Lab 1 : Learning Basic Linux Commands

OS Used : Manjaro Linux x86_64

1. **Command Name:** ls

   Syntax : *ls* [option] [directory]

   Usage : The *ls* command lists files and directories within the file system and shows detailed information about them

   Commands Used:

   - ls => It lists all the unhidden files and directories within current directory.
   - ls Lab1 => It lists all the files and directories of directory
   - ls -l Lab1 => It lists files in long listing format which includes:
     - The file type
     - The file permissions
     - Number of hard links to the file
     - File owner
     - File group
     - File Size
     - Date and Time
     - File name

   Output:

```
sulav@mesulav:$ ls
Lab1
sulav@mesulav:$ ls Lab1/
Lab1.md  OS_LabWorks_Part-1.pdf
sulav@mesulav:$ ls -l Lab1
total 176
-rw-r--r-- 1 tyzrex tyzrex      0 Apr 11 18:40 Lab1.md
-rw-r--r-- 1 tyzrex tyzrex 179914 Mar 27 10:54 OS_LabWorks_Part-1.pdf
sulav@mesulav:$
```

---

2. **Command Name** : cd

   Syntax: *cd* [directory]

   Usage: The *cd* command is used to change the working directory.

   Commands Used:

   - cd => The *cd* command takes us to the home directory of the current user
   - cd Documents/College-Repo-Fourth-Semester/Operating\ System/Practical-Stuff/ => It changes the working directory to the specified directory.
   - cd .. => Go up to the parent directory of the current directory

   Output:

```
sulav@mesulav:$ ls
OS_LabWorks_Part-1.pdf  Practical-Stuff  Theory-Stuff
sulav@mesulav:$ cd
sulav@mesulav:$ ls
 Desktop     Linux-Dot-Files   Pictures                        Videos
 Documents   Music            "PlayOnLinux's virtual drives"
 Downloads   OS-Settings       Public
 go          pictures          Templates
sulav@mesulav:$ cd Documents/College-Repo-Fourth-Semester/Operating\ System/Practical-Stuff/
sulav@mesulav:$ ls
Lab1
sulav@mesulav:$ cd ..
sulav@mesulav:$ ls
OS_LabWorks_Part-1.pdf  Practical-Stuff  Theory-Stuff
sulav@mesulav:$ _
```

---

3. **Command Name:** grep

   Syntax: grep [OPTIONS] PATTERN [FILE...]

   Usage: The grep command is used to search for lines in one or more files that match a given pattern.

   Commands Used:

   - grep apple file1: Searches for lines containing the word "apple" in the file fruits.txt.
   - grep hello . -r : Recursively searches for lines containing the word "hello" in all files under the directory .
   - grep -i "hello" helloworld.sh: Searches for lines containing the word "hello" in a case-insensitive manner in the file helloworld.sh

   Output:

```
sulav@mesulav:$ grep apple file1
apple
sulav@mesulav:$ grep hello . -r
./Lab1/Lab1backup.md:    * chmod +x  helloworld.sh => This command adds execut
able permission to the file.
./Lab1/Lab1backup.md:    * chmod -x helloworld.sh => This command removes exec
utable permission of the file.
./Lab1/Lab1backup.md:    * chmod  +rw helloworld.sh => This command adds read
write permission to the file.
./Lab1/Lab1backup.md:    * cat helloworld.sh => It is used to display the cont
ent of the file in the terminal of the file helloworld.sh.
./Lab1/Lab1backup.md:    * cat helloworls.sh sheesh.sh > combined.sh => It is
used to concatenate multiple files into an output file.
./Lab1/Lab1backup.md:    * echo “hello world” => Prints hello world to the ter
minal.
./Lab1/Lab1backup.md:    * echo -n “hello world” => Prints hello world without
 trailing new line.

sulav@mesulav:$ grep hello -i helloworld.sh
echo "Hello World"
sulav@mesulav:$
```

---

4. **Command Name**: sudo

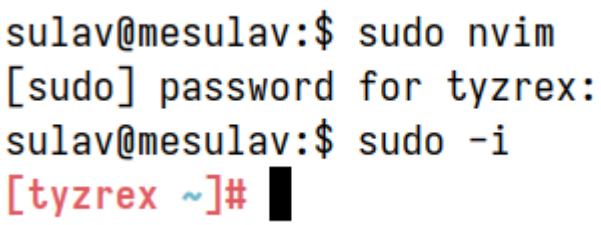
   Syntax: sudo [options] [command]

   Usage: The sudo command is used to execute a command with elevated privileges.

   Commands Used:

   - sudo nvim : Runs the specified command with elevated privileges.
   - sudo -i : Switches to the root user account.

   Output



---

5. **Command Name** : pwd

   Syntax: *pwd*

   Usage: This command prints the name of the current/working directory

   Commands Used:

   - pwd => Prints the name of the current/working directory

   Output:

```
sulav@mesulav:$ pwd
/home/tyzrex/Documents/College-Repo-Fourth-Semester/Operating System/Practical
-Stuff
sulav@mesulav:$ ~
```

---

6. **Command Name** : mv

   Syntax: *mv* [option] [source] [target]

   Usage: This command is used to move or rename files or directories

   Commands Used:

   - mv OS_LabWorks_Part-1.pdf Practical-Stuff/ => Moves the file into the existing Practical-Stuff directory.
   - mv OS_LabWorks_Part-1.pdf OS_Renamed.pdf => Renames the file into a new name.
   - mv file1 file2 file3 Lab1/ => Move multiple files to an existing directory, keeping the filenames unchanged.
   - mv -f file1 Lab1/ => This command doesn't prompt for confirmation before overwriting existing files
   - mv -i file1 Lab1/ => This command prompts for confirmation before overwriting existing files

   Output:

```
sulav@mesulav:$ mv OS_LabWorks_Part-1.pdf Practical-Stuff/
sulav@mesulav:$ cd Practical-Stuff/
sulav@mesulav:$ ls
Lab1  OS_LabWorks_Part-1.pdf
sulav@mesulav:$ mv OS_LabWorks_Part-1.pdf OS_Renamed.pdf
sulav@mesulav:$ ls
Lab1  OS_Renamed.pdf
sulav@mesulav:$ touch file1 file2 file3
sulav@mesulav:$ ls
file1  file2  file3  Lab1  OS_Renamed.pdf
sulav@mesulav:$ mv file1 file2 file3 Lab1/
sulav@mesulav:$ cd Lab1/
sulav@mesulav:$ ls
file1  file2  file3  Lab1.md  Lab1.pdf  OS_LabWorks_Part-1.pdf
sulav@mesulav:$ cd ..
sulav@mesulav:$ ls
Lab1  OS_Renamed.pdf
sulav@mesulav:$ touch file1
sulav@mesulav:$ mv -f  file1 Lab1/
```

```
sulav@mesulav:$ touch file1
sulav@mesulav:$ mv -i file1 Lab1/
mv: overwrite 'Lab1/file1'? y
sulav@mesulav:$
```

---

7. **Command Name** : cp

   Syntax: *cp* [option] [source] [target]

   Usage: This command is used to copy files and directories from a source to a target destination.

   Commands Used:

   - cp OS_Renamed.pdf Lab1/ => This copies the file to the given destination.
   - cp -r Lab1/ copiedLab1/=> This command recursively copy a directory's contents to another location (if the destination exists, the directory is copied inside it).
   - cp -i *.pdf multipleCopy/ => Copy all pdf files to a destination in user interactive mode (prompts before overwriting)

   Output:

```
sulav@mesulav:$ ls
Lab1  OS_Renamed.pdf
sulav@mesulav:$ cp OS_Renamed.pdf Lab1/
sulav@mesulav:$ cd Lab1/
sulav@mesulav:$ ls
file1  file2  file3  Lab1.md  Lab1.pdf  OS_LabWorks_Part-1.pdf  OS_Renamed.pdf
sulav@mesulav:$ cd ..
sulav@mesulav:$ mkdir copiedLab1
sulav@mesulav:$ cp -r Lab1/ copiedLab1/
sulav@mesulav:$ cd copiedLab1/
sulav@mesulav:$ ls
Lab1
sulav@mesulav:$ cd Lab1/
sulav@mesulav:$ ls
file1  file2  file3  Lab1.md  Lab1.pdf  OS_LabWorks_Part-1.pdf  OS_Renamed.pdf


sulav@mesulav:$ cp -i *.pdf multipleCopy/
sulav@mesulav:$ cd multipleCopy/
sulav@mesulav:$ ls
Lab1.pdf  OS_LabWorks_Part-1.pdf  OS_Renamed.pdf
sulav@mesulav:$
```

---

8. **Command Name:** rm

   Syntax: rm [OPTION]... [FILE]...

   Usage: The rm command is used to remove or delete files or directories in a Unix or Linux operating system.

   Commands Used:

   - rm file1: Deletes a file named file1.
   - rm -r directory: Deletes a directory named directory and its contents recursively.
   - rm -f file2: Forces the deletion of a file named file2 without prompting for confirmation.

   Output:

```
sulav@mesulav:$ cd testdir/
sulav@mesulav:$ ls
sulav@mesulav:$ touch file1 file2
sulav@mesulav:$ mkdir test
sulav@mesulav:$ tree .
.
├── file1
├── file2
└── test

2 directories, 2 files
sulav@mesulav:$ rm file1
sulav@mesulav:$ rm -r test/
sulav@mesulav:$ rm file2 -f
sulav@mesulav:$ tree .
.

0 directories, 0 files
sulav@mesulav:$
```

---

9. **Command Name**: mkdir

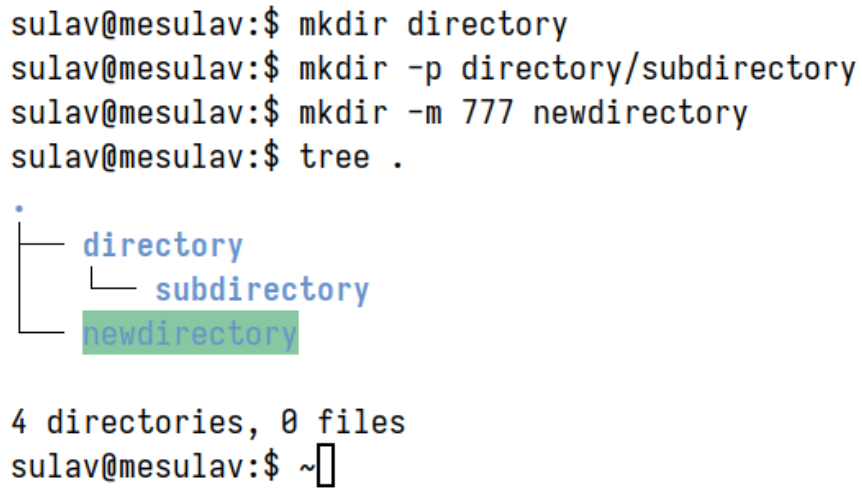
   Syntax: mkdir [option] directory

   Usage: The *mkdir* command is used to create new directories.

   Commands Used:

   - mkdir directory - Creates a new directory named "directory".
   - mkdir -p directory/subdirectory - Creates a new directory named "subdirectory" inside the "directory". If "directory" does not exist, it creates it as well.
   - mkdir -m 777 newdirectory - Creates a new directory named "newdirectory" with permissions set to 777.

   Output:



---

10. **Command Name**: rmdir

    Syntax: rmdir [options] directory

    Usage: The rmdir command is used to remove empty directories.

    Commands Used:

    1. rmdir trydir: This command will remove an empty directory named directory_name in the current working directory.
    2. rmdir -v newdir: This command will remove the directory named directory_name and print a message for each directory that is removed.
    3. rmdir -p directory/subdirectory: This command will remove both the parent and child directories together.

    Output:

```
sulav@mesulav:$ rmdir trydir/
sulav@mesulav:$ rmdir -v newdir/
rmdir: removing directory, 'newdir/'
sulav@mesulav:$ rmdir -p directory/subdirectory/
sulav@mesulav:$ tree .
.

0 directories, 0 files
sulav@mesulav:$
```

---

11. **Command Name** : chmod

    Syntax: *chmod* [option] [model] [file]

    Usage: This command is used to change the access permissions of a file or a directory.

    Commands Used:

    - chmod +x helloworld.sh => This command adds executable permission to the file.
    - chmod -x helloworld.sh => This command removes executable permission of the file.
    - chmod +rw helloworld.sh => This command adds read write permission to the file.

    Output:

```
sulav@mesulav:$ ls
copiedLab1  helloworld.sh  Lab1  OS_Renamed.pdf
sulav@mesulav:$ ./helloworld.sh
bash: ./helloworld.sh: Permission denied
sulav@mesulav:$ chmod +x helloworld.sh
sulav@mesulav:$ ./helloworld.sh
Hello World
sulav@mesulav:$ chmod -x helloworld.sh
sulav@mesulav:$ ./helloworld.sh
bash: ./helloworld.sh: Permission denied
sulav@mesulav:$ chmod +rw helloworld.sh
sulav@mesulav:$
```

---

12. **Command Name** : cat

    Syntax: *cat*

    Usage: This command is used to print and contatenate files to standard output.

    Commands Used:

    - cat helloworld.sh => It is used to display the content of the file in the terminal of the file helloworld.sh.
    - cat helloworls.sh sheesh.sh > combined.sh => It is used to concatenate multiple files into an output file.
    - cat -n combined.sh => It prints the number of all output lines.

    Output:

```
sulav@mesulav:$ cat helloworld.sh
echo "Hello World"
sulav@mesulav:$ cat sheesh.sh
echo "sheesh"
sulav@mesulav:$ ls
copiedLab1  helloworld.sh  Lab1  OS_Renamed.pdf  sheesh.sh
sulav@mesulav:$ cat helloworld.sh sheesh.sh > combined.sh
sulav@mesulav:$ cat combined.sh
echo "Hello World"
echo "sheesh"
sulav@mesulav:$ cat -n combined.sh
     1  echo "Hello World"
     2  echo "sheesh"
sulav@mesulav:$
```

---

13. **Command Name** : chown

    Syntax: *chown* [option] [owner] [group] [file]

    Usage: This command is used to change the user and group ownership of files and directories.

    Commands Used:

    - chown sulav index.html => Changes the owner of the file to the specified username.
    - chown :sulav helloworld.sh=> Changes the group of the file to the specified group name.
    - chown sulav:sulav file2 => Changes both the owner and group of the file to the specified user and group names.

    Output:

```
sulav@mesulav:$ sudo chown sulav index.html
sulav@mesulav:$ sudo chown :sulav helloworld.sh
sulav@mesulav:$ sudo chown sulav:sulav file2
sulav@mesulav:$ la -l
bash: la: command not found
sulav@mesulav:$ ls -l
total 224
-rw-r--r-- 1 tyzrex tyzrex     33 Apr 17 09:30 combinedrename.sh
drwxr-xr-x 3 tyzrex tyzrex   4096 Apr 17 08:27 copiedLab1
-rw-r--r-- 1 sulav  sulav       0 Apr 23 22:13 file2
-rw-r--r-- 1 tyzrex tyzrex      0 Jan  1  2022 filerenamed.txt
-rwxr-xr-x 1 sulav  sulav      92 Apr 23 08:25 helloworld.sh
-rw-r--r-- 1 sulav  tyzrex  16843 Apr 23 21:06 index.html
drwxr-xr-x 2 tyzrex tyzrex   4096 Apr 23 22:19 Lab1
-rw-r--r-- 1 tyzrex tyzrex 179914 Mar 27 10:53 OS_lab.pdf
-rw-r--r-- 1 tyzrex tyzrex     14 Apr 17 09:28 sheesh.sh
drwxr-xr-x 2 tyzrex tyzrex   4096 Apr 23 22:18 testdir
drwxr-xr-x 5 tyzrex tyzrex   4096 Apr 23 21:05 ui8-fleet-html.herokuapp.com
sulav@mesulav:$
```

---

14. **Command Name** : echo

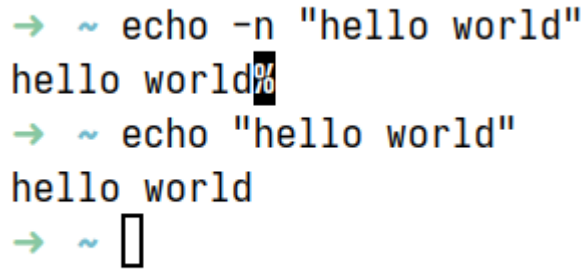
    Syntax: *echo*

    Usage: This command is used to print given arguements.

    Commands Used:

    - echo “hello world” => Prints hello world to the terminal.
    - echo -n “hello world” => Prints hello world without trailing new line.

    Output:



---

15. **Command Name** : wc

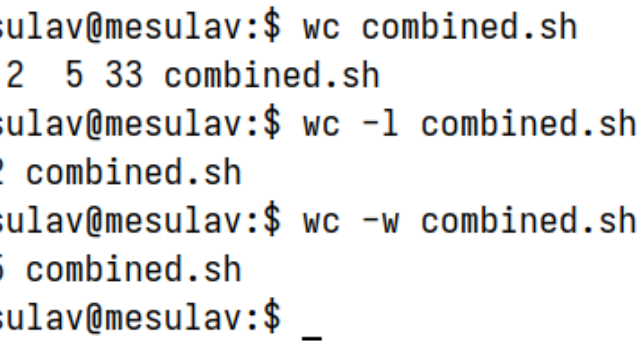
    Syntax: *wc* [options] [file(s)]

    Usage: The *wc* command is primarily used for counting the number of lines, words, and characters in one or more files.

    Commands Used:

    - wc combined.sh => Displays the number of lines, words, and characters in the file
    - wc -l combined.sh => Displays the number of lines in the file
    - wc -w combined.sh => Displays the number of words in the file

    Output:



---

16. **Command Name** : man

    Syntax: *man* [options] [command]

    Usage: The *man* command is used to display the manual page for the specified command.

    Commands Used:

    - man ls => Displays the manual page for ls command.
    - man -k grep => Searches the manual pages for the specified search term.
    - man -f chmod => Displays a short description of the specified command.

    Output:

```
LS(1)                          User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information about the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
       fied.

       Mandatory  arguments  to long options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file
```

```
sulav@mesulav:$ man ls
sulav@mesulav:$ man -k grep
git-bugreport (1)    - Collect information for user to file a bug report
git-grep (1)         - Print lines matching a pattern
grep (1)             - print lines that match patterns
grep (1p)            - search a file for a pattern
lzegrep (1)          - search compressed files for a regular expression
lzfgrep (1)          - search compressed files for a regular expression
lzgrep (1)           - search compressed files for a regular expression
msggrep (1)          - pattern matching on message catalog
pcre2grep (1)        - a grep with Perl-compatible regular expressions.
pcregrep (1)         - a grep with Perl-compatible regular expressions.
pgrep (1)            - look up, signal, or wait for processes based on name...
plugreport (1)       - a program to read all MPR/PCR registers from all dev...
ptargrep (1perl)     - Apply pattern matching to the contents of files in a...
Tcl_InvalidateStringRep (3) - manipulate Tcl values
xzegrep (1)          - search compressed files for a regular expression
xzfgrep (1)          - search compressed files for a regular expression
xzgrep (1)           - search compressed files for a regular expression
zgrep (1)            - search possibly compressed files for a regular expre...
zipgrep (1)          - search files in a ZIP archive for lines matching a p...
zstdgrep (1)         - print lines matching a pattern in zstandard-compress...
sulav@mesulav:$
```

```
sulav@mesulav:$ man -f chmod
chmod (1)            - change file mode bits
chmod (1p)           - change the file modes
chmod (2)            - change permissions of a file
chmod (3p)           - change mode of a file
sulav@mesulav:$ _
```

---

17. **Command Name** : history

    Syntax: *history* [options]

    Usage: This command is used to view the command-line history.

    Commands Used:

    - history => Displays a list of previously executed commands.
    - history 10 => Displays the last 10 executed commands.
    - history -c => Clears the command history.

    Output:

```
  363  clear
  364  wc combined.sh
  365  wc -l combined.sh
  366  wc -w combined.sh
  367  tldr wc
  368  clea
  369  clear
  370  man ls
  371  clear
  372  man ls
  373  man -k grep
  374  clear
  375  man -f chmod
  376  tldr man
  377  man 7 chown
  378  clear
  379  tldr history
  380  clear
  381  history
sulav@mesulav:$ history 10
  373  man -k grep
  374  clear
  375  man -f chmod
  376  tldr man
  377  man 7 chown
  378  clear
  379  tldr history
  380  clear
  381  history
  382  history 10
sulav@mesulav:$ history -c
sulav@mesulav:$ history
    1  history
sulav@mesulav:$
```

---

18. **Command Name** : clear

    Syntax: *clear*

    Usage: The *clear* command is used to clear the screen of the terminal

    Commands Used:

    - clear => Clear the screen of the terminal window.

    Output:

```
sulav@mesulav:$ history
    1  history
    2  tldr clear
    3  man clear
    4  clear -h
    5  clear
    6  clear --h
    7  clear
    8  clear -h
    9  clear
   10  clear -x
   11  clear
   12  clear -x
   13  history
   14  clear -x
   15  history
   16  clear
   17  history
sulav@mesulav:$ clear
```

```
sulav@mesulav:$
```

---

19. **Command Name** : touch

    Syntax: *touch* [option] [file]

    Usage: The *clear* command is used to clear the screen of the terminal

    Commands Used:

    - touch file.txt => Creates an empty file named `file.txt`.
    - touch -t 202201010101 file.txt => Updates the modification time of `file.txt` to January 1, 2022 at 1:01 AM.

    Output:

```
sulav@mesulav:$ touch file.txt
sulav@mesulav:$ touch -t 202201010101 file.txt
sulav@mesulav:$ ls -l
total 196
-rw-r--r-- 1 tyzrex tyzrex     33 Apr 17 09:30 combined.sh
drwxr-xr-x 3 tyzrex tyzrex   4096 Apr 17 08:27 copiedLab1
-rw-r--r-- 1 tyzrex tyzrex      0 Jan  1  2022 file.txt
-rw-r--r-- 1 tyzrex tyzrex     19 Apr 17 09:14 helloworld.sh
drwxr-xr-x 2 tyzrex tyzrex   4096 Apr 17 19:13 Lab1
-rw-r--r-- 1 tyzrex tyzrex 179914 Mar 27 10:53 OS_Renamed.pdf
-rw-r--r-- 1 tyzrex tyzrex     14 Apr 17 09:28 sheesh.sh
sulav@mesulav:$
```

---

20. **Command Name:** locate

    Syntax: locate [OPTION]... PATTERN...

    Usage: The locate command is used to find files and directories on a system by searching a prebuilt database of file names.

    Commands Used:

    - locate hello.php: This command will search the database for any files or directories containing the string "myfile" in their name.
    - locate -i Postdetails.jsx: This command will perform a case-insensitive search for "myfile".
    - locate -r AuthContext: This command will perform a regular expression search for "myfile".

    Output:

```
sulav@mesulav:$ locate hello.php
/home/tyzrex/.local/state/nvim/undo/%home%tyzrex%Documents%hello.php
/home/tyzrex/Documents/hello.php
/home/tyzrex/Documents/Php-Basics/1hello.php
sulav@mesulav:$ locate -i Postdetails.jsx
/home/tyzrex/Documents/GitHub Projects/RideBnb-Semester-Project/client/src/Com
ponents/Pages/PostDetails.jsx
/home/tyzrex/Documents/RideBnb-Semester-Project/client/src/Components/Pages/Po
stDetails.jsx
sulav@mesulav:$ locate -i AuthContext
/home/tyzrex/Documents/Blog-Site-React/client/src/Components/Context/Authconte
xt.jsx
/home/tyzrex/Documents/GitHub Projects/Blog/client/src/Components/Context/Auth
context.jsx
/home/tyzrex/Documents/GitHub Projects/Blog-Site-React/client/src/Components/C
ontext/Authcontext.jsx
/home/tyzrex/Documents/GitHub Projects/RideBnb-Semester-Project/client/node_mo
dules/react-auth-kit/dist/AuthContext.d.ts
/home/tyzrex/Documents/GitHub Projects/RideBnb-Semester-Project/client/node_mo
dules/react-auth-kit/dist/AuthContext.js
/home/tyzrex/Documents/GitHub Projects/RideBnb-Semester-Project/client/node_mo
dules/react-auth-kit/dist/AuthContext.js.map
/home/tyzrex/Documents/GitHub Projects/RideBnb-Semester-Project/client/src/Con
text/AuthContext.jsx
/home/tyzrex/Documents/RideBnb-Semester-Project/client/src/Context/AuthContext
.jsx
/home/tyzrex/Documents/Semester-Projects/RideBnb-Project/client/src/context/Au
thContext.tsx
sulav@mesulav:$
```

---

21. **Command Name:** netstat

    Syntax: netstat [options]

    Usage: The netstat command is used to display various information about the network connections on a system, including both incoming and outgoing connections.

    Commands Used:

    - netstat -a: Shows all connections and listening ports.
    - netstat -n: Shows the numerical addresses instead of host and port names.
    - netstat -t: Shows only the TCP connections.
    - netstat -u: Shows only the UDP connections.

    Output:

```
tcp        0      0 tyzrex:56398            edge-dgw-shv-01-b:https ESTABLISHE
D
tcp        0      0 tyzrex:36752            ec2-35-174-127-31:https ESTABLISHE
D
tcp        0      0 tyzrex:51812            server-52-222-144:https ESTABLISHE
D
tcp        0      0 tyzrex:54360            lb-140-82-113-25-:https ESTABLISHE
D
tcp        0      0 tyzrex:37966            edge-star-shv-01-:https ESTABLISHE
D
sulav@mesulav:$ netstat -u
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State

udp        0      0 tyzrex:bootpc           _gateway:bootps         ESTABLISHE
D
sulav@mesulav:$ netstat -l
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State

tcp        0      0 localhost:mshvlm        0.0.0.0:*               LISTEN

tcp        0      0 localhost:ipps          0.0.0.0:*               LISTEN

tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
```

---

22. **Command Name** : df

    Syntax: *df* [options] [file]

    Usage: The *df* command gives an overview of the filesystem disk space usage.

    Commands Used:

    - df -h => Display all filesystems and their disk usage in human-readble form
    - df -i => Display statistics on the number of free inodes.
    - df -x tmpfs => Display filesystem but exclude the specified type tmpfs.

    Output:

```
sulav@mesulav:$ df -h
Filesystem      Size  Used Avail Use% Mounted on
dev             3.6G     0  3.6G   0% /dev
run             3.6G  1.7M  3.6G   1% /run
/dev/nvme0n1p8  335G  112G  206G  36% /
tmpfs           3.6G   34M  3.6G   1% /dev/shm
tmpfs           3.6G  4.4M  3.6G   1% /tmp
/dev/nvme0n1p1   96M   30M   67M  32% /boot/efi
tmpfs           737M   92K  737M   1% /run/user/1000
sulav@mesulav:$ df -i
Filesystem       Inodes   IUsed    IFree IUse% Mounted on
dev              939546     624   938922    1% /dev
run              942348    1077   941271    1% /run
/dev/nvme0n1p8 22339584 2868597 19470987   13% /
tmpfs            942348     111   942237    1% /dev/shm
tmpfs            942348      46   942302    1% /tmp
/dev/nvme0n1p1        0       0        0     - /boot/efi
tmpfs            188469     104   188365    1% /run/user/1000
sulav@mesulav:$ df -x tmpfs
Filesystem     1K-blocks      Used Available Use% Mounted on
dev              3758184         0   3758184   0% /dev
/dev/nvme0n1p8 350600984 117369720 215348436  36% /
/dev/nvme0n1p1     98304     30516     67788  32% /boot/efi
sulav@mesulav:$ ~
```

---

23. **Command Name** : du

    Syntax: *du* [options] [file]

    Usage: The *df* command estimates and summarize file and directory space usage.

    Commands Used:

    - du -b copiedLab1 => List the sizes of a directory and any subdirectories in the given unit(bytes in this case).
    - du -h => Display the disk usage of the current directory in a human-readable format
    - du -s => Display only the total disk usage of a specific file or directory

    Output:

```
sulav@mesulav:$ du -b copiedLab1/
576180  copiedLab1/Lab1/multipleCopy
1157037 copiedLab1/Lab1
1157037 copiedLab1/
sulav@mesulav:$ du -h
568K    ./copiedLab1/Lab1/multipleCopy
1.2M    ./copiedLab1/Lab1
1.2M    ./copiedLab1
580K    ./Lab1
1.9M    .
sulav@mesulav:$ du -s copiedLab1/
1148    copiedLab1/
sulav@mesulav:$
```

---

24. **Command Name** : uname

    Syntax: *uname* [options]

    Usage: The *uname* command prints information about the machine and operating system it is run on.

    Commands Used:

    - uname –all => This command prints all the information about the machine.
    - uname –nodename => This command prints the current network host name.
    - uname –kernel-release => This command prints the current kernel release.

    Output:

```
sulav@mesulav:$ uname --all
Linux tyzrex 6.1.22-1-MANJARO #1 SMP PREEMPT_DYNAMIC Fri Mar 31 07:11:32 UTC 2
023 x86_64 GNU/Linux
sulav@mesulav:$ uname --nodename
tyzrex
sulav@mesulav:$ uname --kernel-release
6.1.22-1-MANJARO
sulav@mesulav:$ _
```

---

25. **Command Name** : passwd

    Syntax: *passwd* [options] [username]

    Usage: The *passwd* command is used to set or change the password of a user.

    Commands Used:

    - passwd root : Change the password of the root user.

    Output:

```
sulav@mesulav:$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
sulav@mesulav:$ _
```

---

26. **Command Name** : useradd

    Syntax: *useradd* [options] [name]

    Usage: The *useradd* command creates a new user or update default new user information.

Commands Used:

- sudo useradd testtyz => Adds a new user sulav.
- passwd -S => Gets the current status of the user.

Output:

```
sulav@mesulav:$ sudo useradd testtyz
[sudo] password for tyzrex:
sulav@mesulav:$ passwd testtyz
passwd: You may not view or modify password information for testtyz.
sulav@mesulav:$ sudo passwd testtyz
New password:
Retype new password:
passwd: password updated successfully
sulav@mesulav:$ su testtyz
Password:
tyzrex% passwd -S
testtyz P 2023-04-23 0 99999 7 -1
tyzrex% _
```

---

27. **Command Name** : userdel

    Syntax: *userdel* [options] [name]

    Usage: The *useradd* command is used to remove a user account or remove a user from a group.

    Commands Used:

    - sudo userdel testtyz => Deletes the user account named testtyz

    Output:

```
➜  ~ bash
sulav@mesulav:$ sudo userdel testtyz
[sudo] password for tyzrex:
sulav@mesulav:$ su testtyz
su: user testtyz does not exist or the user entry does not contain all the req
uired fields
sulav@mesulav:$ _
```

---

28. **Command Name** : apt-get

Syntax: *apt-get* [options] [packagename]

Usage: The *apt-get* command is used to manage packages in the Debian operating system.

Commands Used:

- sudo apt-get update => Update the list of available packages and versions
- sudo apt-get install kitty => Install a package with the given packagename.
- sudo apt-get remove kitty => Remove the package with the given package name.

Output:

```
sulav@mesulav: $apt-get update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1000 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [911 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [959 kB]
Get:9 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [23.2 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [28.6 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1301 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1153 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1054 kB]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [49.4 kB]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [25.6 kB]
Fetched 26.8 MB in 8s (3478 kB/s)
Reading package lists... Done
```

```
sulav@mesulav: $apt-get install kitty
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kitty is already the newest version (0.21.2-1ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
sulav@mesulav: $apt-get remove kitty
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  alsa-topology-conf alsa-ucm-conf dbus fontconfig-config fonts-dejavu-core javascript-common kitty-doc kitty-terminfo libapparmor1 libasound2 libasound2-data
  libbrotli1 libbsd0 libcanberra0 libdbus-1-3 libexpat1 libfontconfig1 libfreetype6 libglib2.0-0 libglib2.0-data libgraphite2-3 libharfbuzz0b libicu70
  libjs-jquery libjs-sphinxdoc libjs-underscore liblcms2-2 libltdl7 libmd0 libmpdec3 libogg0 libpng16-16 libpython3-stdlib libpython3.10 libpython3.10-minimal
  libpython3.10-stdlib libreadline8 libsqlite3-0 libtdb1 libvorbis0a libvorbisfile3 libwayland-client0 libx11-6 libx11-data libx11-xcb1 libxau6 libxcb-xkb1
  libxcb1 libxdmcp6 libxkbcommon-x11-0 libxkbcommon0 libxml2 media-types python3 python3-minimal python3.10 python3.10-minimal readline-common shared-mime-info
  sound-theme-freedesktop ucf xdg-user-dirs xkb-data
Use 'apt autoremove' to remove them.
The following packages will be REMOVED:
  kitty
0 upgraded, 0 newly installed, 1 to remove and 4 not upgraded.
After this operation, 7415 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 7428 files and directories currently installed.)
Removing kitty (0.21.2-1ubuntu0.22.04.1) ...
sulav@mesulav: $
```

---

29. **Command Name** : ping

    Syntax: *ping* [options] [destination]

    Usage: The *ping* command is used to send ICMP ECHO_REQUEST packets to network hosts

    Commands Used:

    - ping google.com => Ping the given hostname
    - ping -c 2 youtube.com => Ping the given hostname 2 times

    Output:

```
sulav@mesulav:$ ping google.com
PING google.com (142.250.193.206) 56(84) bytes of data.
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=1 ttl=116 time=56.8 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=2 ttl=116 time=33.5 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=3 ttl=116 time=40.6 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=4 ttl=116 time=75.6 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=5 ttl=116 time=36.6 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=6 ttl=116 time=34.3 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=7 ttl=116 time=37.7 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=8 ttl=116 time=34.8 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=9 ttl=116 time=42.4 ms
^C
--- google.com ping statistics ---
10 packets transmitted, 9 received, 10% packet loss, time 9001ms
rtt min/avg/max/mdev = 33.455/43.575/75.587/13.155 ms
sulav@mesulav:$ ping -c 2 youtube.com
PING youtube.com (142.250.193.206) 56(84) bytes of data.
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=1 ttl=116 time=110 ms
64 bytes from del11s17-in-f14.1e100.net (142.250.193.206): icmp_seq=2 ttl=116 time=35.4 ms

--- youtube.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 35.356/72.486/109.617/37.130 ms
sulav@mesulav:$ _
```

---

30. **Command Name** : find

    Syntax: find [where to start searching from] [expression determines what to find] [-options] [what to find]

    Usage: The *find* command in UNIX is a command line utility for walking a file hierarchy. It can be used to find files and directories and perform subsequent operations on them.

    Commands Used:

    - find ./Lab1/ -name file1 => Searches for the files with the given name file1.
    - find ./Lab1/ -name “*.pdf” => Searches for the file with the given extension name.

    Output:

```
sulav@mesulav:$ tree Lab1/
Lab1/
├── file1
├── file2
├── file3
├── Lab1.md
├── Lab1.pdf
├── OS_LabWorks_Part-1.pdf
└── OS_Renamed.pdf

1 directory, 7 files
sulav@mesulav:$ find ./Lab1/ -name file1
./Lab1/file1
sulav@mesulav:$ find ./Lab1/ -name "*.pdf"
./Lab1/Lab1.pdf
./Lab1/OS_Renamed.pdf
./Lab1/OS_LabWorks_Part-1.pdf
sulav@mesulav:$ _
```

---

31. **Command Name** : head

    Syntax: head [option] [file]

    Usage: The *head* command outputs the first part of files.

    Commands Used:

    - head –lines 2 helloworld.sh => Output the first two lines of the file helloworld.sh
    - head –lines -1 helloworld.sh => Output everything but the last 1 line of the file helloworld.sh
    - head –bytes 2 helloworld.sh => Output first 2 bytes of the file helloworld.sh

    Output:

```
sulav@mesulav:$ head --lines 2  helloworld.sh
echo "Hello World"
echo "Today is a beautiful day"
sulav@mesulav:$ head --lines -1 helloworld.sh
echo "Hello World"
echo "Today is a beautiful day"
echo "I want to eat pizza"
sulav@mesulav:$ head --bytes 2 helloworld.sh
ecsulav@mesulav:$
```

---

32. **Command Name** : rename

    Syntax: rename [option] [expression] [replacement] [file]

    Usage: The *head* command outputs the first part of files.

    Commands Used:

    - rename filerenamed filerenamedonceagain * => rename the file filerenamed to filerenamedonceagain
    - rename -vn file2renamed toberenamed * => show what file is going to be renamed to given renamed filename

    Output:

```
sulav@mesulav:$ ls
file2renamed  file3  filerenamed  hello.txt
sulav@mesulav:$ rename filerenamed filerenamedonceagain *
sulav@mesulav:$ ls
file2renamed  file3  filerenamedonceagain  hello.txt
sulav@mesulav:$ rename -vn file2renamed toberenamed *
`file2renamed' -> `toberenamed'
sulav@mesulav:$
```

---

33. **Command Name** : tail

    Syntax: tail [option] [file]

    Usage: The *tail* command is used to display the last part of the file.

    Commands Used:

    - tail –lines 1 hello.txt => displays the last line of the file hello.txt
    - tail –lines +2 hello.txt => displays all the lines from the line 2 in the file
    - tail –bytes 10 hello.txt => displays the last 10 bytes from the file

    Output:

```
sulav@mesulav:$ tail --lines 1 hello.txt
this is the last line
sulav@mesulav:$ tail --lines +2 hello.txt
hi there are lines
hi there are three lines
this is the last line
sulav@mesulav:$ tail --bytes 10 hello.txt
ast line
sulav@mesulav:$
```

---

34. **Command Name** : tac

    Syntax: tac [option] [file]

    Usage: The *tac* command is used to display and concatenate files with lines in reversed order.

    Commands Used:

    - tac combinedrename.sh => Concatenates specific file in reverse order
    - tac combinedrename.sh –separator sheesh => It separates the content of the file “combinedrename.sh” from the string “sheesh” to rest of the content.

    Output:

```
sulav@mesulav:$ ls
combinedrename.sh  filerenamed.txt  Lab1        sheesh.sh
copiedLab1         helloworld.sh    OS_lab.pdf
sulav@mesulav:$ tac combinedrename.sh
echo "sheesh"
echo "Hello World"
sulav@mesulav:$ tac combinedrename.sh --separator sheesh
"
echo "Hello World"
echo "sheeshsulav@mesulav:$
```

---

35. **Command Name**: comm

    Syntax: *comm* [OPTION]... FILE1 FILE2

    Usage: The *comm* command compares two sorted files line by line and prints the lines that are unique or common between them.

    Commands Used:

    - comm -23 FILE1 FILE2: Prints lines that are unique to FILE1.
    - comm -13 FILE1 FILE2: Prints lines that are unique to FILE2.
    - comm -12 FILE1 FILE2: Prints lines that are common to both FILE1 and FILE2.

    Output:

```
sulav@mesulav:$ comm -23 file1 file2
apple
cherry
orange
sulav@mesulav:$ comm -13 file1 file2
lemon
peach
watermelon
sulav@mesulav:$ comm -12 file1 file2
banana
grape
sulav@mesulav:$
```

---

36. **Command Name**: cut

    Syntax: *cut* [OPTION]... [FILE]...

    Usage: The *cut* command is used to extract sections from each line of a file or from multiple files. It can be used to extract columns from a text file or to split a line into fields based on a delimiter.

    Commands Used:

    1. cut -d ',' -f 1 FILE: Extracts the first column from a comma-separated file.
    2. cut -c 1-5 FILE: Extracts the first five characters from each line of a file.
    3. cut -d ' ' -f 1,3 FILE: Extracts the first and third columns from a space-separated file.

    Output:

```
sulav@mesulav:$ touch newfile
sulav@mesulav:$ nvim newfile
sulav@mesulav:$ cut -d ',' -f 1 newfile
Name
John
Mary
Mike

sulav@mesulav:$ cut -c 1-5 newfile
Name,
John,
Mary,
Mike,

sulav@mesulav:$ cut -d ' ' -f 1,3 newfile
Name, Gender
John, Male
Mary, Female
Mike, Male

sulav@mesulav:$
```

---

37. **Command Name**: sort

    Syntax: *sort* [option] [file]

    Usage: The *sort* command is used to sort the contents of a file.

    Commands Used:

    - sort testfile: Sorts the contents of "testfile" alphabetically.
    - sort -n testfile: Sorts the contents of "testfile" numerically.
    - sort -r testfile: Sorts the contents of "testfile" in reverse order.

    Output:

```
sulav@mesulav:$ sort testfile

John, 25, Male
Mary, 30, Female
Mike, 40, Male
Name, Age, Gender
sulav@mesulav:$ sort -n testfile

John, 25, Male
Mary, 30, Female
Mike, 40, Male
Name, Age, Gender
sulav@mesulav:$ sort -r testfile
Name, Age, Gender
Mike, 40, Male
Mary, 30, Female
John, 25, Male
```

---

38. **Command Name**: date

    Syntax: *date* [option]

    Usage: The *date* command is used to display the current date and time.

    Commands Used:

    - date: Displays the current date and time in the default format.
    - date +"%Y-%m-%d": Displays the current date in the format "YYYY-MM-DD".
    - date +"%H:%M:%S": Displays the current time in the format "HH:MM:SS".``

    Output:

```
sulav@mesulav:$ date
Sun Apr 23 08:53:34 PM +0545 2023
sulav@mesulav:$ date +"%Y-%m-%d"
2023-04-23
sulav@mesulav:$ date +"%H:%M:%S"
20:53:41
sulav@mesulav:$
```

---

39. **Command Name**: cal

    Syntax: cal [option] [month] [year]

    Usage: The cal command is used to display a calendar for the specified month and year.

    Commands Used:

    - cal: Displays the calendar for the current month and year.
    - cal [month] [year]: Displays the calendar for the specified month and year. For example,cal 12 2023` displays the calendar for December 2023.
    - cal -3: Displays the current month, the previous month, and the next month.

    Output:

```
sulav@mesulav:$ cal
     April 2023
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
sulav@mesulav:$ cal march 2022
     March 2022
Su Mo Tu We Th Fr Sa
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

sulav@mesulav:$ cal -3
     March 2023            April 2023            May 2023
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4                     1      1  2  3  4  5  6
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   7  8  9 10 11 12 13
12 13 14 15 16 17 18   9 10 11 12 13 14 15  14 15 16 17 18 19 20
19 20 21 22 23 24 25  16 17 18 19 20 21 22  21 22 23 24 25 26 27
26 27 28 29 30 31     23 24 25 26 27 28 29  28 29 30 31
                      30
sulav@mesulav:$
```

---

40. **Command Name**: time

    Syntax: *time* [command]

    Usage: The *time* command is used to measure the execution time of a command.

    Commands Used:

    - time ls: Measures the execution time of the ls command and displays the output.
    - time sleep 5: Measures the execution time of the sleep command for 5 seconds and displays the output.
    - time ./my_program: Measures the execution time of the my_program executable file and displays the output.

    Output:

```
sulav@mesulav:$ time ls
combinedrename.sh  file1  filerenamed.txt  Lab1        sheesh.sh
copiedLab1         file2  helloworld.sh    OS_lab.pdf  testfile

real    0m0.003s
user    0m0.002s
sys     0m0.001s
sulav@mesulav:$ time sleep 5

real    0m5.002s
user    0m0.002s
sys     0m0.001s
sulav@mesulav:$ time ./helloworld.sh
Hello World
Today is a beautiful day
I want to eat pizza
Coffee

real    0m0.002s
user    0m0.001s
sys     0m0.000s
sulav@mesulav:$
```

---

41. **Command Name**: host

    Syntax: host [option] [hostname]

    Usage: The host command is used to perform DNS lookups and display information about a hostname.

    Commands Used:

    - host google.com: Performs a DNS lookup for the `google.com` hostname and displays the IP address(es) associated with it.
    - host -t mx yahoo.com: Performs a DNS lookup for the `yahoo.com` hostname and displays the MX records associated with it.
    - host -a 192.168.1.1: Performs a reverse DNS lookup for the IP address `192.168.1.1` and displays the hostname associated with it.

    Output:

```
sulav@mesulav:$ host google.com
google.com has address 142.250.194.110
google.com has IPv6 address 2404:6800:4002:821::200e
google.com mail is handled by 10 smtp.google.com.
sulav@mesulav:$ host -t mx yahoo.com
yahoo.com mail is handled by 1 mta6.am0.yahoodns.net.
yahoo.com mail is handled by 1 mta7.am0.yahoodns.net.
yahoo.com mail is handled by 1 mta5.am0.yahoodns.net.
sulav@mesulav:$ host -a 192.168.1.1
Trying "1.1.168.192.in-addr.arpa"
Host 1.1.168.192.in-addr.arpa. not found: 3(NXDOMAIN)
Received 42 bytes from 1.1.1.1#53 in 6 ms
sulav@mesulav:$
```

---

42. **Command Name**: wget

    Syntax: wget [option] [URL]

    Usage: The wget command is used to download files from the web.

    Commands Used:

    - wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.3.tar.xz: Downloads the tar file from the kernel.org website and saves it to the current directory.
    - wget www.google.com: Download the contents of the given URL

    Output:

```
sulav@mesulav:$ wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.3
.tar.xz
--2023-04-23 21:06:03--  https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.
12.3.tar.xz
Loaded CA certificate '/etc/ssl/certs/ca-certificates.crt'
Resolving cdn.kernel.org (cdn.kernel.org)... 151.101.1.176, 151.101.65.176, 15
1.101.129.176, ...
Connecting to cdn.kernel.org (cdn.kernel.org)|151.101.1.176|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 118122820 (113M) [application/x-xz]
Saving to: ‘linux-5.12.3.tar.xz.1’

linux-5.12.3.tar.xz 100%[==================>] 112.65M  16.0MB/s    in 6.8s

2023-04-23 21:06:10 (16.7 MB/s) - ‘linux-5.12.3.tar.xz.1’ saved [118122820/118
122820]

sulav@mesulav:$ wget www.google.com
--2023-04-23 21:06:15--  http://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.207.228, 2404:6800:4002:8
2f::2004
Connecting to www.google.com (www.google.com)|142.250.207.228|:80... connected
.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘index.html’

index.html              [ <=>               ]  16.45K  --.-KB/s    in 0.02s

2023-04-23 21:06:16 (709 KB/s) - ‘index.html’ saved [16843]

sulav@mesulav:$
```

---

43. **Command Name**: *id*

    Syntax: *id* [options] [username]

    Usage: Display user and group information for a specific user or the current user.

    Commands Used:

    - id: Display user and group information for the current user.
    - id username: Display user and group information for the specified user.
    - id -u: Display only the user ID of the current user.
    - id -g: Display only the group ID of the current user.

    Output:

```
sulav@mesulav:$ id
uid=1000(tyzrex) gid=1000(tyzrex) groups=1000(tyzrex),90(network),98(power),95
2(docker),972(libvirt),984(users),987(storage),991(lp),994(input),996(audio),9
98(wheel)
sulav@mesulav:$ id tyzrex
uid=1000(tyzrex) gid=1000(tyzrex) groups=1000(tyzrex),998(wheel),996(audio),99
4(input),991(lp),987(storage),984(users),90(network),98(power),972(libvirt),95
2(docker)
sulav@mesulav:$ id -u
1000
sulav@mesulav:$ id -g
1000
```

---

44. **Command Name**: ps

    Syntax: *ps* [options]

    Usage: The *ps* command is used to display information about running processes on a Linux or Unix system.

    Commands Used:

    - ps: Display information about all running processes.
    - ps -u sulav: Display information about processes owned by a specific user.
    - ps -f: Display information about processes in a tree format.
    - ps -l: Display detailed information about processes.
    - ps -e: Display all processes, including those without a controlling terminal.

Output

```
sulav@mesulav:$ ps
    PID TTY          TIME CMD
 638483 pts/0    00:00:00 zsh
 638513 pts/0    00:00:00 bash
 656588 pts/0    00:00:00 ps
sulav@mesulav:$ ps -u sulav
    PID TTY          TIME CMD
sulav@mesulav:$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
tyzrex    638483  638469  0 20:39 pts/0    00:00:00 /bin/zsh
tyzrex    638513  638483  0 20:39 pts/0    00:00:00 bash
tyzrex    656638  638513  0 21:12 pts/0    00:00:00 ps -f
sulav@mesulav:$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  638483  638469  0  80   0 -  2878 sigsus pts/0    00:00:00 zsh
0 S  1000  638513  638483  0  80   0 -  1993 do_wai pts/0    00:00:00 bash
4 R  1000  656655  638513  0  80   0 -  2554 -      pts/0    00:00:00 ps
sulav@mesulav:$ ps -e
    PID TTY          TIME CMD
      1 ?        00:00:04 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      5 ?        00:00:00 slub_flushwq
      6 ?        00:00:00 netns
      8 ?        00:00:00 kworker/0:0H-events_highpri
     10 ?        00:00:00 mm_percpu_wq
     12 ?        00:00:00 rcu_tasks_kthread
     13 ?        00:00:00 rcu_tasks_rude_kthread
     14 ?        00:00:00 rcu_tasks_trace_kthread
     15 ?        00:00:01 ksoftirqd/0
     16 ?        00:00:23 rcu_preempt
     17 ?        00:00:00 rcub/0
     18 ?        00:00:00 migration/0
```

---

45. **Command Name**: top

    Syntax: *top* [options]

    Usage: The *top* command is used to display information about running processes on a Linux or Unix system.

    Commands Used:

    - top: Display real-time information about the system, including CPU usage, memory usage, and running processes.
    - top -i : Do not show any idle or zombie processes.
    - top -c: Display the command name of each process.

    Output:

```
sulav@mesulav:$ top
top - 21:13:48 up 1 day,  3:55,  1 user,  load average: 0.39, 0.49, 0.52
Tasks: 263 total,   2 running, 261 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.1 us,  1.6 sy,  0.0 ni, 92.8 id,  0.0 wa,  0.4 hi,  0.2 si,  0.0 s
MiB Mem :   7362.1 total,    977.9 free,   2707.9 used,   3676.3 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   4227.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
637284	tyzrex	20	0	1129.4g	227620	119484	S	18.9	3.0	1:13.60
637194	tyzrex	20	0	1121.3g	183160	129040	S	4.0	2.4	0:23.35
629419	root	20	0	25.1g	263704	189096	S	2.7	3.5	0:59.76
637241	tyzrex	20	0	32.6g	203036	145832	S	2.3	2.7	0:15.99
632456	tyzrex	20	0	1130.9g	286280	111388	S	2.0	3.8	0:41.27
629539	tyzrex	20	0	608272	132892	100892	S	0.7	1.8	0:32.01
629578	tyzrex	20	0	845428	24352	18884	S	0.7	0.3	0:12.36
638469	tyzrex	20	0	1444060	173828	124156	S	0.7	2.3	0:08.40
657569	**tyzrex**	**20**	**0**	**239288**	**29072**	**25364**	**R**	**0.7**	**0.4**	**0:00.02**
750	root	20	0	1575744	28876	7564	S	0.3	0.4	2:58.77
632387	tyzrex	20	0	32.6g	372864	205332	S	0.3	4.9	0:36.39
632424	tyzrex	20	0	32.4g	121544	98980	S	0.3	1.6	0:07.59
637253	tyzrex	20	0	32.3g	71460	59688	S	0.3	0.9	0:00.13
657122	**tyzrex**	**20**	**0**	**10932**	**4280**	**3396**	**R**	**0.3**	**0.1**	**0:00.15**
1	root	20	0	169484	9500	5784	S	0.0	0.1	0:04.51
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03

```
sulav@mesulav:$ top -i
top - 21:15:07 up 1 day,  3:56,  1 user,  load average: 0.31, 0.45, 0.51
Tasks: 263 total,   1 running, 262 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.4 us,  2.2 sy,  0.0 ni, 93.9 id,  0.0 wa,  0.4 hi,  0.2 si,  0.0 s
MiB Mem :   7362.1 total,    942.7 free,   2742.7 used,   3676.7 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   4192.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
629419	root	20	0	25.1g	263704	189096	S	7.2	3.5	1:01.77
629539	tyzrex	20	0	608276	132892	100892	S	2.3	1.8	0:32.62
638469	tyzrex	20	0	1444060	173828	124156	S	1.5	2.3	0:08.79
617775	root	20	0	0	0	0	I	0.4	0.0	0:02.03
629560	tyzrex	20	0	677956	24016	16368	S	0.4	0.3	0:00.37
629578	tyzrex	20	0	845428	24356	18884	S	0.4	0.3	0:12.70
632423	tyzrex	20	0	32.7g	255048	174732	S	0.4	3.4	0:47.06
632827	tyzrex	20	0	1130.9g	146304	104708	S	0.4	1.9	0:00.72
657778	tyzrex	20	0	1132.9g	85460	64432	S	0.4	1.1	0:00.04
658235	**tyzrex**	**20**	**0**	**10932**	**4424**	**3552**	**R**	**0.4**	**0.1**	**0:00.02**

```
top - 21:14:25 up 1 day,  3:56,  1 user,  load average: 0.58, 0.51, 0.53
Tasks: 263 total,   2 running, 261 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.0 us,  2.0 sy,  0.0 ni, 93.1 id,  3.0 wa,  0.0 hi,  0.0 si,  0.0 s
MiB Mem :   7362.1 total,    958.9 free,   2723.3 used,   3679.9 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   4208.4 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+
632387	tyzrex	20	0	32.6g	382836	205364	S	6.7	5.1	0:36.64
1	root	20	0	169484	9500	5784	S	0.0	0.1	0:04.51
2	root	20	0	0	0	0	S	0.0	0.0	0:00.03
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00
14	root	20	0	0	0	0	I	0.0	0.0	0:00.00
15	root	20	0	0	0	0	S	0.0	0.0	0:01.40
16	**root**	**-2**	**0**	**0**	**0**	**0**	**R**	**0.0**	**0.0**	**0:23.03**

---

46. **Command Name**: kill

    Syntax: *kill* [options] PID

    Usage: The *kill* command is used to send a signal to a process, allowing you to terminate it or modify its behavior.

    Commands Used:

    - kill PID: Terminate the process with the specified process ID.
    - kill -l: List all available signal names and numbers.
    - kill -9 32426: Terminate the process with the specified process ID using the SIGKILL signal, which cannot be caught or ignored by the process.

    Output:

```
sulav@mesulav:$ sudo kill 6
[sudo] password for tyzrex:
sulav@mesulav:$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+1
3
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-1
2
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
sulav@mesulav:$ kill -9 32426
```

---

47. **Command Name**: pkill

    Syntax: *pkill* [options] pattern

    Usage: The *pkill* command is used to send a signal to one or more processes based on a pattern in their process names.

    Commands Used:

    - pkill processname: Send the default signal (SIGTERM) to all processes that match the specified process name.
    - pkill -f pattern: Send the default signal to all processes that match the specified pattern in their command line arguments.
    - pkill -u username: Send the default signal to all processes owned by the specified user.

    Output:

```
sulav@mesulav:$ pkill zathura
sulav@mesulav:$ pkill -f python
pkill: killing pid 643 failed: Operation not permitted
pkill: killing pid 704 failed: Operation not permitted
sulav@mesulav:$ pkill -u sulav
sulav@mesulav:$
```

---

48. **Command Name**: killall

    Syntax: *killall* [options] command_name

    Usage: The *killall* command is used to send a signal to all processes that are associated with a given command name.

    Commands Used:

    - killall polybar: Send the default signal (SIGTERM) to all processes that match the specified process name.
    - killall -e polybar: Only match processes whose entire command line matches the specified process name.
    - killall -u sulav: Only match processes that are owned by the specified user.

Output

```
→  ~ bash
sulav@mesulav:$ killall polybar
sulav@mesulav:$ killall -e zathura
sulav@mesulav:$ killall -u sulav
sulav@mesulav:$ ~
```

---

49. **Command Name**: bg

    Syntax: *bg* [job_spec]

    Usage: The *bg* command is used to place a suspended or stopped job in the background, allowing it to continue executing without being connected to the terminal.

    Commands Used:

    - bg: Place the most recent suspended or stopped job in the background.
    - bg %job_spec: Place the job with the specified job ID (job_spec) in the background.

    Output:

```
→  ~ bash
sulav@mesulav:$ jobs
sulav@mesulav:$ sleep 500
^Z
[1]+  Stopped                 sleep 500
sulav@mesulav:$ bg
[1]+ sleep 500 &
sulav@mesulav:$ bg %1
bash: bg: job 1 already in background
sulav@mesulav:$ jobs
[1]+  Running                 sleep 500 &
sulav@mesulav:$
```

---

50. **Command Name:** fg

    Syntax: fg [job_id]

    Usage: The fg command is used to bring a job that is running in the background to the foreground.

    Commands Used:

    - fg %1: Brings the job with ID 1 to the foreground.

    Output:

```
➜  ~ bash
sulav@mesulav:$ jobs
sulav@mesulav:$ sleep 500
^Z
[1]+  Stopped                 sleep 500
sulav@mesulav:$ bg
[1]+ sleep 500 &
sulav@mesulav:$ bg %1
bash: bg: job 1 already in background
sulav@mesulav:$ jobs
[1]+  Running                 sleep 500 &
sulav@mesulav:$ fg %1
sleep 500
```

---

51. **Command Name:** tar

    Syntax: tar [options] [file or directory]

    Usage: The tar command is used to create and manipulate archive files.

    Commands Used:

    - tar -cvf archive.tar file1 file2: Creates a new archive file named archive.tar and adds file1 and file2 to it.
    - tar -xvf archive.tar: Extracts the contents of the archive.tar file to the current directory.
    - tar -tvf archive.tar: Lists the contents of the archive.tar file

    Output:

```
sulav@mesulav:$ tar -cvf archive.tar file1 file2
file1
file2
sulav@mesulav:$ tar -xvf archive.tar
file1
file2
sulav@mesulav:$ tar -tvf archive.tar
-rw-r--r-- tyzrex/tyzrex    33 2023-04-23 20:39 file1
-rw-r--r-- tyzrex/tyzrex    36 2023-04-23 20:40 file2
```

---

52. **Command Name**: gzip

    Syntax: gzip [options] [file]

    Usage: The gzip command is used to compress files.

    Commands Used:

    - gzip testfile : Compresses the file named testfile.
    - gzip -r testdir : Recursively compresses all files in the directory named testdir.
    - gzip -c testfile > testfile.gz : Compresses the file named testfile and saves it as testfile.gz.

    Output:

```
sulav@mesulav:$ gzip testfile
sulav@mesulav:$ ls
archive.tar         filerenamed.txt      linux-5.12.3.tar.xz.1
combinedrename.sh   helloworld.sh        OS_lab.pdf
copiedLab1          index.html           sheesh.sh
file1               Lab1                 testfile.gz
file2               linux-5.12.3.tar.xz  ui8-fleet-html.herokuapp.com
sulav@mesulav:$ gzip -r copiedLab1/
sulav@mesulav:$ gzip -c index.html > index.gz
sulav@mesulav:$ ls
archive.tar         helloworld.sh          OS_lab.pdf
combinedrename.sh   index.gz               sheesh.sh
copiedLab1          index.html             testfile.gz
file1               Lab1                   ui8-fleet-html.herokuapp.com
file2               linux-5.12.3.tar.xz
filerenamed.txt     linux-5.12.3.tar.xz.1
sulav@mesulav:$
```
