Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre

A Lab Report
on
“Operating System Lab-I”

[Code No.: COMP 307]

Submitted by

Shreyash Mahato
Roll No.: 33

Submitted to
Ms. Rabina Shrestha
Department of Computer Science and Engineering

December 10, 2025

Questions

Q1: What is Linux?

Linux is an open-source operating system inspired by the Unix design. It controls hardware components, runs programs, and offers a stable, secure multitasking environment. It is widely used in servers, personal computers, embedded systems, cloud infrastructure, and even supercomputers. Popular Linux distributions include Ubuntu, Arch Linux, and Red Hat.

Q2: The Linux Hierarchical File System

Linux organizes data using a hierarchical file system that begins at the root directory (/). Every component in Linux is treated as a file or directory, and all paths originate from /. Important directories include:

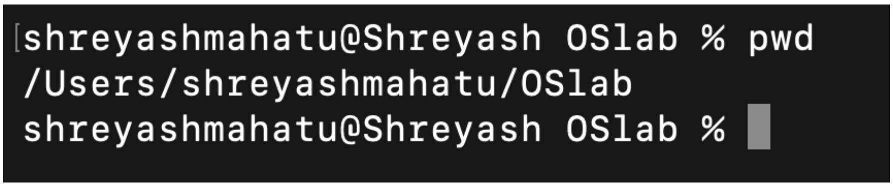
- /home – Contains user-specific directories
- /bin – Essential system commands and binaries
- /etc – System and application configuration files
- /usr – User applications, libraries, and utilities
- /var – Log files, caches, and variable data

Q3: Importance of Linux commands in Operating Systems

Linux commands provide powerful, direct interaction with the operating system. They help users navigate directories, manage files, monitor the system, and automate repetitive tasks. Compared to graphical interfaces, the command line is faster, more efficient, and offers deeper control. Understanding these commands enhances productivity, improves troubleshooting, and builds strong foundational knowledge—especially for developers, system administrators, and advanced users.

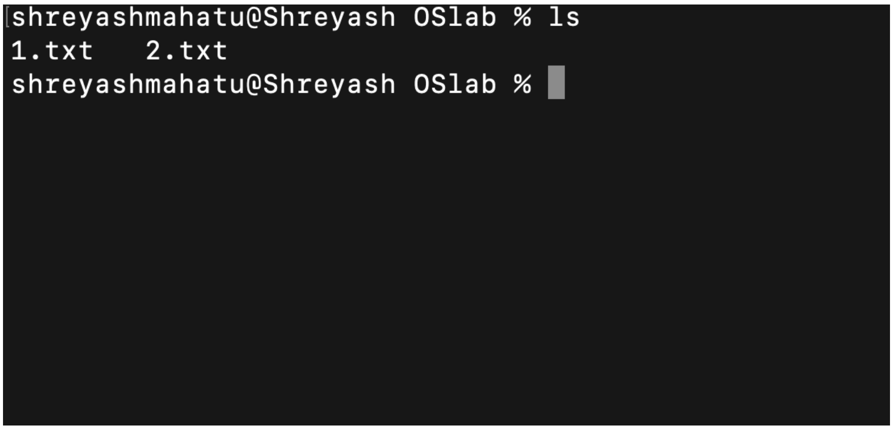
Linux Commands

1. **`pwd`**
   The pwd command displays your current directory path. It is helpful when navigating complex directory structures, verifying locations, or writing scripts. Since Linux follows a root-based hierarchical structure, pwd ensures you always know your exact position in the filesystem.



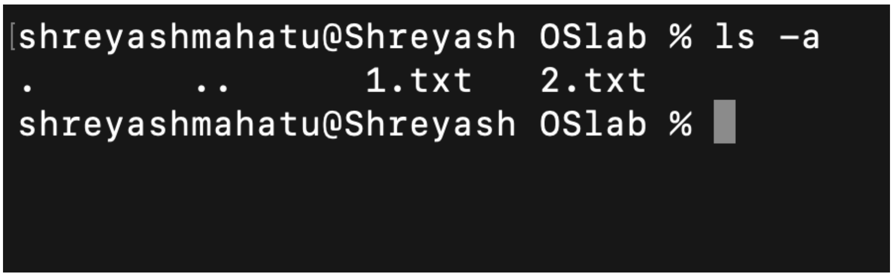
```
[shreyashmahatu@Shreyash OSlab % pwd
/Users/shreyashmahatu/OSlab
shreyashmahatu@Shreyash OSlab %
```

2. **`ls`**
   The ls command shows all visible files and folders in the current directory. It provides a quick overview of directory contents and is one of the most frequently used commands in Linux.



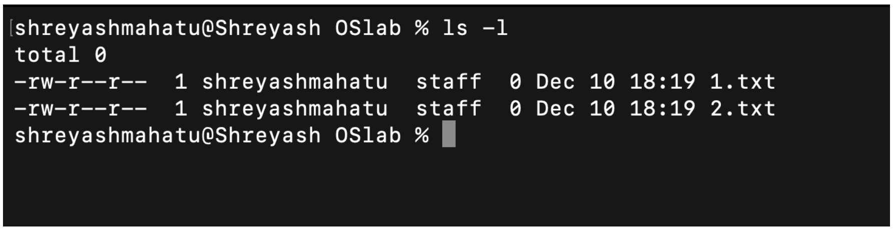
```
[shreyashmahatu@Shreyash OSlab % ls
1.txt   2.txt
shreyashmahatu@Shreyash OSlab %
```

3. **`ls -a`**
   This variation of ls lists every file, including hidden ones (prefixed with a dot, like .bashrc). Hidden files generally store configurations and user preferences.



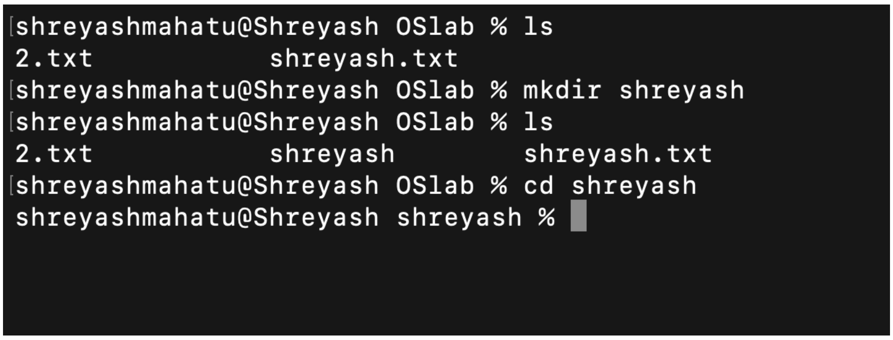
```
[shreyashmahatu@Shreyash OSlab % ls -a
.       ..      1.txt   2.txt
shreyashmahatu@Shreyash OSlab %
```

4. **`ls -l`**
   The -l option shows a long, detailed listing of files. It includes permissions, ownership, size, and modification dates. This format is especially useful for managing access control and analyzing file properties.



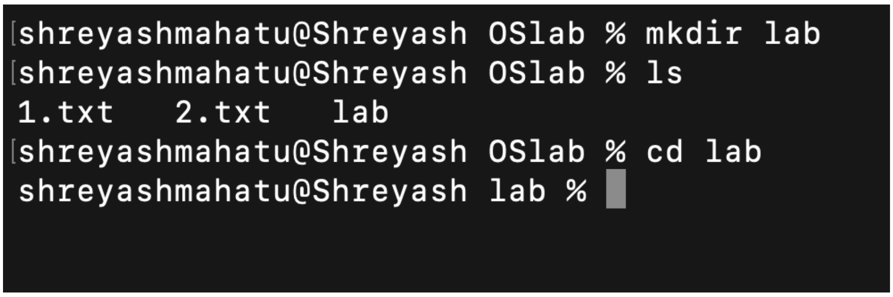
```
[shreyashmahatu@Shreyash OSlab % ls -l
total 0
-rw-r--r--  1 shreyashmahatu  staff  0 Dec 10 18:19 1.txt
-rw-r--r--  1 shreyashmahatu  staff  0 Dec 10 18:19 2.txt
shreyashmahatu@Shreyash OSlab % 
```

5. **cd**
   The cd command changes your current directory. You can navigate to subdirectories, return to the parent directory using `cd ..`, or move directly to a specific absolute or relative path.



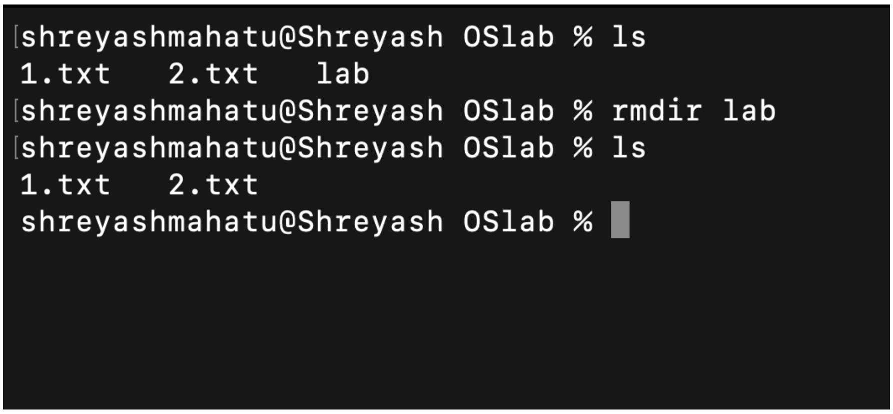
```
[shreyashmahatu@Shreyash OSlab % ls
2.txt           shreyash.txt
[shreyashmahatu@Shreyash OSlab % mkdir shreyash
[shreyashmahatu@Shreyash OSlab % ls
2.txt           shreyash        shreyash.txt
[shreyashmahatu@Shreyash OSlab % cd shreyash
shreyashmahatu@Shreyash shreyash % 
```

6. **mkdir**
   mkdir creates a new directory. It is used for organizing files into folders. With the -p option, you can also create nested directories in one command.



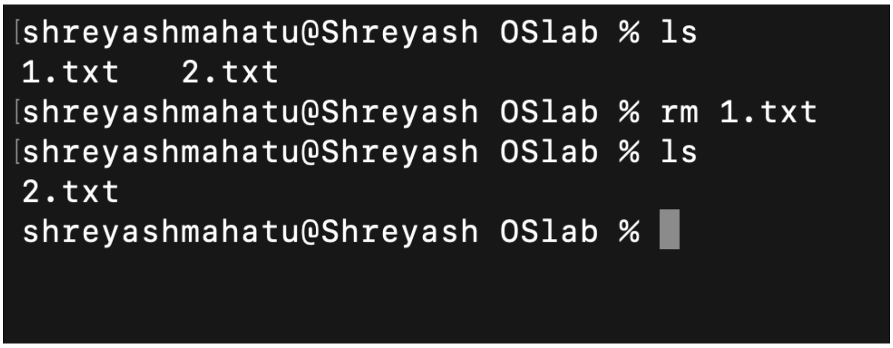
```
[shreyashmahatu@Shreyash OSlab % mkdir lab
[shreyashmahatu@Shreyash OSlab % ls
1.txt   2.txt   lab
[shreyashmahatu@Shreyash OSlab % cd lab
shreyashmahatu@Shreyash lab % 
```

7. **rmdir**
   rmdir removes an empty directory. It cannot delete directories containing files. It is useful when cleaning up unused or temporary folders.



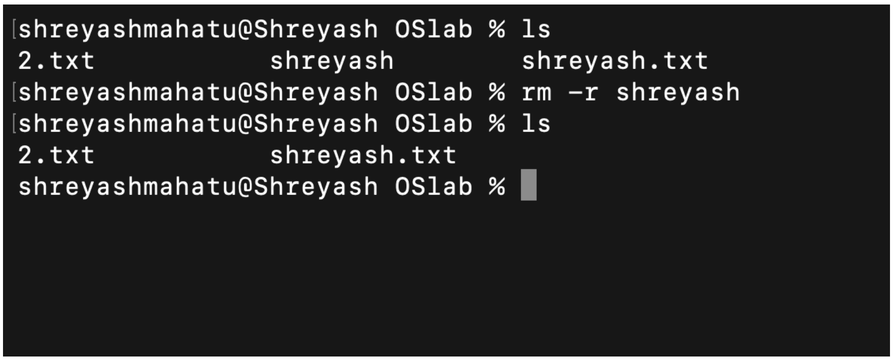
```
[shreyashmahatu@Shreyash OSlab % ls
1.txt   2.txt   lab
[shreyashmahatu@Shreyash OSlab % rmdir lab
[shreyashmahatu@Shreyash OSlab % ls
1.txt   2.txt
shreyashmahatu@Shreyash OSlab % 
```

8. **`rm`**
   The rm command permanently deletes files. Since there is no recycle bin, caution is required. It also supports deleting multiple files in a single command.



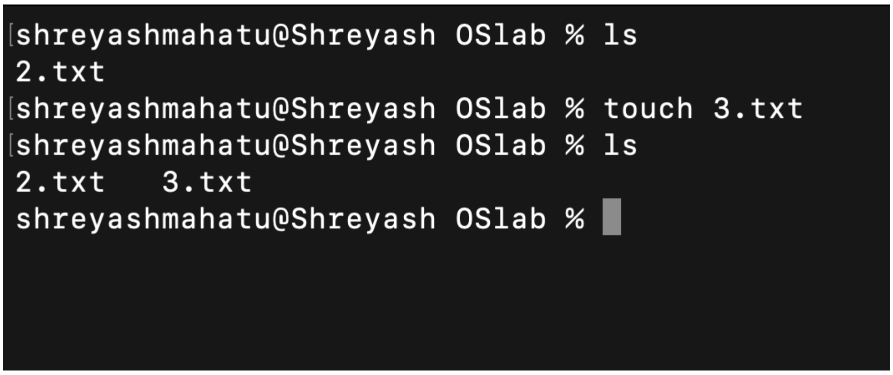
```
[shreyashmahatu@Shreyash OSlab % ls
1.txt   2.txt
[shreyashmahatu@Shreyash OSlab % rm 1.txt
[shreyashmahatu@Shreyash OSlab % ls
2.txt
shreyashmahatu@Shreyash OSlab % 
```

9. **`rm -r folder_name`**
   The -r (recursive) option deletes a directory along with all its subdirectories and files. This command is powerful and should be used carefully to avoid accidental data loss.



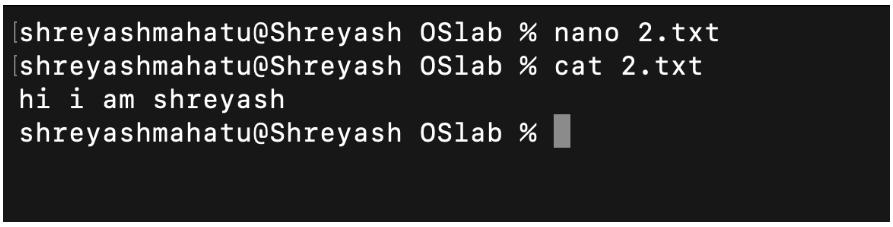
```
[shreyashmahatu@Shreyash OSlab % ls
2.txt           shreyash        shreyash.txt
[shreyashmahatu@Shreyash OSlab % rm -r shreyash
[shreyashmahatu@Shreyash OSlab % ls
2.txt           shreyash.txt
shreyashmahatu@Shreyash OSlab % 
```

10. **`touch`**
    touch creates a new empty file or updates the timestamp of an existing one. It is frequently used when preparing files for scripting or development work.



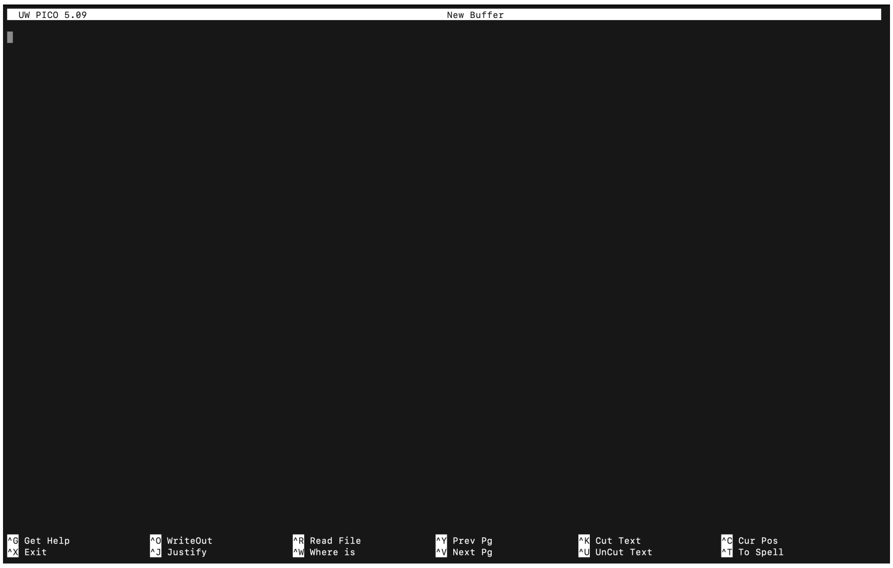
```
shreyashmahatu@Shreyash OSlab % ls
2.txt
shreyashmahatu@Shreyash OSlab % touch 3.txt
shreyashmahatu@Shreyash OSlab % ls
2.txt   3.txt
shreyashmahatu@Shreyash OSlab %
```

11. **`cat`**
    cat displays the content of a file directly in the terminal. It can also be used to merge files or create new files with redirection operators.

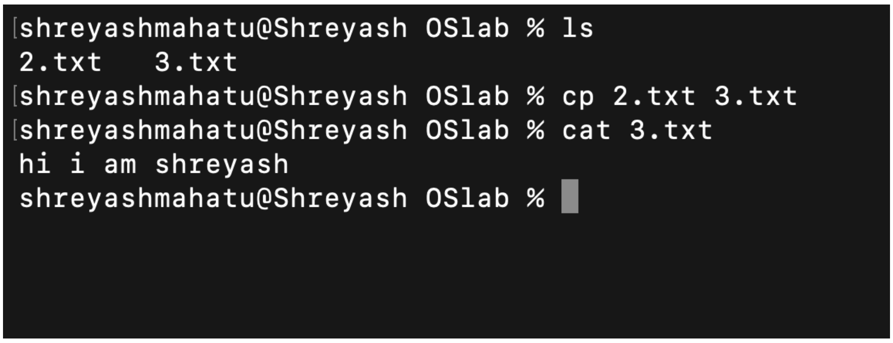


```
shreyashmahatu@Shreyash OSlab % nano 2.txt
shreyashmahatu@Shreyash OSlab % cat 2.txt
hi i am shreyash
shreyashmahatu@Shreyash OSlab %
```

12. **`nano, vi, jed`**
    These are terminal-based editors for creating and editing files. nano is user-friendly, vi/vim is a highly powerful editor preferred by developers, and jed offers a lightweight editing interface.

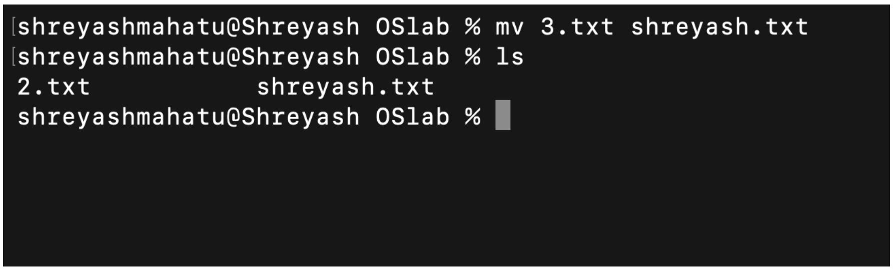


13. **`cp`**
    cp copies files or directories (using -r for folders). It is useful for backups, duplication, and reorganizing directory structures.



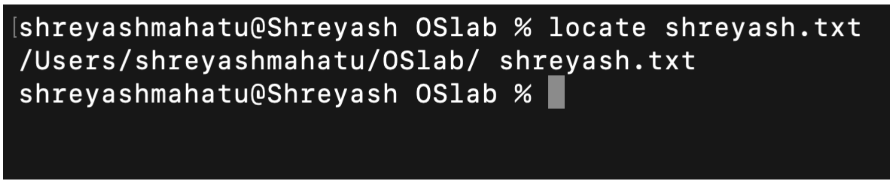
```
[shreyashmahatu@Shreyash OSlab % ls
2.txt   3.txt
[shreyashmahatu@Shreyash OSlab % cp 2.txt 3.txt
[shreyashmahatu@Shreyash OSlab % cat 3.txt
hi i am shreyash
shreyashmahatu@Shreyash OSlab % 
```

14. **mv**
    mv moves or renames files and directories. Renaming is simply a move within the same directory. It is used for reorganizing or updating file names efficiently.



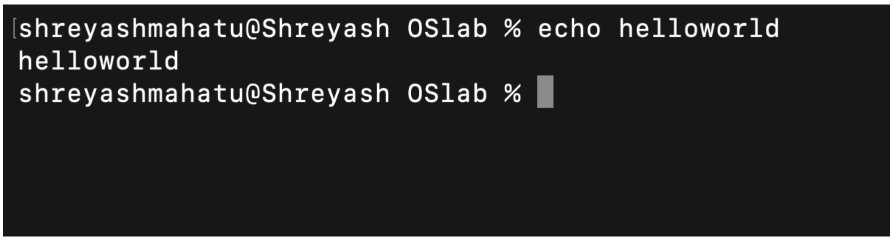
```
[shreyashmahatu@Shreyash OSlab % mv 3.txt shreyash.txt
[shreyashmahatu@Shreyash OSlab % ls
2.txt           shreyash.txt
shreyashmahatu@Shreyash OSlab % 
```

15. **locate**
    locate quickly searches for files by name using a pre-indexed database, making searches extremely fast. It is ideal for finding lost or unknown file locations.



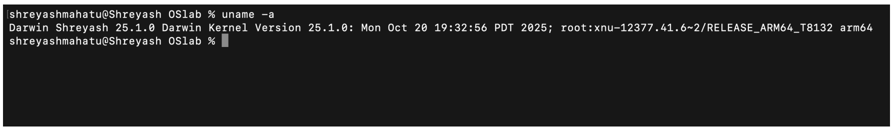
```
[shreyashmahatu@Shreyash OSlab % locate shreyash.txt
/Users/shreyashmahatu/OSlab/ shreyash.txt
shreyashmahatu@Shreyash OSlab % 
```

16. **echo**
    echo prints text or variable values to the terminal. It is widely used in scripting to display messages, check variable values, or write content into files.



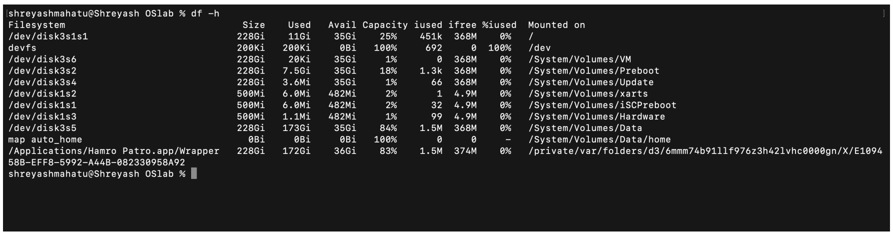
```
[shreyashmahatu@Shreyash OSlab % echo helloworld
helloworld
shreyashmahatu@Shreyash OSlab % 
```

17. **uname -a**
    This command shows detailed system information, including kernel version, architecture, hostname, and OS type. It is useful for diagnostics and environment checks.



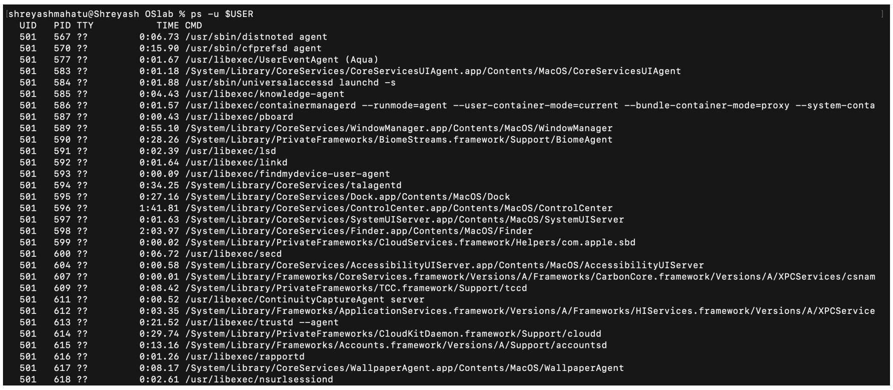
```
shreyashmahatu@Shreyash OSlab % uname -a
Darwin Shreyash 25.1.0 Darwin Kernel Version 25.1.0: Mon Oct 20 19:32:56 PDT 2025; root:xnu-12377.41.6~2/RELEASE_ARM64_T8132 arm64
shreyashmahatu@Shreyash OSlab % 
```

18. **`df -h`**
    df -h displays disk usage in a readable format (GB/MB). It provides details about total space, used space, available space, and mounted partitions—helpful for monitoring storage.



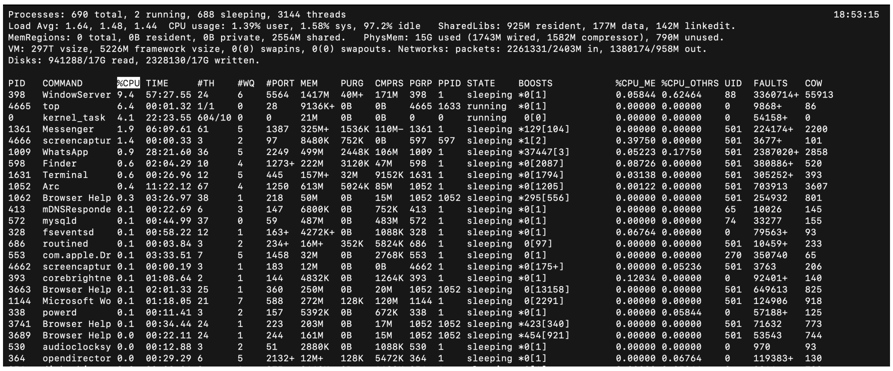
```
shreyashmahatu@Shreyash OSlab % df -h
Filesystem                          Size    Used   Avail Capacity iused ifree %iused  Mounted on
/dev/disk3s1s1                     228Gi    11Gi    35Gi    25%    451k  368M    0%   /
devfs                              200Ki   200Ki    0Bi   100%     692     0  100%   /dev
/dev/disk3s6                       228Gi    20Ki    35Gi     1%       0  368M    0%   /System/Volumes/VM
/dev/disk3s2                       228Gi   7.5Gi    35Gi    18%    1.3k  368M    0%   /System/Volumes/Preboot
/dev/disk3s4                       228Gi   3.6Mi    35Gi     1%      66  368M    0%   /System/Volumes/Update
/dev/disk1s2                       500Mi   6.0Mi   482Mi     2%       1  4.9M    0%   /System/Volumes/xarts
/dev/disk1s1                       500Mi   6.0Mi   482Mi     2%      32  4.9M    0%   /System/Volumes/iSCPreboot
/dev/disk1s3                       500Mi   1.1Mi   482Mi     1%      99  4.9M    0%   /System/Volumes/Hardware
/dev/disk3s5                       228Gi   173Gi    35Gi    84%    1.5M  368M    0%   /System/Volumes/Data
map auto_home                        0Bi     0Bi     0Bi   100%       0     0     -   /System/Volumes/Data/home
/Applications/Hamro Patro.app/Wrapper   228Gi   172Gi    36Gi    83%    1.5M  374M    0%   /private/var/folders/d3/6mmm74b91llf976z3h42lvhc0000gn/X/E1094
58B-EFF8-5992-A44B-082330958A92
shreyashmahatu@Shreyash OSlab % 
```

19. **`ps -u $USER`**
    This command lists all running processes for your user account. It includes process IDs, CPU usage, memory consumption, and executed commands—useful for identifying active or misbehaving processes.



```
shreyashmahatu@Shreyash OSlab % ps -u $USER
  UID   PID TTY           TIME CMD
  501   567 ??         0:06.73 /usr/sbin/distnoted agent
  501   570 ??         0:15.90 /usr/sbin/cfprefsd agent
  501   577 ??         0:01.67 /usr/libexec/UserEventAgent (Aqua)
  501   583 ??         0:01.18 /System/Library/CoreServices/CoreServicesUIAgent.app/Contents/MacOS/CoreServicesUIAgent
  501   584 ??         0:01.88 /usr/sbin/universalaccessd launchd -s
  501   585 ??         0:04.43 /usr/libexec/knowledge-agent
  501   586 ??         0:01.57 /usr/libexec/containermanagerd --runmode=agent --user-container-mode=current --bundle-container-mode=proxy --system-conta
  501   587 ??         0:00.43 /usr/libexec/pboard
  501   589 ??         0:55.10 /System/Library/CoreServices/WindowManager.app/Contents/MacOS/WindowManager
  501   590 ??         0:28.26 /System/Library/PrivateFrameworks/BiomeStreams.framework/Support/BiomeAgent
  501   591 ??         0:02.39 /usr/libexec/lsd
  501   592 ??         0:01.64 /usr/libexec/linkd
  501   593 ??         0:00.09 /usr/libexec/findmydevice-user-agent
  501   594 ??         0:34.25 /System/Library/CoreServices/talagentd
  501   595 ??         0:27.16 /System/Library/CoreServices/Dock.app/Contents/MacOS/Dock
  501   596 ??         1:41.81 /System/Library/CoreServices/ControlCenter.app/Contents/MacOS/ControlCenter
  501   597 ??         0:01.63 /System/Library/CoreServices/SystemUIServer.app/Contents/MacOS/SystemUIServer
  501   598 ??         2:03.97 /System/Library/CoreServices/Finder.app/Contents/MacOS/Finder
  501   599 ??         0:00.02 /System/Library/PrivateFrameworks/CloudServices.framework/Helpers/com.apple.sbd
  501   600 ??         0:06.72 /usr/libexec/secd
  501   604 ??         0:00.58 /System/Library/CoreServices/AccessibilityUIServer.app/Contents/MacOS/AccessibilityUIServer
  501   607 ??         0:00.01 /System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/CarbonCore.framework/Versions/A/XPCServices/csnam
  501   609 ??         0:08.42 /System/Library/PrivateFrameworks/TCC.framework/Support/tccd
  501   611 ??         0:00.52 /usr/libexec/ContinuityCaptureAgent server
  501   612 ??         0:03.35 /System/Library/Frameworks/ApplicationServices.framework/Versions/A/Frameworks/HIServices.framework/Versions/A/XPCService
  501   613 ??         0:21.52 /usr/libexec/trustd --agent
  501   614 ??         0:29.74 /System/Library/PrivateFrameworks/CloudKitDaemon.framework/Support/cloudd
  501   615 ??         0:13.16 /System/Library/Frameworks/Accounts.framework/Versions/A/Support/accountsd
  501   616 ??         0:01.26 /usr/libexec/rapportd
  501   617 ??         0:08.17 /System/Library/CoreServices/WallpaperAgent.app/Contents/MacOS/WallpaperAgent
  501   618 ??         0:02.61 /usr/libexec/nsurlsessiond
```

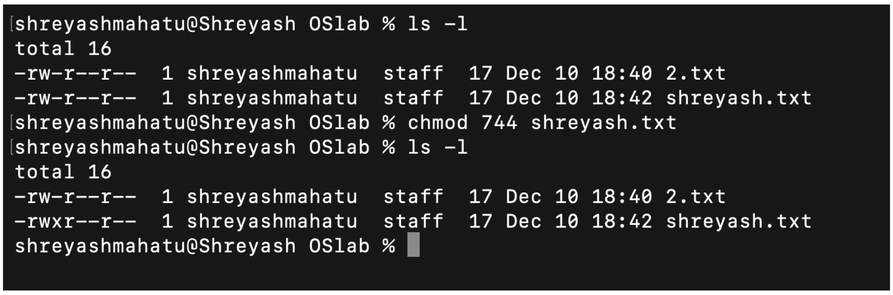
20. **`top`**
    top displays real-time system activity, including CPU and memory usage, active processes, and system load. It is one of the key tools for performance monitoring and troubleshooting.



```
Processes: 690 total, 2 running, 688 sleeping, 3144 threads                                                          18:53:15
Load Avg: 1.64, 1.48, 1.44  CPU usage: 1.39% user, 1.58% sys, 97.2% idle   SharedLibs: 925M resident, 177M data, 142M linkedit.
MemRegions: 0 total, 0B resident, 0B private, 2554M shared.   PhysMem: 15G used (1743M wired, 1582M compressor), 790M unused.
VM: 297T vsize, 5226M framework vsize, 0(0) swapins, 0(0) swapouts. Networks: packets: 2261331/2403M in, 1380174/958M out.
Disks: 941288/17G read, 2328130/17G written.

PID   COMMAND      %CPU TIME     #TH    #WQ  #PORT MEM    PURG   CMPRS  PGRP PPID STATE    BOOSTS           %CPU_ME %CPU_OTHRS UID  FAULTS   COW
398   WindowServer 9.4  57:27.55 24     6    5564  1417M  40M+   171M   398  1    sleeping *0[1]            0.05844 0.62464    88   3360714+ 55913
4665  top          6.4  00:01.32 1/1    0    28    9136K+ 0B     0B     4665 1633 running  *0[1]            0.00000 0.00000    0    9868+    86
0     kernel_task  4.1  22:23.55 604/10 0    0     21M    0B     0B     0    0    running   0[0]            0.00000 0.00000    0    54158+   0
1361  Messenger    1.9  06:09.61 61     5    1387  325M+  1536K  110M-  1361 1    sleeping *129[104]        0.00000 0.00000    501  224174+  2200
4666  screencaptur 1.4  00:00.33 3      2    97    8480K  752K   0B     597  597  sleeping *1[2]             0.39750 0.00000    501  3677+    101
1009  WhatsApp     0.9  28:21.60 36     5    2249  499M   2448K  106M   1009 1    sleeping *37447[3]        0.05223 0.17750    501  2387020+ 2858
598   Finder       0.6  02:04.29 10     4    1273+ 222M   3120K  47M    598  1    sleeping *0[2087]         0.08726 0.00000    501  380886+  520
1631  Terminal     0.6  00:26.96 12     5    445   157M+  32M    9152K  1631 1    sleeping *0[1794]         0.03138 0.00000    501  305252+  393
1052  Arc          0.4  11:22.12 67     4    1250  613M   5024K  85M    1052 1    sleeping *0[1205]         0.00122 0.00000    501  703913   3607
1062  Browser Help 0.3  03:26.97 38     1    218   50M    0B     15M    1052 1052 sleeping *295[556]        0.00000 0.00000    501  254932   801
413   mDNSResponde 0.1  00:22.69 6      3    147   6800K  0B     752K   413  1    sleeping *0[1]            0.00000 0.00000    65   10026    145
572   mysqld       0.1  00:44.99 37     0    59    487M   0B     483M   572  1    sleeping *0[1]            0.00000 0.00000    74   33277    155
328   fseventsd    0.1  00:58.22 12     1    163+  4272K+ 0B     1088K  328  1    sleeping *0[1]            0.06764 0.00000    0    79563+   93
686   routined     0.1  00:03.84 3      2    234+  16M+   352K   5824K  686  1    sleeping  0[97]           0.00000 0.00000    501  10459+   233
553   com.apple.Dr 0.1  03:33.51 7      5    1458  32M    0B     2768K  553  1    sleeping  0[1]            0.00000 0.00000    270  350740   65
4662  screencaptur 0.1  00:00.19 3      1    183   12M    0B     0B     4662 1    sleeping *0[175+]         0.00000 0.05236    501  3763     206
393   corebrightne 0.1  01:08.64 2      1    144   4832K  0B     1264K  393  1    sleeping *0[1]            0.12034 0.00000    0    92401+   140
3663  Browser Help 0.1  02:01.33 25     1    360   250M   0B     20M    1052 1052 sleeping  0[13158]        0.00000 0.00000    501  649613   825
1144  Microsoft Wo 0.1  01:18.05 21     7    588   272M   128K   120M   1144 1    sleeping  0[2291]         0.00000 0.00000    501  124906   918
338   powerd       0.1  00:11.41 3      2    157   5392K  0B     672K   338  1    sleeping *0[1]            0.00000 0.05844    0    57188+   125
3741  Browser Help 0.1  00:34.44 24     1    223   203M   0B     17M    1052 1052 sleeping *423[340]        0.00000 0.00000    501  71632    773
3689  Browser Help 0.0  00:22.11 24     1    244   161M   0B     15M    1052 1052 sleeping *454[921]        0.00000 0.00000    501  53543    744
530   audioclocksy 0.0  00:12.88 3      2    51    2880K  0B     1088K  530  1    sleeping *0[1]            0.00000 0.00000    0    970      93
364   opendirector 0.0  00:29.29 6      5    2132+ 12M+   128K   5472K  364  1    sleeping *0[1]            0.00000 0.06764    0    119383+  130
```

21. **`chmod`**
    chmod changes file permissions, controlling who can read, write, or execute a file. It is essential for managing security, enabling scripts, and controlling access within the system.



```
shreyashmahatu@Shreyash OSlab % ls -l
total 16
-rw-r--r--  1 shreyashmahatu  staff  17 Dec 10 18:40 2.txt
-rw-r--r--  1 shreyashmahatu  staff  17 Dec 10 18:42 shreyash.txt
shreyashmahatu@Shreyash OSlab % chmod 744 shreyash.txt
shreyashmahatu@Shreyash OSlab % ls -l
total 16
-rw-r--r--  1 shreyashmahatu  staff  17 Dec 10 18:40 2.txt
-rwxr--r--  1 shreyashmahatu  staff  17 Dec 10 18:42 shreyash.txt
shreyashmahatu@Shreyash OSlab % 
```