      0      0 I   0.0   0.0   0:00.00 kworker/u8:0-ipv6_addrconf
   13 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-mm_percpu_wq
   14 root      20   0      0      0      0 S   0.0   0.0   0:24.42 ksoftirqd/0
   15 root      20   0      0      0      0 I   0.0   0.0   0:01.42 rcu_preempt
   16 root      20   0      0      0      0 S   0.0   0.0   0:00.00 rcu_exp_par_gp_kthread_worker/0
   17 root      20   0      0      0      0 S   0.0   0.0   0:00.04 rcu_exp_gp_kthread_worker
   18 root      rt   0      0      0      0 S   0.0   0.0   0:00.11 migration/0
   19 root     -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   20 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   21 root      20   0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   22 root     -51   0      0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
   23 root      rt   0      0      0      0 S   0.0   0.0   0:00.82 migration/1
   24 root      20   0      0      0      0 S   0.0   0.0   0:10.98 ksoftirqd/1
   25 root      20   0      0      0      0 I   0.0   0.0   0:00.37 kworker/1:0-mm_percpu_wq
   27 root      20   0      0      0      0 I   0.0   0.0   0:11.51 kworker/u9:0-flush-8:0
   31 root      20   0      0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   32 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-inet_frag_wq
   33 root      20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
   34 root      20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
   35 root      20   0      0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
   36 root      20   0      0      0      0 S   0.0   0.0   0:00.00 kauditd
   37 root      20   0      0      0      0 S   0.0   0.0   0:00.01 khungtaskd
   38 root      20   0      0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   40 root       0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/R-writeback
```

Figure 16: top

The top command provides us real time information regarding processes and use of resources in the system.

Kill

```
root@kali: ~
Session Actions Edit View Help

(root@kali)-[~]
# kill 3893

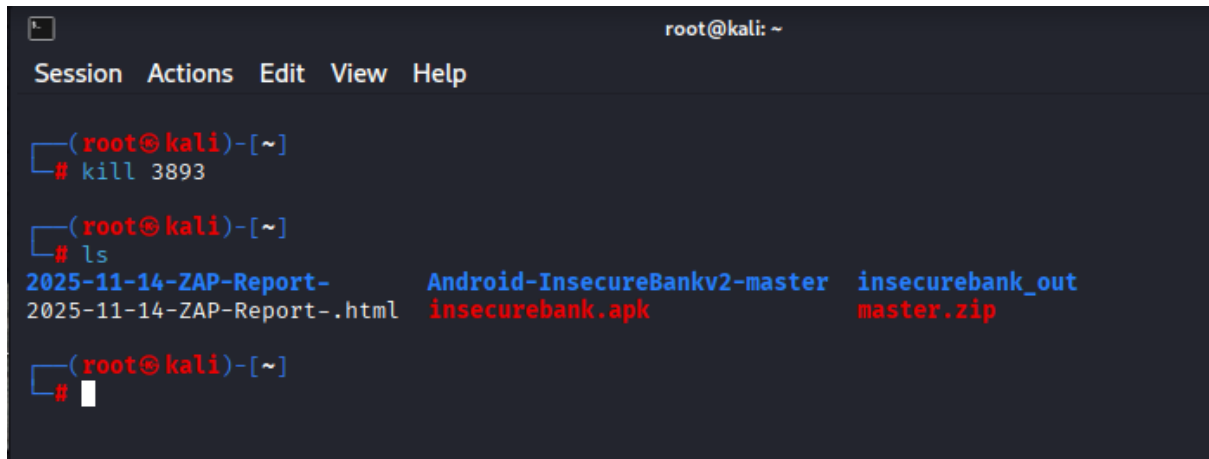
(root@kali)-[~]
#
```

Figure 17: kill 3893

The process is terminated through the Process ID.

File Management

Ls



```
root@kali: ~  
Session Actions Edit View Help  
(root@kali)-[~]  
# kill 3893  
(root@kali)-[~]  
# ls  
2025-11-14-ZAP-Report-  Android-InsecureBankv2-master  insecurebank_out  
2025-11-14-ZAP-Report-.html  insecurebank.apk               master.zip  
(root@kali)-[~]  
#
```

Figure 18:lsmkdir

Lists files and directories within the current directory.

cd

cd foldername

Changes the current working directory.

mkdir

mkdir newfolder

Creates a new directory.

cp

cp file.txt folder/

Copies files from one location to another.

rm

```
ethical@kali: ~/Desktop
Session Actions Edit View Help
zsh: corrupt history file /home/ethical/.zsh_history
(ethical@kali)-[~]
$ cd Desktop

(ethical@kali)-[~/Desktop]
$ ls
Files file.txt Nessus-10.11.0-ubuntu1604_amd64.deb

(ethical@kali)-[~/Desktop]
$ mkdir New

(ethical@kali)-[~/Desktop]
$ cp file.txt New/

(ethical@kali)-[~/Desktop]
$ ls
Files file.txt Nessus-10.11.0-ubuntu1604_amd64.deb New

(ethical@kali)-[~/Desktop]
$ ls New
file.txt

(ethical@kali)-[~/Desktop]
$ rm file.txt

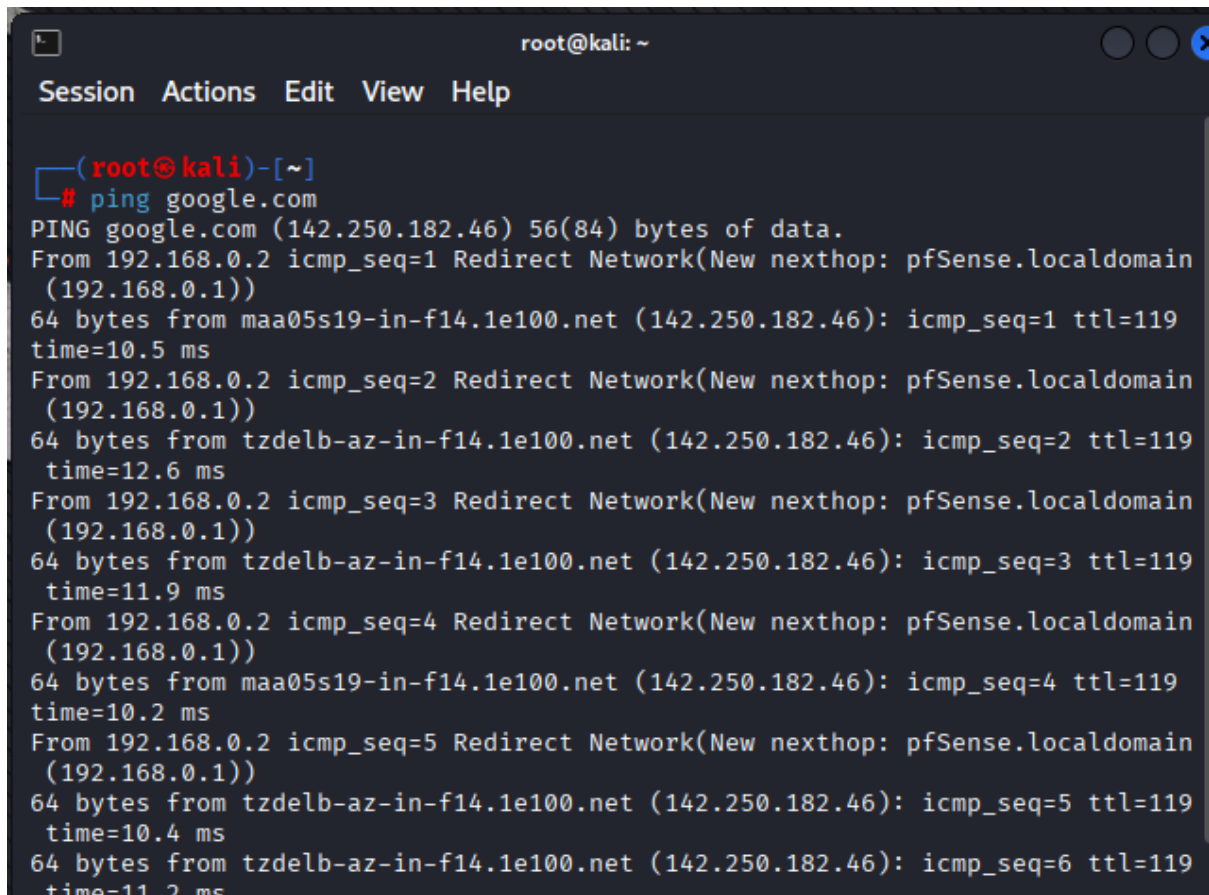
(ethical@kali)-[~/Desktop]
$
```

Figure 19:rm file.txt

Deletes a file from the system.

Network Monitoring

ping



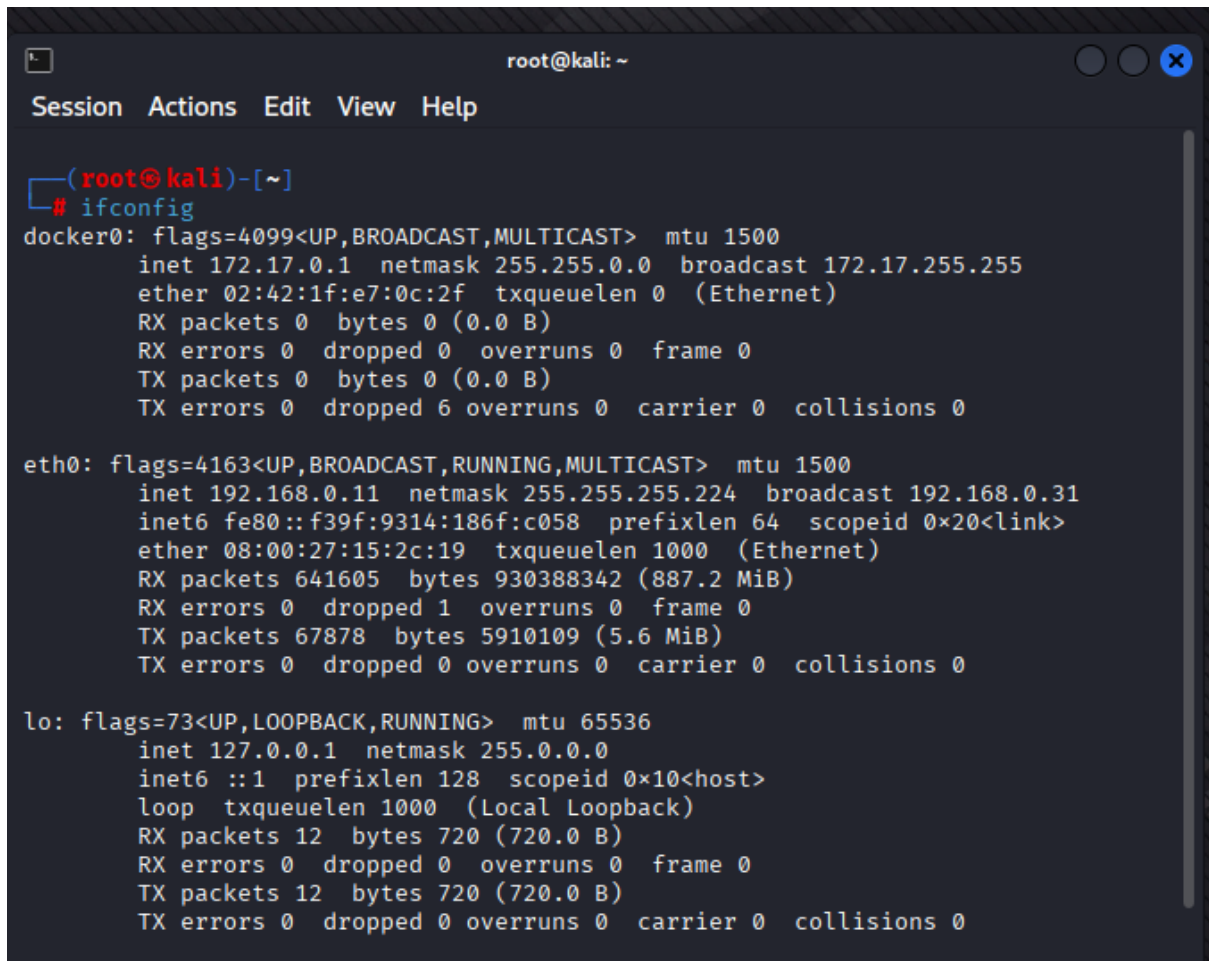
```
root@kali: ~
Session Actions Edit View Help

(root@kali)-[~]
# ping google.com
PING google.com (142.250.182.46) 56(84) bytes of data.
From 192.168.0.2 icmp_seq=1 Redirect Network(New nexthop: pfSense.localdomain
(192.168.0.1))
64 bytes from maa05s19-in-f14.1e100.net (142.250.182.46): icmp_seq=1 ttl=119
time=10.5 ms
From 192.168.0.2 icmp_seq=2 Redirect Network(New nexthop: pfSense.localdomain
(192.168.0.1))
64 bytes from tzdelb-az-in-f14.1e100.net (142.250.182.46): icmp_seq=2 ttl=119
time=12.6 ms
From 192.168.0.2 icmp_seq=3 Redirect Network(New nexthop: pfSense.localdomain
(192.168.0.1))
64 bytes from tzdelb-az-in-f14.1e100.net (142.250.182.46): icmp_seq=3 ttl=119
time=11.9 ms
From 192.168.0.2 icmp_seq=4 Redirect Network(New nexthop: pfSense.localdomain
(192.168.0.1))
64 bytes from maa05s19-in-f14.1e100.net (142.250.182.46): icmp_seq=4 ttl=119
time=10.2 ms
From 192.168.0.2 icmp_seq=5 Redirect Network(New nexthop: pfSense.localdomain
(192.168.0.1))
64 bytes from tzdelb-az-in-f14.1e100.net (142.250.182.46): icmp_seq=5 ttl=119
time=10.4 ms
64 bytes from tzdelb-az-in-f14.1e100.net (142.250.182.46): icmp_seq=6 ttl=119
time=11.2 ms
```

Figure 20: ping google.com

Tests connectivity to a remote host.

ifconfig



```
root@kali: ~  
Session Actions Edit View Help  
  
(root@kali)~  
# ifconfig  
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255  
    ether 02:42:1f:e7:0c:2f txqueuelen 0 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 6 overruns 0 carrier 0 collisions 0  
  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.0.11 netmask 255.255.255.224 broadcast 192.168.0.31  
    inet6 fe80::f39f:9314:186f:c058 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:15:2c:19 txqueuelen 1000 (Ethernet)  
    RX packets 641605 bytes 930388342 (887.2 MiB)  
    RX errors 0 dropped 1 overruns 0 frame 0  
    TX packets 67878 bytes 5910109 (5.6 MiB)  
    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 12 bytes 720 (720.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 12 bytes 720 (720.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 21: ifconfig

Displays network interface configuration.

netstat

```

root@kali: ~
Session Actions Edit View Help

(root@kali)-[~]
# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 192.168.0.11:bootpc    192.168.0.2:bootps     ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State         I-Node  Path
unix   3      [ ]      STREAM    CONNECTED    43966
unix   3      [ ]      STREAM    CONNECTED    42946
unix   3      [ ]      STREAM    CONNECTED    42637
unix   3      [ ]      STREAM    CONNECTED    43291    @/tmp/.X11-unix/X0
unix   3      [ ]      STREAM    CONNECTED    11430
unix   3      [ ]      STREAM    CONNECTED    50088
unix   3      [ ]      STREAM    CONNECTED    42931
unix   3      [ ]      STREAM    CONNECTED    14935
unix   3      [ ]      STREAM    CONNECTED    50809    /tmp/dbus-piiFxDZRdM
unix   3      [ ]      STREAM    CONNECTED    42875    /run/dbus/system_bus_
unix   3      [ ]      STREAM    CONNECTED    41813    /run/dbus/system_bus_
socket
unix   3      [ ]      STREAM    CONNECTED    14724    /run/systemd/journal/
stdout
unix   3      [ ]      STREAM    CONNECTED    14337
unix   3      [ ]      DGRAM     CONNECTED    7025
unix   3      [ ]      STREAM    CONNECTED    12979    /run/user/1000/bus
unix   3      [ ]      STREAM    CONNECTED    42703
unix   3      [ ]      STREAM    CONNECTED    15711
unix   3      [ ]      STREAM    CONNECTED    50915    /run/dbus/system_bus_
socket
unix   3      [ ]      STREAM    CONNECTED    43835
unix   3      [ ]      STREAM    CONNECTED    43573    @/tmp/.ICE-unix/3401
unix   3      [ ]      STREAM    CONNECTED    42759    /run/user/1000/bus
unix   3      [ ]      STREAM    CONNECTED    42618    @/tmp/.X11-unix/X0
unix   3      [ ]      STREAM    CONNECTED    13466    /run/systemd/journal/
stdout
unix   3      [ ]      STREAM    CONNECTED    10005
unix   3      [ ]      STREAM    CONNECTED    50147
unix   3      [ ]      STREAM    CONNECTED    14825
unix   3      [ ]      STREAM    CONNECTED    13399
unix   3      [ ]      STREAM    CONNECTED    42811    /run/systemd/journal/
stdout

```

Figure 22: netstatclear

Shows network connections and routing information.

User Management

adduser

sudo adduser username

Creates a new user account.

passwd

```
root@kali: ~
Session Actions Edit View Help

(root@kali)-[~]
# adduser usertest
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for usertest
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Figure 23: passwd username

Changes the password for a user account.

Task Scheduling

crontab

```
ethical@kali: ~
Session Actions Edit View Help

(ethical@kali)-[~]
$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * * echo "Hello" >> test.txt
```

Figure 24: crontab -l

Displays refer to activities that are carried out automatically on the system.

Task 4 – Windows Batch Scripts

Windows Batch scripts are plain text, which are given a set of commands, contemplated by the windows command interpreter (Command Prompt) [7]. The scripts come in handy in automation of administrative processes that are routine such as file management systems checks and basic calculations. A batch scripting can also be used to have variables, user input, loops and conditional statements to control the flow of the program. Using Batch scripts, the system administrators and users can achieve their usual business using it more effectively, they do less manual jobs and also improve their productivity in windows operating systems.

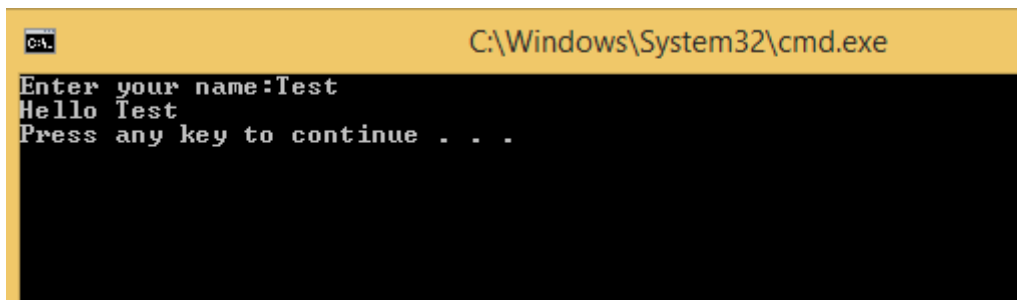
Script 1 – User Greeting

```
@echo off
```

```
set /p name=Enter your name:
```

```
echo Hello %name%
```

```
pause
```



In this script, the user is asked to enter his/her name and once this is done he/she is presented with a greeting message. It demonstrates the usage of the user input and variables.

Script 2 – Simple Calculator

```
@echo off
```

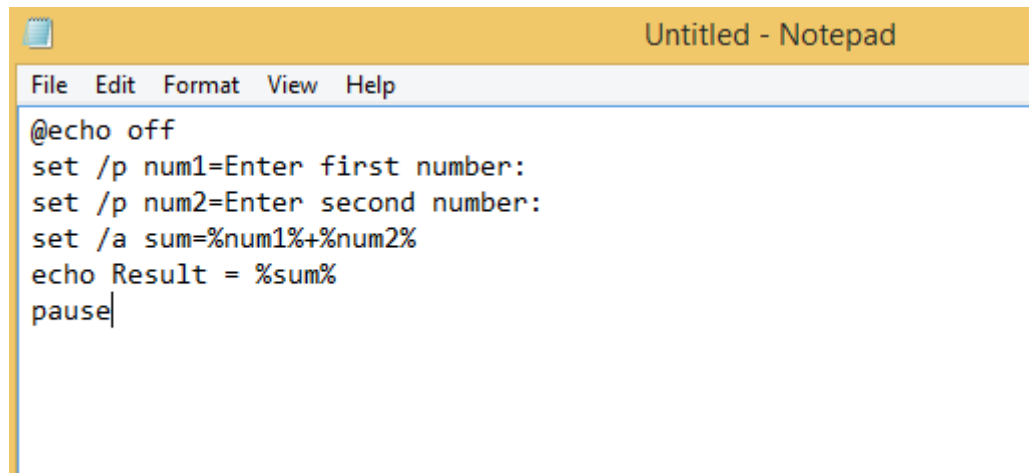
```
set /p num1=Enter first number:
```

```
set /p num2=Enter second number:
```

```
set /a sum=%num1%+%num2%
```

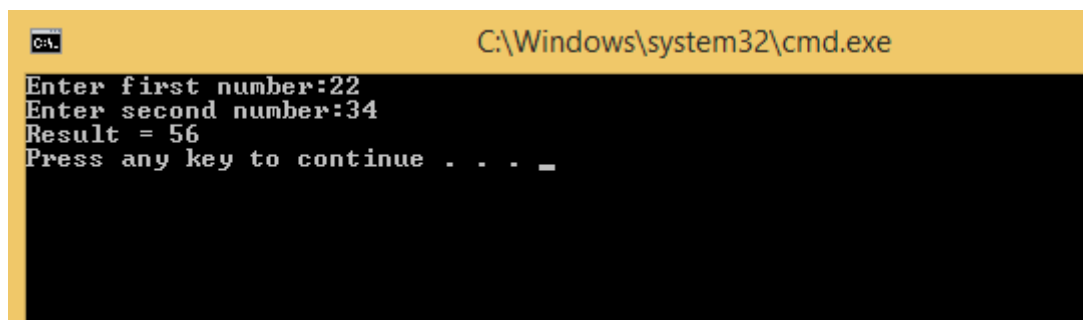
```
echo Result = %sum%
```

```
pause
```



```
@echo off
set /p num1=Enter first number:
set /p num2=Enter second number:
set /a sum=%num1%+%num2%
echo Result = %sum%
pause
```

Figure 25: Script Code



```
C:\Windows\system32\cmd.exe
Enter first number:22
Enter second number:34
Result = 56
Press any key to continue . . . _
```

Figure 26: Script Output

In this program the user will be required to put in two numbers which will then perform a mathematical addition operation. It represents the use of variables and arithmetic operation.

Task 5 – Linux Bash Scripts

The Linux Bash scripting allows the user with the opportunity of automation and control of the system activities easily through the terminal [8]. Shell scripts consist of a sequence of commands and are run later. They will support variables, user input, loops and condition statement which the administrator can automate repetitive operations, managing files, system monitoring and maintaining regular task of Linux.

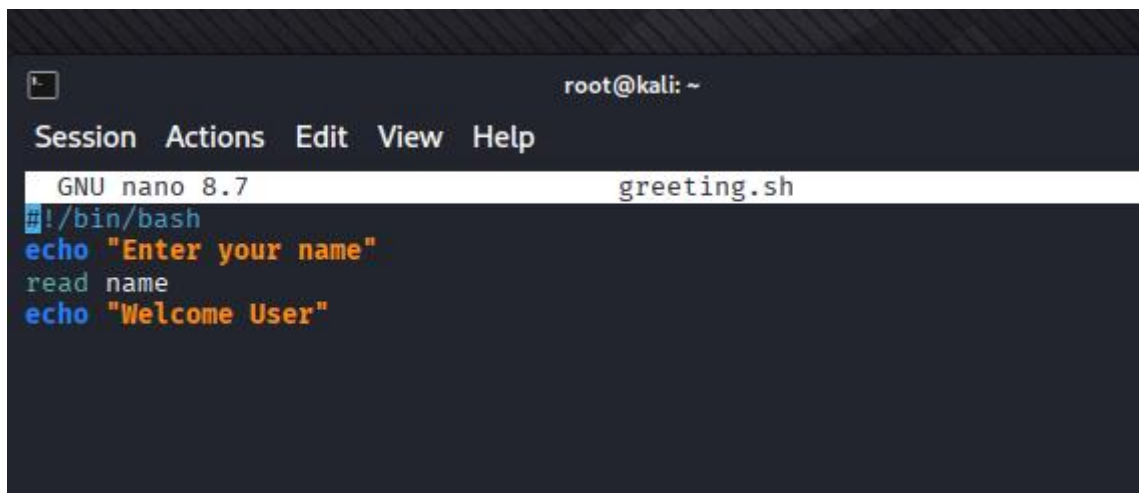
Script 1 – Greeting Script

```
#!/bin/bash

echo "Enter your name"

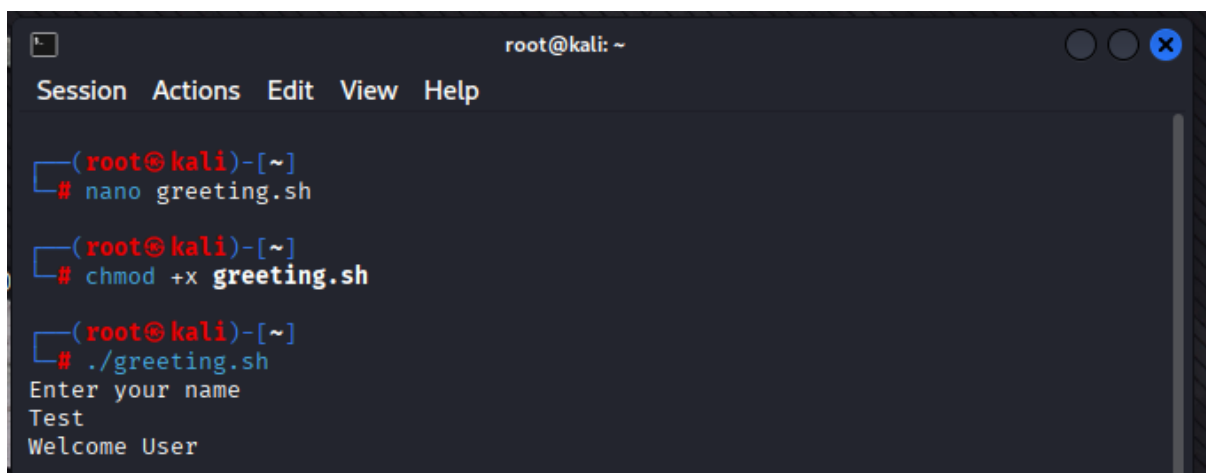
read name

echo "Welcome User"
```



```
root@kali: ~
Session Actions Edit View Help
GNU nano 8.7 greeting.sh
#!/bin/bash
echo "Enter your name"
read name
echo "Welcome User"
```

Figure 27: Greeting Script



```
root@kali: ~
Session Actions Edit View Help
(root@kali)-[~]
# nano greeting.sh
(root@kali)-[~]
# chmod +x greeting.sh
(root@kali)-[~]
# ./greeting.sh
Enter your name
Test
Welcome User
```

Figure 28: Script Output

This script identifies the name of the user and welcomes the user. it forecasts variables and users utilized and entered in Bash.

Script 2 – Conditional Script

```
#!/bin/bash

echo "Enter number"

read num

if [ $num -gt 10 ]

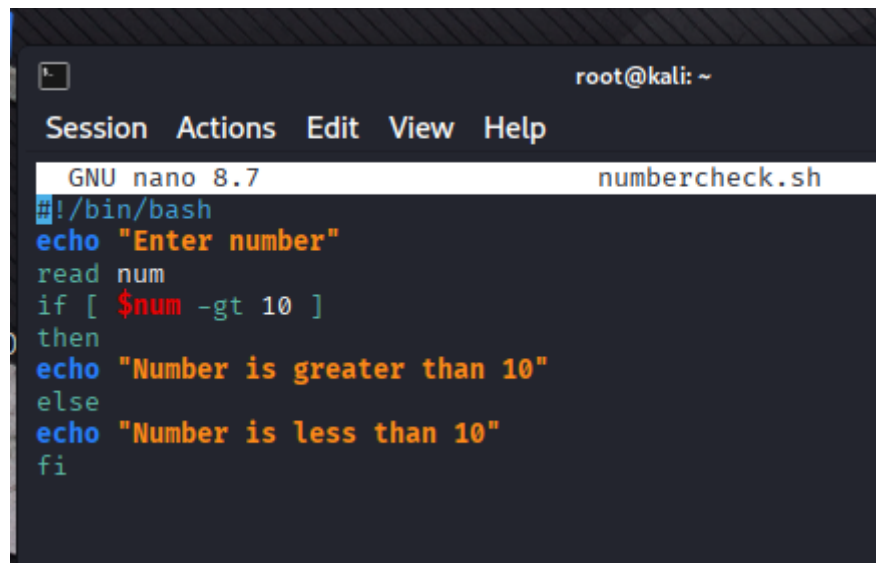
then

echo "Number is greater than 10"

else
```

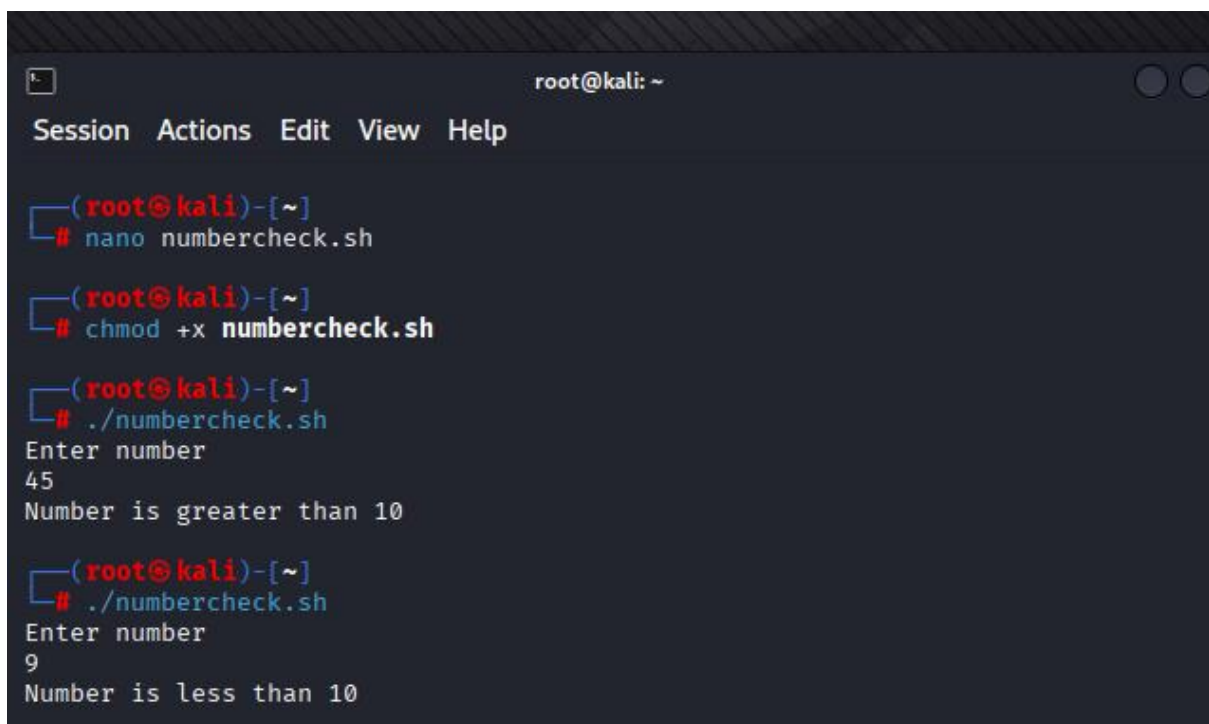
```
echo "Number is less than 10"
```

```
fi
```



```
root@kali: ~  
Session Actions Edit View Help  
GNU nano 8.7 numbercheck.sh  
#!/bin/bash  
echo "Enter number"  
read num  
if [ $num -gt 10 ]  
then  
echo "Number is greater than 10"  
else  
echo "Number is less than 10"  
fi
```

Figure 29: Script



```
root@kali: ~  
Session Actions Edit View Help  
  
(root@kali)-[~]  
# nano numbercheck.sh  
  
(root@kali)-[~]  
# chmod +x numbercheck.sh  
  
(root@kali)-[~]  
# ./numbercheck.sh  
Enter number  
45  
Number is greater than 10  
  
(root@kali)-[~]  
# ./numbercheck.sh  
Enter number  
9  
Number is less than 10
```

Figure 30: Output Screen

This code involves the user to enter a number and affirm with a conditional statement whether the number is greater than 10 or not.

Conclusion

The current computer systems heavily depend on the operating systems since they are the ones that control the hardware resources and even offer a platform through which the software programs can successfully execute. Processes management, file system, device, security, and networking management are the most important functions of the operating systems as have been explained in this report. As shown by the practical activities, command-line interfaces of the Windows and Linux operating systems can be utilized to undertake an administrative task, e.g., tracing processes, working with files, testing networks, and controlling user accounts.

The report also discussed the way to use the scripting in windows batch and the Linux bash. Another way of simplifying the system efficiency is that the users can also automate the repetitive tasks using scripts which reduces manual work. Across this assignment, one was able to grasp the fact that command-line tools provide incredible system management capabilities. Their familiarity with these tools and scripting languages is of importance to the IT experts because they will assist them to administer the operating systems and troubleshoots.

References

- [1] Z. Bin Akhtar, “Operating Systems (OS): An Insight Investigative Research Analysis and Future Directions,” *Journal of Technology and Informatics (JoTI)*, vol. 6, no. 1, pp. 58–69, Oct. 2024, doi: <https://doi.org/10.37802/joti.v6i1.637>.
- [2] S. Jia, X. Wang, M. Song, and G. Chen, “Agent Centric Operating System -- a Comprehensive Review and Outlook for Operating System,” *arXiv.org*, 2024. <https://arxiv.org/abs/2411.17710> (accessed Mar. 13, 2026).
- [3] O. Ali, M. K. Ishak, M. K. L. Bhatti, I. Khan, and K.-I. Kim, “A Comprehensive Review of Internet of Things: Technology Stack, Middlewares, and Fog/Edge Computing Interface,” *Sensors*, vol. 22, no. 3, p. 995, Jan. 2022, doi: <https://doi.org/10.3390/s22030995>.
- [4] Z. Bin Akhtar, “Operating Systems (OS): An Insight Investigative Research Analysis and Future Directions,” *Journal of Technology and Informatics (JoTI)*, vol. 6, no. 1, pp. 58–69, Oct. 2024, doi: <https://doi.org/10.37802/joti.v6i1.637>.
- [5] P. Liguori, C. Marescalco, R. Natella, V. Orbinato, and L. Pianese, “The Power of Words: Generating PowerShell Attacks from Natural Language,” in *Proc. 18th USENIX WOOT Conf. on Offensive Technologies (WOOT 24)*, 2024, pp. 27–43. [Online]. Available: <https://www.usenix.org/system/files/woot24-liguori.pdf>
- [6] H. Wicaksono, F. Azzahra, and H. Setiawan, “Human–Computer Interaction Enhancement for Linux CLI Application Using Telegram Bot Piping,” *SAINTEKBU*, vol. 16, no. 2, pp. 71–79, 2024. [Online]. Available: <http://ejournal.unwaha.ac.id/index.php/sainstek/article/download/4764/2723>
- [7] A. N. Marsh, V. Sharma, S. K. Mani, O. Vitek, M. J. MacCoss, and B. X. MacLean, “Skyline batch: An intuitive user interface for batch processing with Skyline,” *Journal of Proteome Research*, vol. 21, no. 1, pp. 289–294, 2021. [Online]. Available: <https://pubs.acs.org/doi/pdf/10.1021/acs.jproteome.1c00749>
- [8] B. Ward, *How Linux Works: What Every Superuser Should Know*. San Francisco, CA, USA: No Starch Press, 2021. [Online]. Available: https://trendoceans.com/wp-content/uploads/2022/03/How-Linux-Works_-What-Every-Superuser-Should-Know-PDFDrive-.pdf

Appendix

Key functions of general Operating Systems

AI Mode **All** Images Videos Shopping Forums Short videos More ▾ Tools ▾

✦ AI Overview हिन्दी 🔊

Operating systems (OS) manage computer hardware and software, acting as an intermediary between users and the machine. Core functions include **processor scheduling (CPU management), memory allocation, file system management, device input/output (I/O) control, security, and user interface provision**. These ensure efficient, conflict-free, and secure system operation. IBM +5



Key Functions of an Operating System:

- **Process Management (CPU Scheduling):** Handles the creation, scheduling, and termination of processes, allowing multiple tasks to run simultaneously without interference.
- **Memory Management:** Manages RAM, allocating and deallocating memory space to processes to ensure efficient usage. It also handles virtual memory (swapping) when physical memory is full.
- **File Management:** Manages the creation, deletion, and organization of files and directories on storage devices (HDD, SSD), ensuring data is stored and retrieved efficiently.
- **Device Management (I/O Management):** Manages communication with peripheral devices (printers, scanners, monitors) via device drivers.

✦ AI Overview हिन्दी 🔊

The Windows Command Line Interface (CLI), primarily accessed via **Command Prompt (CMD)** or **PowerShell**, is a text-based tool used for system administration, file management, and automation. It offers precise control and efficiency often not available through the graphical user interface (GUI). [Amazon Web Services +3](#)

Accessing the Windows CLI

You can access the CLI using several methods:

- **Run Dialog:** Press the Windows Key + R, type `cmd`, and press Enter.
- **Start Menu Search:** Click the Start menu, type "Command Prompt" or "cmd" in the search bar, and press Enter.
- **Run as Administrator:** For tasks that require elevated permissions, right-click "Command Prompt" in the Start menu search results and select **Run as administrator**. [NinjaOne +2](#)

Essential Windows CLI Commands

Below are some commonly used commands, categorized by function: [Learn to Code — For Free +1](#)

Use of Windows OS scripts

AI Mode

All

Images

Videos

Shopping

Forums

Short videos

More ▾

Tools ▾

✦ AI Overview हिन्दी 🔊

Windows OS scripts, primarily using **PowerShell (.ps1)** and **Batch (.bat)** files, are powerful text-based tools for automating repetitive tasks, managing system configurations, and executing complex commands without a GUI. They are essential for system administration, such as deploying software, managing user accounts in Active Directory, backing up data, and modifying registry settings.  Microsoft Learn +4

Linux OS scripts, commonly known as **shell scripts**, are programs that **automate a sequence of commands, system tasks, and administrative functions**. They serve as an interface between the user and the operating system's kernel, allowing complex interactions to be simplified into single, repeatable commands. [GeeksforGeeks +3](#)

Primary Uses of Linux OS Scripts

- **Task Automation:** Shell scripts automate repetitive, time-consuming tasks like file cleanup, routine backups, and software installations, which saves significant time and reduces manual errors.
- **System Administration:** System administrators use scripts to manage users and permissions, monitor system resources (CPU, memory, disk space), manage processes, and handle logging and monitoring.
- **Process and Workflow Control:** Scripts can link together different command-line utilities and programs using pipes and redirection, allowing the output of one command to become the input of another, creating complex workflows.
- **Scheduling Jobs:** They can be scheduled to run at specific times or intervals using utilities like `cron`, which is essential for maintenance and regular updates.
- **Development and DevOps:** Developers and DevOps professionals use scripts to