20221104_linux_system

- Installing windows subsystem for linux (wsl)
    - open `command prompt` as `administrator`
    - `wsl -- install`
    - restart the pc

Vim Installation

- open `ubuntu`
- type `which vim` to check whether `vim` is already installed or not.
- `sudo apt install vim-nox` - to install special version of vim

> This package contains a version of vim compiled with support for scripting with Lua, Perl, Python 3, Ruby, and Tcl but no GUI

Basic Linux commands

1. whoami

- who am I logged in as - user name of currently logged in user.

2. man

- get a manual or documentation on `commands`. like help. manual on `whoami`
- `q` to get out of the page.

```
WHOAMI(1)                                    User Commands                                    WHOAMI(1)

NAME
       whoami - print effective userid

SYNOPSIS
       whoami [OPTION]...

DESCRIPTION
       Print the user name associated with the current effective user ID.  Same as id -un.

       --help display this help and exit

       --version
              output version information and exit

AUTHOR
       Written by Richard Mlynarik.

REPORTING BUGS
       GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
       Report whoami translation bugs to <https://translationproject.org/team/>

COPYRIGHT
       Copyright © 2018 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
       This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
       Full documentation at: <https://www.gnu.org/software/coreutils/whoami>
       or available locally via: info '(coreutils) whoami invocation'

GNU coreutils 8.30                           September 2019                                    WHOAMI(1)
 Manual page whoami(1) line 1/32 (END) (press h for help or q to quit)
```

3. clear or (ctrl + l)

- to clear the terminal

4. pwd

- your current working directory

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ pwd
/mnt/c
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$
```

5. ls

- to list the directories or folders

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ ls
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'PageFile.sys': Permission denied
ls: cannot access 'swapfile.sys': Permission denied
'$Recycle.Bin'          'DL classes'             Intel                      PageFile.sys          Recovery                          bootTel.dat
'$WinREAgent'           'Documents and Settings' 'ML Theory'                PerfLogs              Resume                            datasets
 ASAP                    DumpStack.log.tmp       'MLOps - Notes'           'Personal Details'    'Slides - DL'                      hiberfil.sys
 Aptitude               'Edge Computing'         'NVIDIA - Presentation Slides' 'Program Files'  'System Volume Information'        project
'Books On ML and DL'     Gernolt                  OEM                       'Program Files (x86)'  Users                            swapfile.sys
 Certifications          Git                      OneDriveTemp               ProgramData           Windows
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ ls Personal Details
ls: cannot access 'Personal': No such file or directory
ls: cannot access 'Details': No such file or directory
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ ls 'Personal Details'
 ApplicationFormDraftPrintForAll.pdf  'GOWTHAM SIGNATURE.jpg'      'Gowtham_Unni_T_Aadhar card.jpg'        'SSLC Certificate.pdf'
 ConfirmationPage-223510414917.pdf    'GOWTHAM UNNI.jpg'           'Gowtham_Unni_T_PAN Card.jpg'           'Screenshot 2022-08-16 07.05.06.png'
'GOWTHAM (2).jpg'                      GowthamInternshipLetterSigned.pdf 'Graduarion- Consolidated Grade Card.pdf' 'Windows 10'
'GOWTHAM IMG (2).jpg'                 'Gowtham_Unni_ T_AIML ID Card.png' 'Plus Two Certificate.pdf'            scientific_assistant_ssc.pdf
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ ls 'Program Files'
 7-Zip              Git               MSBuild                 'Mozilla Firefox'           'Uninstall Information'  'Windows Photo Viewer'
 Acer              'GoTrust ID Plugin' 'Microsoft Office'      'NVIDIA Corporation'         WinRAR                  'Windows Sidebar'
 BraveSoftware      Google            'Microsoft Office 15'    'NVIDIA GPU Computing Toolkit' 'Windows Defender'     WindowsApps
 CodeBlocks         Intel            'Microsoft Update Health Tools' 'Norton Security'       'Windows Mail'          WindowsPowerShell
'Common Files'     'Internet Explorer' 'Microsoft Visual Studio'  PCHealthCheck           'Windows Media Player'   desktop.ini
 DriverSetupUtility JetBrains         ModifiableWindowsApps   'Reference Assemblies'      'Windows NT'
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ ls 'Program Files'/Acer
'Quick Access Service'   StorPSTL  'User Experience Improvement Program Service'
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$
```

- `ls -l` : gives other details related to the folder.

- `ls -a` : shows additional files and folders beginning with `.`

6. cd

- to change the current working directory

- `cd ~` - home directory.
- `cd /` - root directory.

7. mkdir

- make directory.

8. touch

- to create an empty file(with any extension).

9. rmdir

- remove empty directories.

10. rm command

- to remove files or directories.
  - `-v` - verbose (to show the log or result of command)
  - `-i` - `interactive` . asks users permission when removing the files or directories.
  - `-r` - `recursive` : to delete non - empty directories.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
bed  dwayne.txt  night.txt  rain
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ rm dwayne.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
bed   night.txt  rain
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ rm -v bed
removed 'bed'
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ cd ..
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102$ rm -i learnlinux
rm: cannot remove 'learnlinux': Is a directory
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102$ rm -ri learnlinux
rm: descend into directory 'learnlinux'? y
rm: remove regular empty file 'learnlinux/night.txt'? n
rm: remove regular empty file 'learnlinux/rain'? y
rm: remove directory 'learnlinux'? n
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102$ ls
learnlinux
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102$ ls learnlinux/
night.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102$
```

11. xdg-open

- to open files or folders

- xdg-open night.txt or `xdg-open .`

```
NAME
      xdg-open - opens a file or URL in the user's preferred application

SYNOPSIS
      xdg-open {file | URL}

      xdg-open {--help | --manual | --version}

DESCRIPTION
      xdg-open opens a file or URL in the user's preferred application. If a URL is provided the URL will be opened
      in the user's preferred web browser. If a file is provided the file will be opened in the preferred
      application for files of that type. xdg-open supports file, ftp, http and https URLs.

      xdg-open is for use inside a desktop session only. It is not recommended to use xdg-open as root.
```

12. mv

1. to `move` and `rename` folders and files.
    1. to rename a the folder night just enter `mv night.txt NIGHT.txt`. And that's renamed.(-v for verbose)

    2. to rename the folder `mv learnlinux/ Learnlinux`.

    3. to move the files to a folder `mv night.txt rain learnlinux/`. It would identify the last one as the folder to be moved to.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
night.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ mkdir morning
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ mv night.txt evernight.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
evernight.txt  morning
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ mv -v evernight.txt morning
renamed 'evernight.txt' -> 'morning/evernight.txt'
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
morning
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls morning
evernight.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ touch bird
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ |
```

13. cp

- to copy file or folder.

- `cp -r learnlinux Learlinux`

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ cp bird crow
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
bird   crow   morning
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ cp crow morning/evernight.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls morning/
evernight.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ cp -r morning/ night
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
bird   crow   morning   night
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls night/
evernight.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$
```

14. head

- outputs the first part of the file. By default it's set to 10. But the number of lines to be printed can be set using `-n` command.

15. tail

- similar to `head` but prints the end part of the file.

16. date

```
NAME
        date - print or set the system date and time

SYNOPSIS
        date [OPTION]... [+FORMAT]
        date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
        Display the current time in the given FORMAT, or set the system date.

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

17. redirecting the output.

1. `>` : redirects the outputs to a file (like a text file). But this command would replace the current contents.
2. `>>` : this commands appends instead of overwriting.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ ls
bird  bird.txt  crow  morning  night
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ xdg-open bird.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ man date > bird.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ xdg-open bird.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ head -n 20 bird.txt
DATE(1)                                        User Commands

NAME
       date - print or set the system date and time

SYNOPSIS
       date [OPTION]... [+FORMAT]
       date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
       Display the current time in the given FORMAT, or set the system date.

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

       -d, --date=STRING
              display time described by STRING, not 'now'

       --debug
              annotate the parsed date, and warn about questionable usage to stderr
```

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ date >> date.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ date >> date.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ date >> date.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ date >> date.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ date >> date.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ head date.txt
Thu Nov  3 10:47:55 IST 2022
Thu Nov  3 10:47:56 IST 2022
Thu Nov  3 10:47:57 IST 2022
Thu Nov  3 10:47:58 IST 2022
Thu Nov  3 10:47:58 IST 2022
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$
```

18. cat

- concatenates the files and prints it out. If enter a single file the it would just print everything in it.
  `cat` along with `redirection` has plenty of possibilites.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ cat bird bird.txt > crow
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/New folder/codes/20221102/learnlinux$ head crow
bird
bird.txt
crow
date.txt
morning
night
now
crow
peacock
hen
```

19. [piping](#)

- `ls -l | wc` - listed directories to word count.

- to send outputs of one command to another command.

- Pipes are unidirectional i.e data flows from left to right through the pipeline.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$ ls -l | wc
ls: cannot access 'DumpStack.log.tmp': Permission denied
ls: cannot access 'hiberfil.sys': Permission denied
ls: cannot access 'PageFile.sys': Permission denied
ls: cannot access 'swapfile.sys': Permission denied
ls: 'System Volume Information': Permission denied
     37      342     2412
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c$
```

20. wc

- prints the word count
- `-l` - line count
- `-c` - bytes count
- `-m` - character count

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221104$ ls -l | wc
      5       38      289
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221104$
```

- First term is the number of lines.
- Second term is number of characters
- Third term is `bytes`.

21. sort

- sort lines of text files

22.uniq

- print unique lines (omit repeated ones).

23. grep

- to search for patterns with in a file.if a file is not given then if recursive (-r) then searchs through all files in working directory and nonrecursive searches read standard input.

```
NAME
       grep, egrep, fgrep, rgrep - print lines that match patterns

SYNOPSIS
       grep [OPTION...] PATTERNS [FILE...]
       grep [OPTION...] -e PATTERNS ... [FILE...]
       grep [OPTION...] -f PATTERN_FILE ... [FILE...]

DESCRIPTION
       grep  searches  for  PATTERNS in each FILE.  PATTERNS is one or more patterns separated by newline characters,
       and grep prints each line that matches a pattern.  Typically PATTERNS should be quoted when grep is used in  a
       shell command.

       A  FILE  of  "-"  stands  for  standard  input.   If  no FILE is given, recursive searches examine the working
       directory, and nonrecursive searches read standard input.

       In addition, the variant programs egrep, fgrep and rgrep are  the  same  as  grep -E,  grep -F,  and  grep -r,
       respectively.  These variants are deprecated, but are provided for backward compatibility.
```

```
Options Description
-c : This prints only a count of the lines that match a pattern
```

```
-h : Display the matched lines, but do not display the filenames.
-i : Ignores, case for matching
-l : Displays list of a filenames only.
-n : Display the matched lines and their line numbers.
-v : This prints out all the lines that do not matches the pattern
-e exp : Specifies expression with this option. Can use multiple times.
-f file : Takes patterns from file, one per line.
-E : Treats pattern as an extended regular expression (ERE). search for more
than one expression.
-w : Match whole word
-o : Print only the matched parts of a matching line,with each such part on
a separate output line.
-A n : Prints searched line and nlines after the result.
-B n : Prints searched line and n line before the result.
-C n : Prints searched line and n lines after before the result.
```


- `^` : to find lines that strat with that particular string add `^` in the beginning.
- `$` : to find lines that ends with that string add `$` to the end.

24. [tee](tee)

- reads the standard file print the output and copies the output to new file.

```
NAME
       tee - read from standard input and write to standard output and files


SYNOPSIS
       tee [OPTION]... [FILE]...


DESCRIPTION
       Copy standard input to each FILE, and also to standard output.

       -a, --append
              append to the given FILEs, do not overwrite


       -i, --ignore-interrupts
              ignore interrupt signals
```

```
       -p        diagnose errors writing to non pipes


       --output-error[=MODE]
               set behavior on write error.  See MODE below


       --help display this help and exit


       --version
               output version information and exit
```

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ ls -l | tee noon.txt
total 20
-rwxrwxrwx 1 gowthamunni gowthamunni 1702 Nov  3 11:43 harrypotter.txt
-rwxrwxrwx 1 gowthamunni gowthamunni 9801 Nov  3 11:45 lordofmysteries.txt
-rwxrwxrwx 1 gowthamunni gowthamunni 1608 Nov  3 11:44 lordofrings.txt
-rwxrwxrwx 1 gowthamunni gowthamunni    0 Nov  3 13:56 noon.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ ls
harrypotter.txt  lordofmysteries.txt  lordofrings.txt  noon.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ cat noon.txt
total 20
-rwxrwxrwx 1 gowthamunni gowthamunni 1702 Nov  3 11:43 harrypotter.txt
-rwxrwxrwx 1 gowthamunni gowthamunni 9801 Nov  3 11:45 lordofmysteries.txt
-rwxrwxrwx 1 gowthamunni gowthamunni 1608 Nov  3 11:44 lordofrings.txt
-rwxrwxrwx 1 gowthamunni gowthamunni    0 Nov  3 13:56 noon.txt
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ |
```

- `2>&1` to append the error message to a file.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ ls tansan 2>&1 | tee tacos
ls: cannot access 'tansan': No such file or directory
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ cat tacos
ls: cannot access 'tansan': No such file or directory
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ |
```

25. ssh

- `Secure Shell` or `secure socket shell`.
  The ssh command consists of three different types of components:

- ssh command: It instructs the machine to create a secure encrypted connection with the host system.

- User name: User name is the name of the Linux user, which is being accessed by the host machine.

- Host: A host is a machine that is being accessed by the user, such as a computer or a router. A domain name or an IP address also refers as Host

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ ssh -V
OpenSSH_8.2p1 Ubuntu-4, OpenSSL 1.1.1f  31 Mar 2020
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ |
```
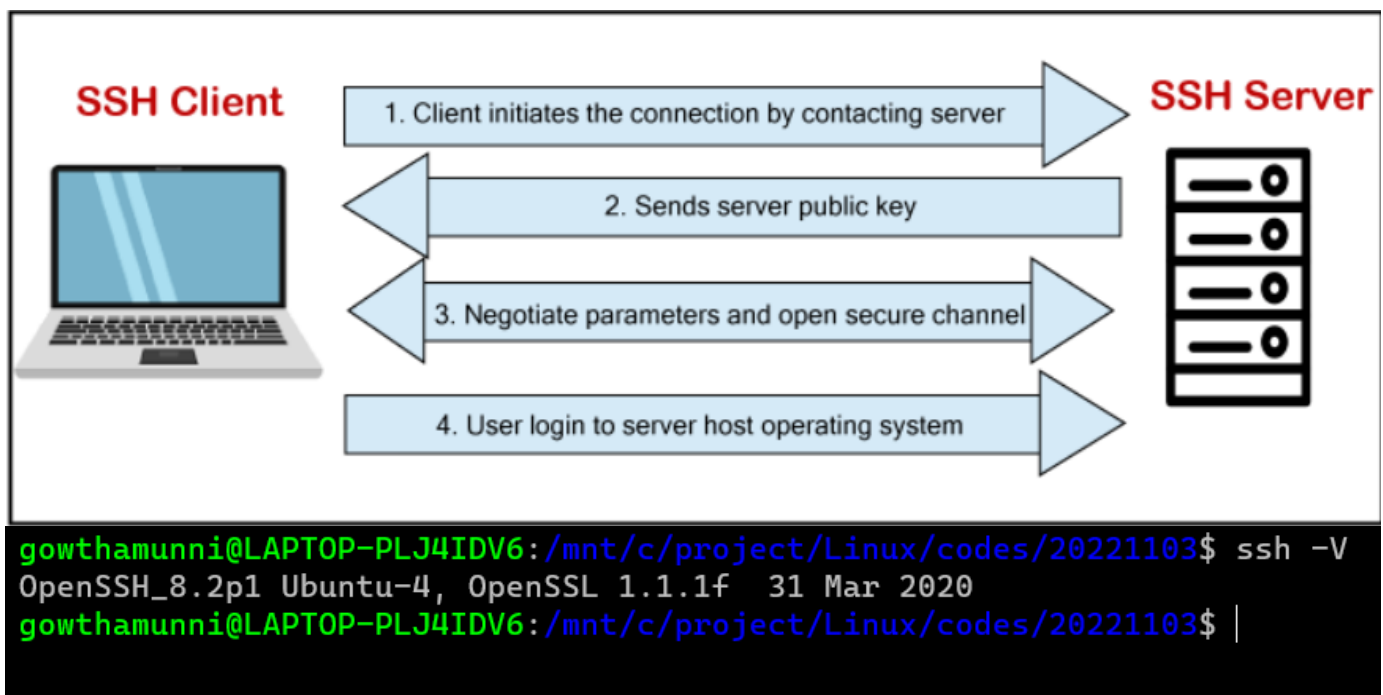
- so in this device the `OpenSSH` is preinstalled (it comes with latest ubuntu).
- `synatx`
    - `ssh [option] [user]@[server_address]`
    - the server address can be either the address of the remote host which you wish to connect or it can be the IP address of your remote host.
- `connection refused.`

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221104$ sudo apt-get install openssh-server ii
Reading package lists... Done
Building dependency tree
Reading state information... Done
ii is already the newest version (1.8-2).
openssh-server is already the newest version (1:8.2p1-4ubuntu0.5).
0 upgraded, 0 newly installed, 0 to remove and 285 not upgraded.
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221104$ ssh localhost
ssh: connect to host localhost port 22: Connection refused
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221104$ |
```

26. scp

- `secure copy protocol` - copy files from host to client and vice versa.
- `syntax`
    - `scp [OPTIONS] [[user@]src_host:]file1 [[user@]dest_host:]file2`
        - .eg : `scp test.txt userbravo@destination:/location2` if we have a file named `text.txt` , to copy it to the server. (userbravo is the username and `destination:` then location (location to which it's copied to)). This is the case for local host to remote server.
    - `scp file <remote_username>@<IPorHost>:<PathToFile> <LocalFileLocation>` - remote to local.
        - eg: `scp linuxcheatsheet lary@192.168.1.100: .` - linuxcheatsheet is the file in remote server. `.` points to current directory.

27. rsync

> rsync or remote synchronization is a software utility for Unix-Like systems that efficiently sync files and directories between two hosts or machines.

> Rsync is famous for its delta-transfer algorithm, in which it copies only the differences between the source files present in the local-host and the existing files in the destination or the remote host.

> `rsync local-file user@remote-host:remote-file`

28. [awk](#)

> pattern scanning and processing language

- it's used for processing the textfiles (standard outs).

Syntax : `awk options 'selection _criteria {action }' input-file > output-file`

awk can be used to find, seperate and write the contents of file.Search the contents (if it's a table or consider the stdout of date then we can use) prints outs the contents based on some `selection criteria`.

- what ever you want to search (the string , maybe a name, maybe salary ,maybe some files that starts with some particular character) it's typed inside `/ /`. check the link on geeksforgeeks docs for example.

use cases

1. `awk '{print}' night.txt` : prints outs whatever it's in `night.txt` . Here `print` is the action and note that the `selction criteria` is with in single quotes.

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project$ ls -l | awk '/Aug/ {print}'
drwxrwxrwx 1 gowthamunni gowthamunni   512 Aug 18 16:05 Quantization
drwxrwxrwx 1 gowthamunni gowthamunni   512 Aug 18 22:17 VisionTransformer
drwxrwxrwx 1 gowthamunni gowthamunni   512 Aug 10 10:18 onnx
drwxrwxrwx 1 gowthamunni gowthamunni   512 Aug 18 12:27 onnx-tf-conversion
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project$ ls -l | awk '/Aug/ {print}' | wc
      4      36     279
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project$ ls -l | awk '/Aug/ {print $6}'
Aug
Aug
Aug
Aug
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project$ ls -l | awk '/Aug/ {print $6}'|wc
      4       4      16
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project$ ls -l | awk '/Aug/ {print $6}'|wc -l
4
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ awk '/Mr/ {print}' harrypotter.txt
Potters. Mrs. Potter was Mrs. Dursley's sister, but they hadn't met
for several years; in fact, Mrs. Dursley pretended she didn't have a
When Mr. and Mrs. Dursley woke up on the dull, gray Tuesday
suggest that strange and mysterious things would soon be happening all over the country. Mr. Dursley hummed as he picked out his
most boring tie for work, and Mrs. Dursley gossiped away happily
At half past eight, Mr. Dursley picked up his briefcase, pecked
Mrs. Dursley on the cheek, and tried to kiss Dudley good-bye but
his cereal at the walls. "Little tyke," chortled Mr. Dursley as he left
of something peculiar — a cat reading a map. For a second, Mr.
```

[Reference to geeksforgeeks](#)

bashrc and vimrc

1) .bashrc

```
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ cat .bashrc
cat: .bashrc: No such file or directory
gowthamunni@LAPTOP-PLJ4IDV6:/mnt/c/project/Linux/codes/20221103$ cd ~
gowthamunni@LAPTOP-PLJ4IDV6:~$ ls -a
.    .bash_history  .bashrc   .landscape   .profile   .sudo_as_admin_successful
..   .bash_logout   .config   .motd_shown  .ssh        .viminfo
gowthamunni@LAPTOP-PLJ4IDV6:~$ cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
```

- it's a script file that's executed when a user logs in. It's a hidden file so it won't be visible unless we use `ls -a` to show hidden files.
- so `.bashrc` comes into play only if we are running interactive session.
- open it in vim (xdg-open .bashrc) then you can edit and customize terminal.

> The .bashrc file is a script file that's executed when a user logs in. The file itself contains a series of configurations for the terminal session. This includes setting up or enabling: coloring, completion, shell history, command aliases, and more.

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
                                                              1,1           Top
```

1.1.

> Aliases are different names for the same command. Consider them as shortcuts to a longer form command. The .bashrc file already has a set of predefined aliases.

```
# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$/
/'\''')"'
```

1.2.

```
# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
        # a case would tend to support setf rather than setaf.)
        color_prompt=yes
    else
        color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
```

1.3. user title

```
# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
    ;;
*)
    ;;
esac
```

Note:

- terminal -> settings -> ubuntu -> change starting directory.

The Linux Kernel

Another developer, Linus Torvalds, was working on creating his own kernel known as Linux. The kernel is the part of an OS that facilitates interactions between hardware and software.

At the time, many GNU "pieces" were complete, but it lacked a kernel. Torvalds combined his kernel with the existing GNU components to create a full operating system.
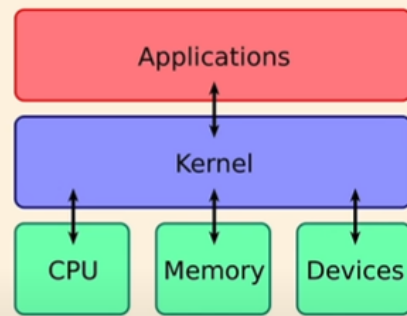
Bobbo, CC BY-SA 3.0, via Wikimedia Commons