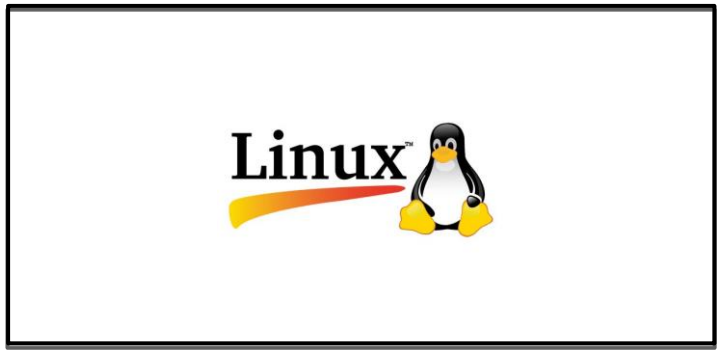
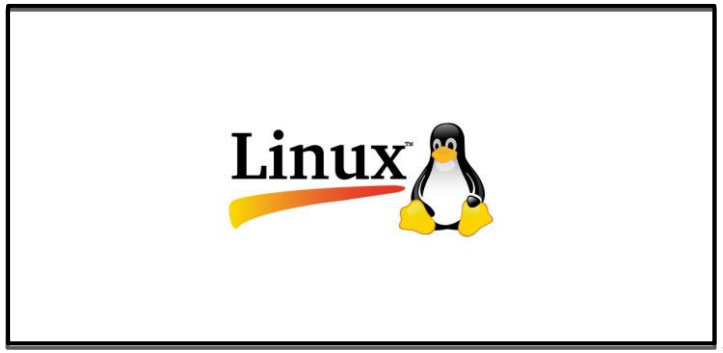
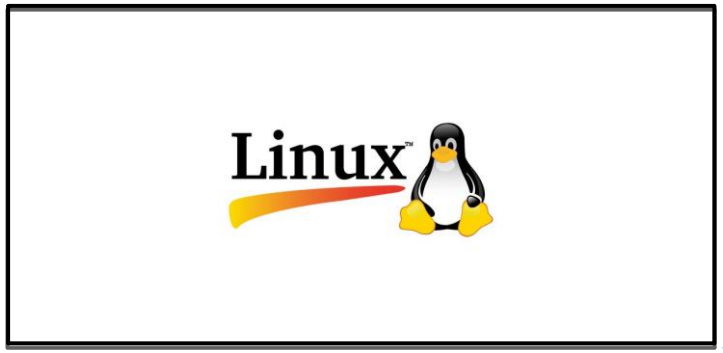


Linux



Content

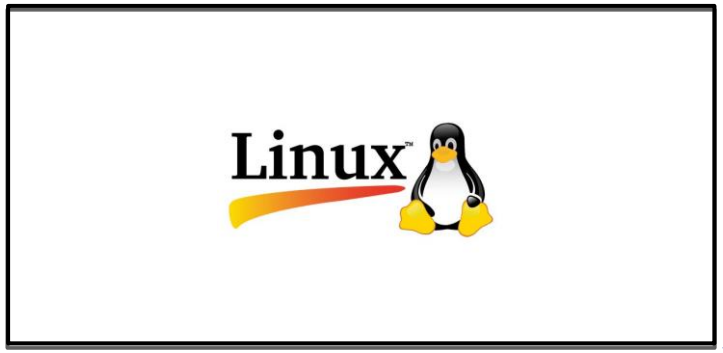
- Introduction
- Windows VS. Linux
- Linux Commands
- Bash



What is Linux?

- Linux is an open source operating system
- Different distributions (or known as distros)
- Most common one is Ubuntu

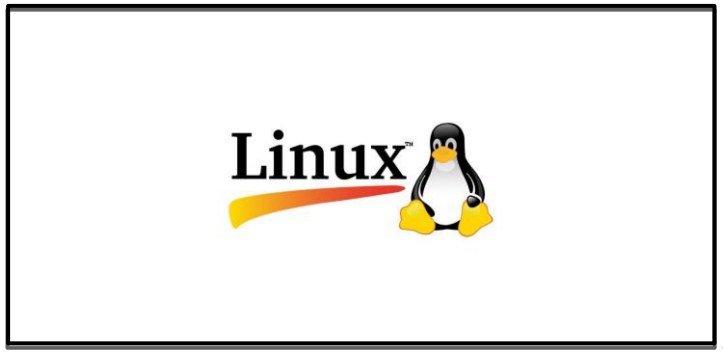
Introduction



Usage

- Servers
- Development
- Embedded Systems
- Desktop

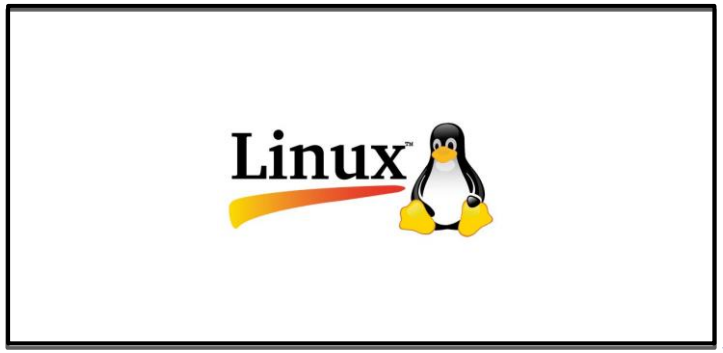
Introduction



What is an OS?

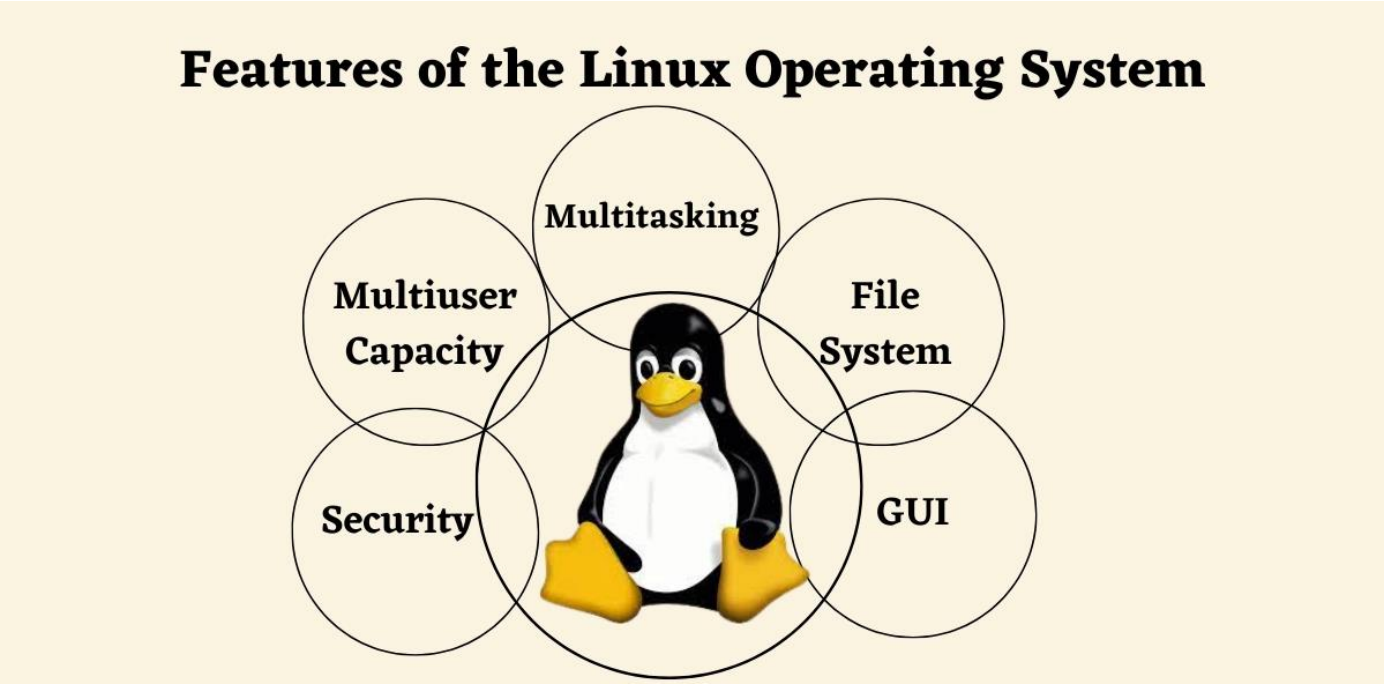
Introduction

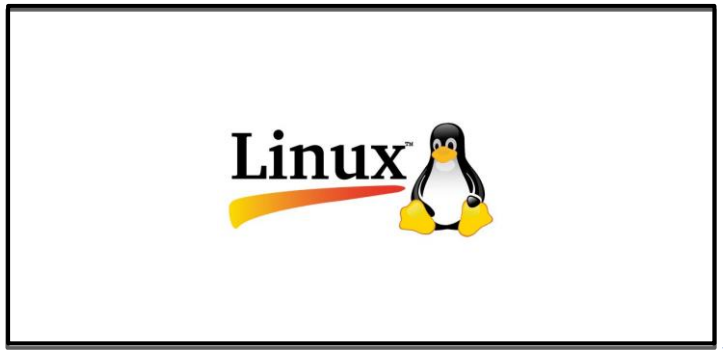
- software that manages computer hardware and software resources
- Ensures running programs don’t interfere with each other
- Process Management
- Memory Management
- File System Management
- Device Management
- Security and Access Control
- User Interface



Features

Introduction

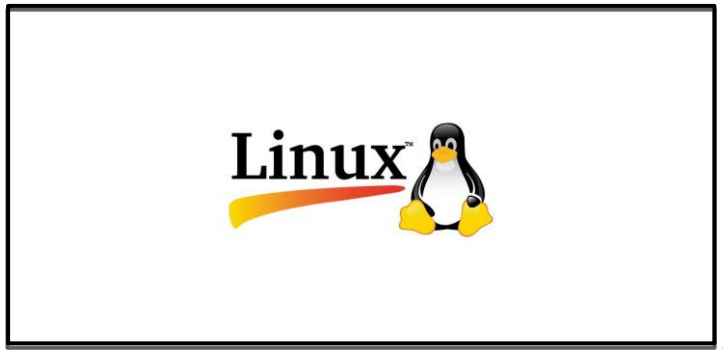




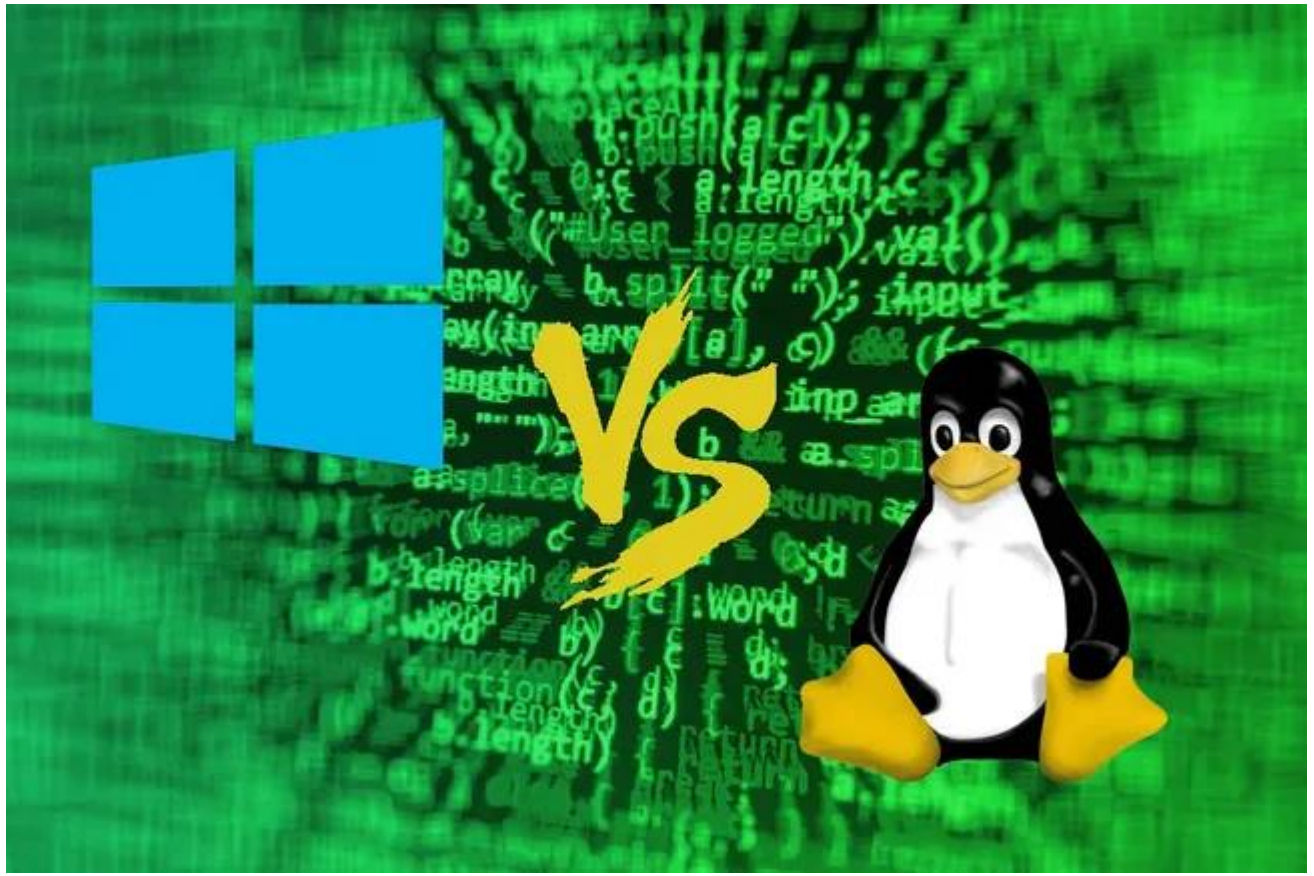
Naming

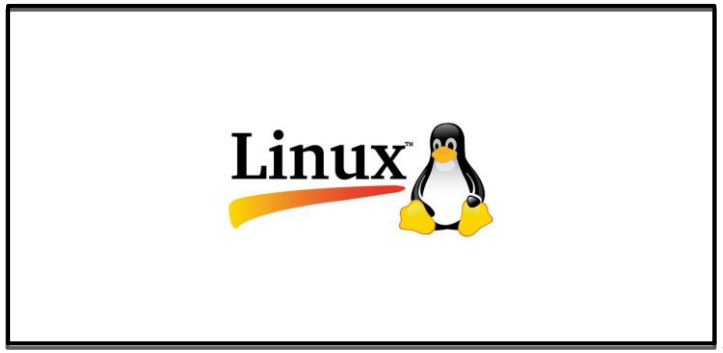
Introduction

- Named after Linus Torvalds
- Combo of **Linu**s + Uni**x**
- Developed it in 1991
- Initially was named: Freax
- Combo of: **Fre**e, **Frea**k, Uni**x**



Windows VS.
Linux

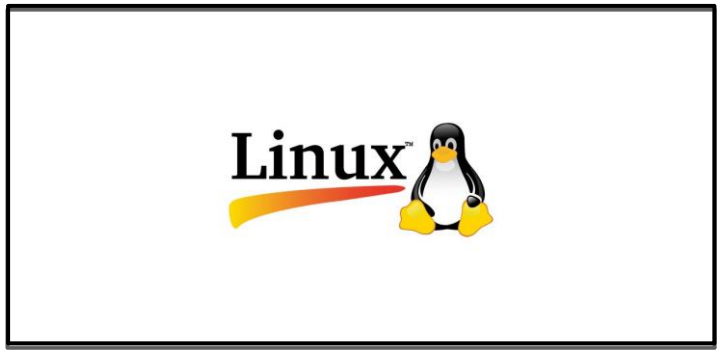




Which is better?

Windows VS. Linux

- Linux and Windows.
- Each has its own set of unique features
- advantages and disadvantages.
- While it is difficult to say which one is the better choice
- it is not as difficult to answer which is the better choice given your needs.

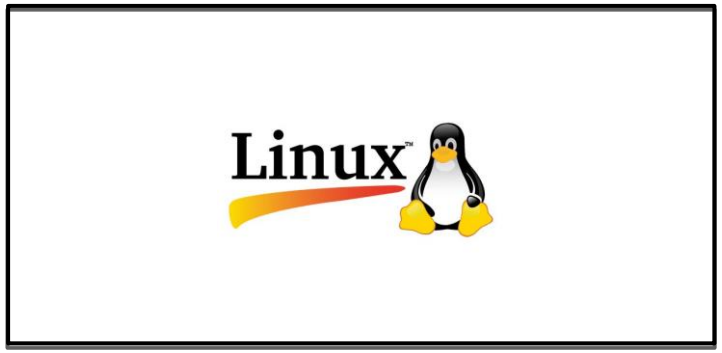


Licensing

Windows VS. Linux

Windows
- Closed Source
- Need to Purchase a License to use it

Linux
- Open-Source
- Free or Low Cost

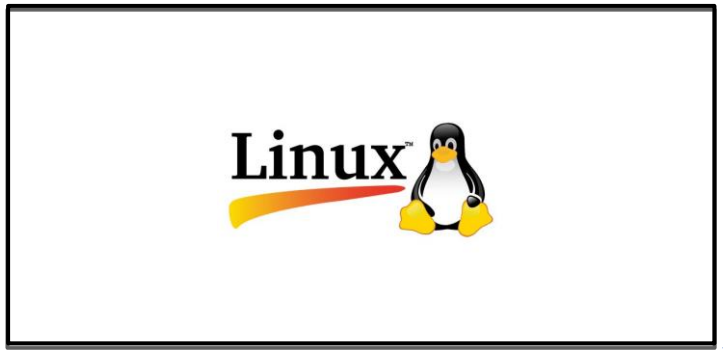


User Interface

Windows VS. Linux

Windows
- Known for its user-friendly GUI
- Customization: allows some customization, but is limited

Linux
- variety of desktop envs: GNOME, KDE, XFCE
- users normally use CLI extensively
- UI is highly customizable

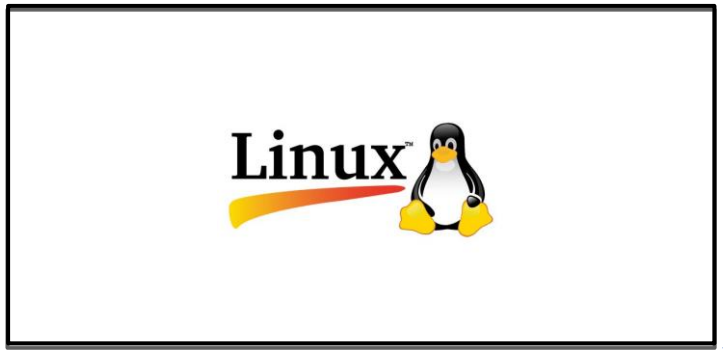


Software Support

Windows VS. Linux

Windows
- Wide Compatibility
- Limited Open Source software

Linux
- Limited Proprietary software
- Rich in open source software

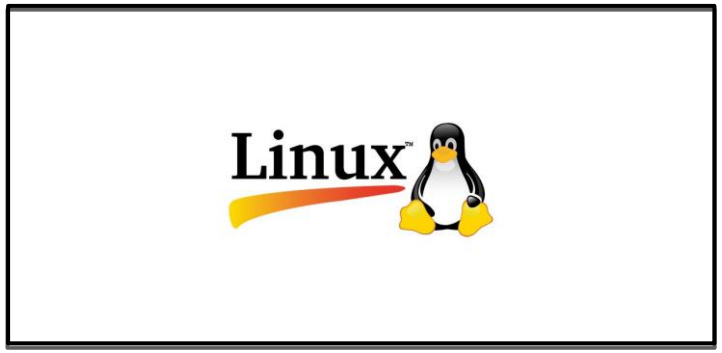


Security

Windows VS. Linux

Windows
- Target for malware
- Improved security over the years, with features like UAC

Linux
- Strong Security
- Open Source Advantage

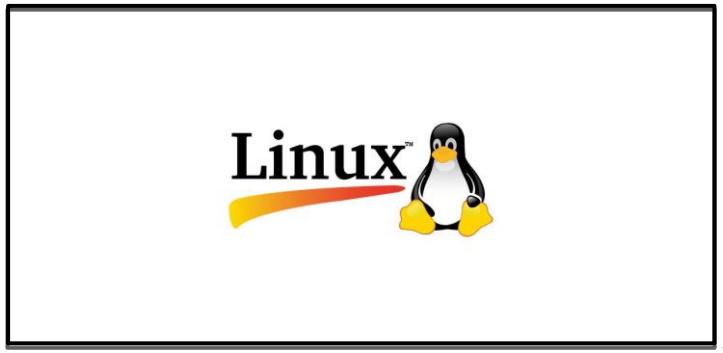


Performance

Windows VS. Linux

Windows
- Resource Intensive
- Optimized for GUI

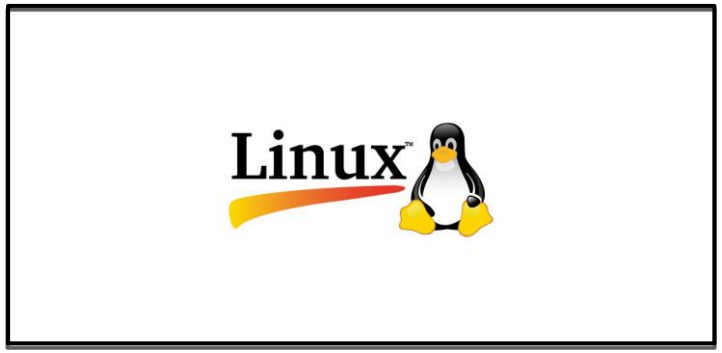
Linux
- Efficient and Lightweight
- Scalable Performance



Other differences

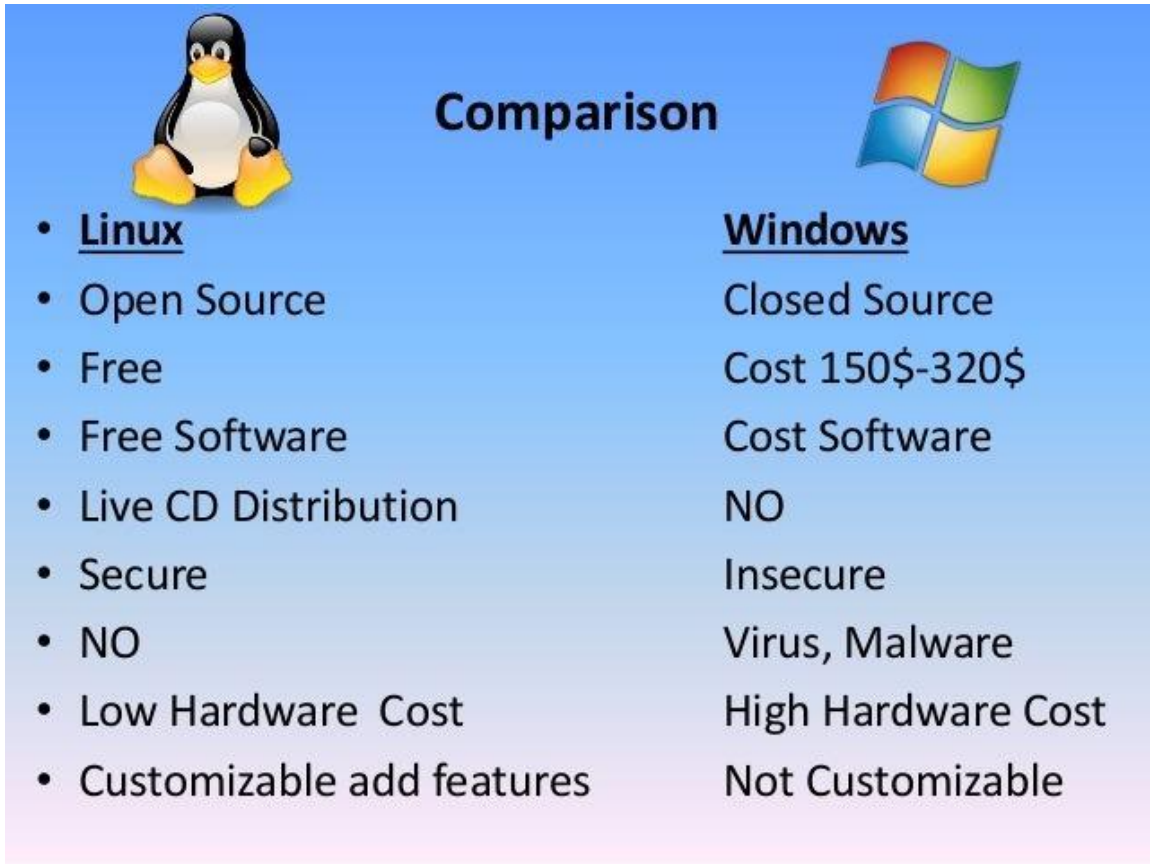
- File System and Management
- Updates and Software Management
- Community and Support
- Use Cases

Windows VS. Linux



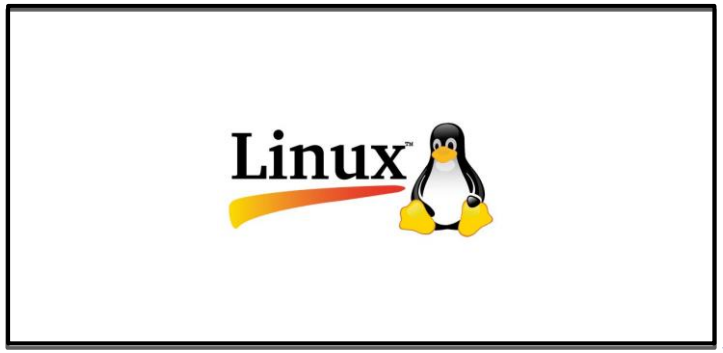
Quick Comparison

Windows VS. Linux



Linux

Linux Commands



Terminal

Linux Commands

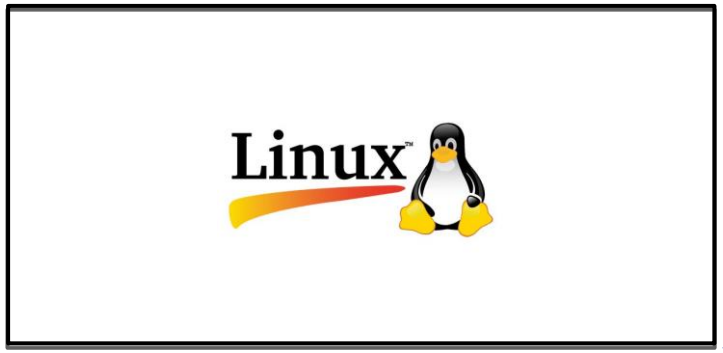
- The Terminal is a text-based interface
- where all commands in Linux are run

Usage:

- File Management
- Process Management
- Text Manipulation
- Package Management
- Networking

Run:

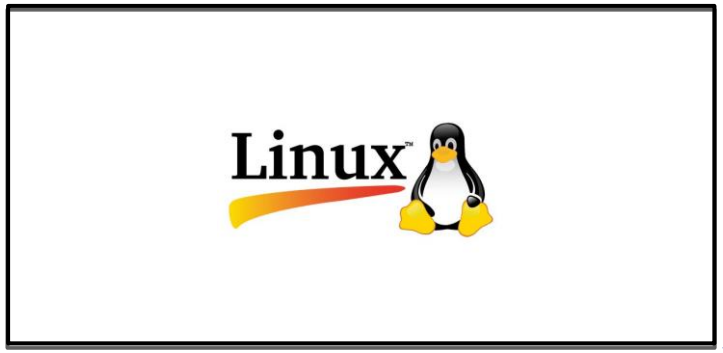
- Let’s Open a terminal
- Right click -> new terminal window



pwd command

- Get current working directory
- Always returns the “Absolute Path”

Linux Commands



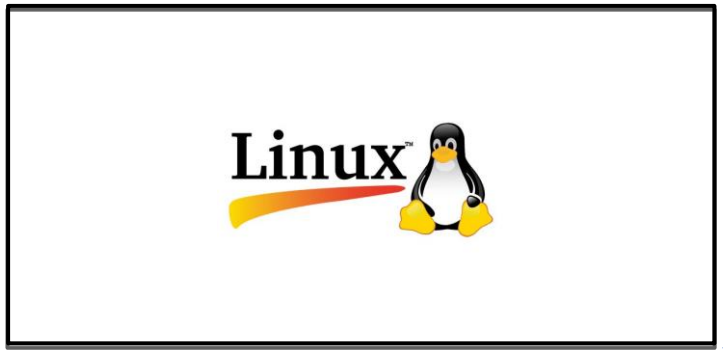
ls command

Linux Commands

- Used for listing files and directories

Usage:

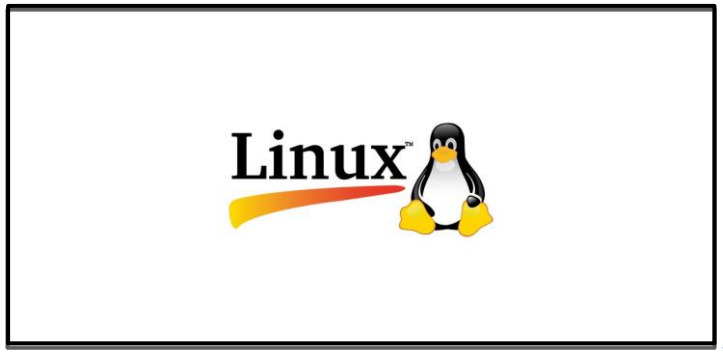
- ‘ls’ – list the files in current directory
- ‘ls file/folder name’ – list the files / folders per the given input
- ‘ll’ – not a standard command it is an alias to ‘ls –l’



ls command options

Linux Commands

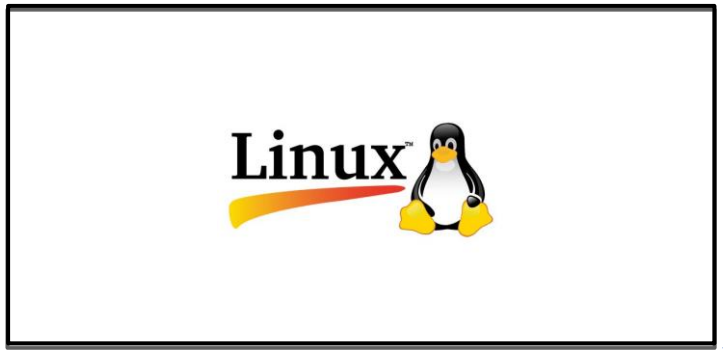
- -r : reverse the order of listing
- -l : use long listing format
- -t : sort by modification time, newest first
- -a : do not ignore entries starting with .
- -d : list directories themselves, not their contents
- Combining options



history command

Linux Commands

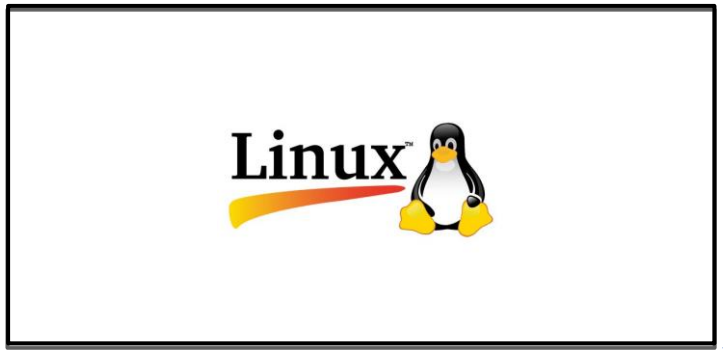
- Used for list and use the command history used so far
- !n
- !!
- !-n
- -c option



Permission Categories

- User (Owner)
- Group
- Others

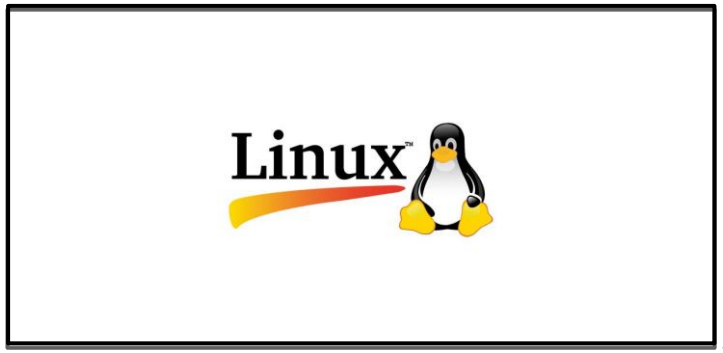
Linux Commands



Permission Types

- Read (r)
- Write (w)
- Execute (x)

Linux Commands



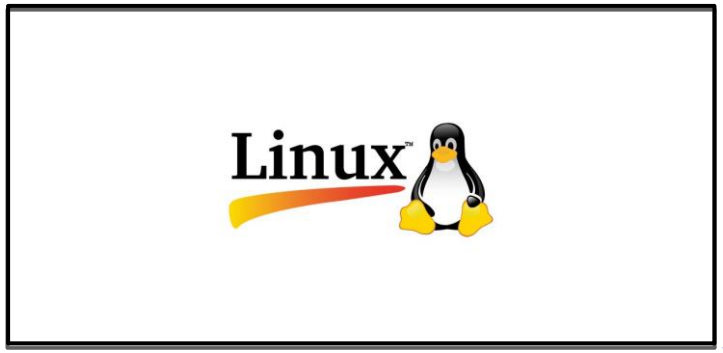
Changing Permissions

Linux Commands

- chmod command

Examples:

- Add Permission: chmod u+x file_name
  (add execute permission for the owner)
- Remove Permission: chmod g-w file_name
  (remove write permission for the group)
- Set Exact Permissions: chmod u=rwx,g=rx,o=r file_name
  (set permissions exactly)



Changing Permissions (numbers)

Linux Commands

- Permissions can also be represented using octal numbers

Each permission type has a numeric value:

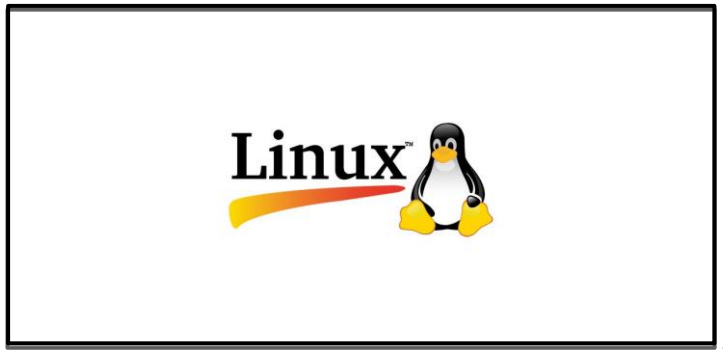
- Read (r) = 4
- Write (w) = 2
- Execute (x) = 1

Combine these values to set permissions:

- 7 = rwx (4+2+1)
- 6 = rw- (4+2)
- 5 = r-x (4+1)
- 4 = r-- (4)

Example:

- chmod 755 file_name



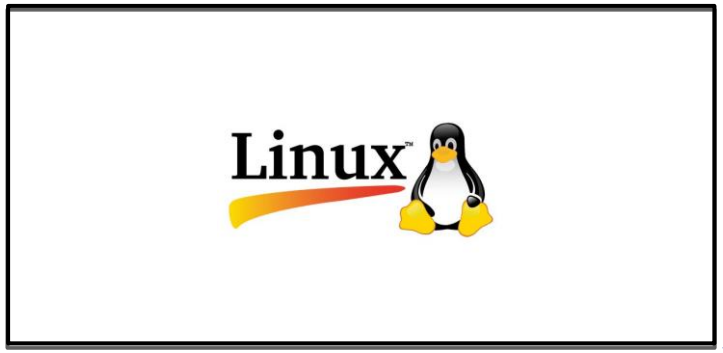
mkdir command

Linux Commands

- Creates a new directory

Usage:

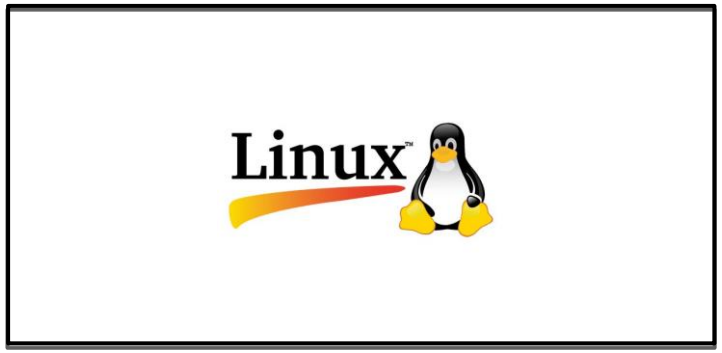
- mkdir [options] directory_name
- Create multiple directories at once



rmdir command

Linux Commands

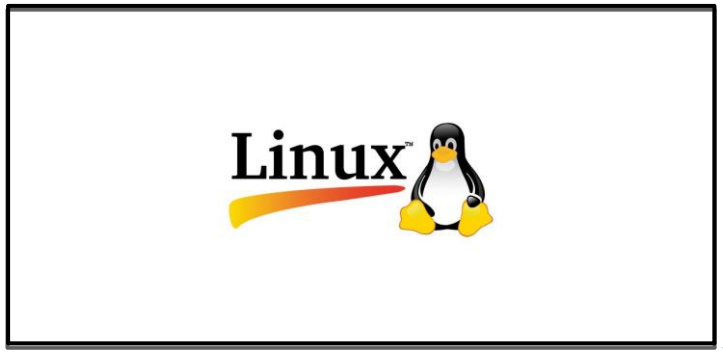
- Removes a directory
- What happens in case the directory exists?
- What if it exists and has some files and folders?
- What if it doesn’t exist?
- Removing multiple directories
- Using relative / absolute path



cd command

Linux Commands

- cd = Change Directory
- Changes the current directory to the given one
- Relative path / Absolute path
- The ‘..’ directory
- The ‘/’ directory
- The ‘~’ directory



echo command

Linux Commands

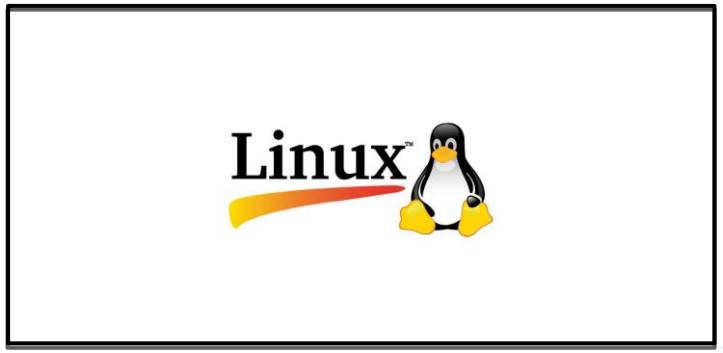
- Display a line of text to the terminal

Usage:

- echo [options] [string]

Example:

- echo Hello, World!
- echo -e "Line 1\nLine 2"



Redirection >

Linux Commands

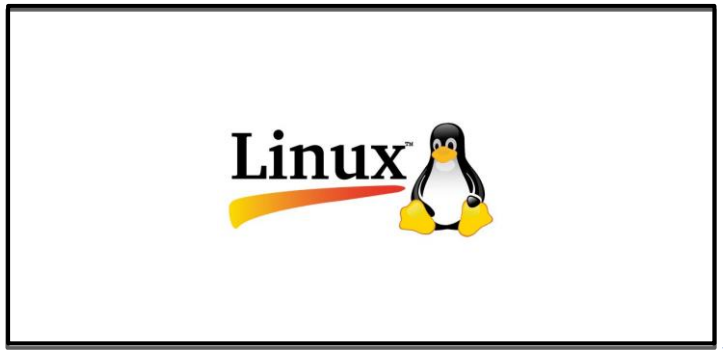
- redirection is a technique used to change the default input and output sources for commands

Example:

- echo “Hello MST World!” > “mst.txt”

types of redirection:

- > Overwrite Redirection
- >> Append Redirection
- 2>&1 Redirects the standard error to the standard output



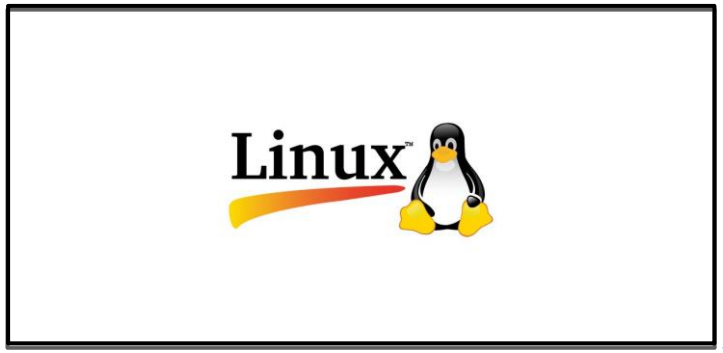
touch command

Linux Commands

- Creates a new empty file

Usage:

- touch [options] file_name
- Creating multiple files
- What happens if the file exists?



cat command

Linux Commands

- concatenates and displays the contents of files

Usage:

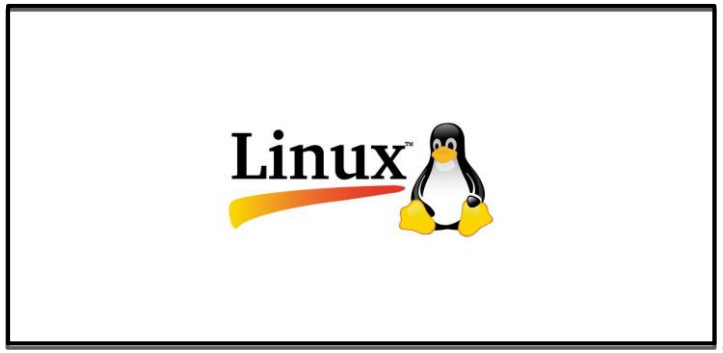
- cat [options] [file...]
- Concatenate Multiple files
- Example:
- redirect output to new file

  cat file1.txt file2.txt > combined.txt
- Create a New File

  cat > newfile.txt

- Display line numbers

  cat –n file.txt



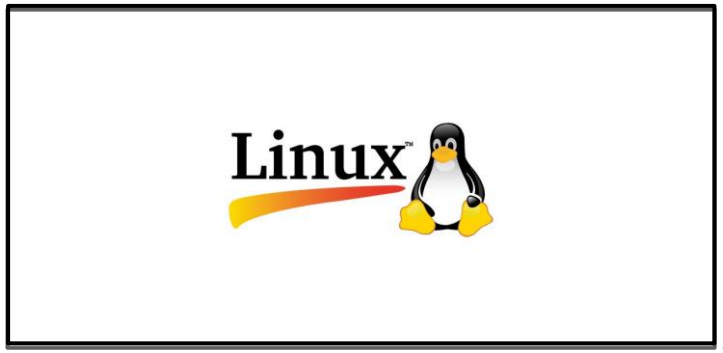
less command

Linux Commands

- Linux is a pager program used for viewing the contents of a file one screen at a time
- View multiple files
- -N : show line numbers

Navigation:

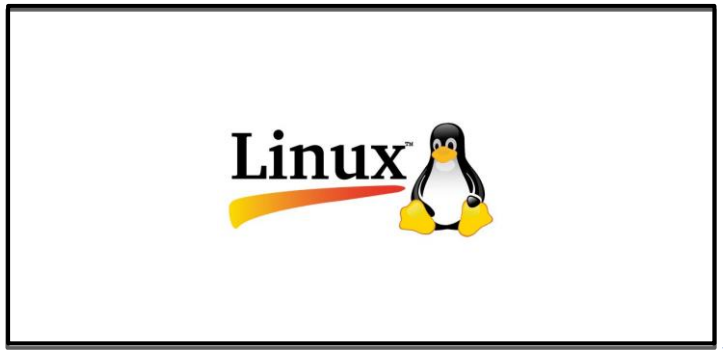
- Scroll up/down [one line / one page]
- Go To begin/end
- Search forward/backward
- q - quit



grep command

Linux Commands

- Search for some data inside files
- grep “Hello” mst.txt
- grep –l “Hello” *
- here * means all files, since it is used as a wildcard
- -i: case insensitive
- -c: count
- -r: recursively



Pipelining |

Linux Commands

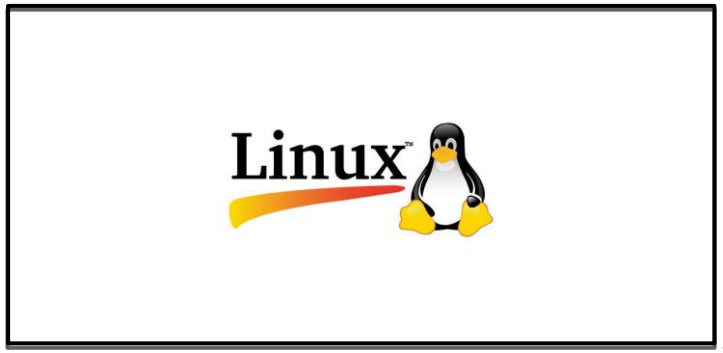
- Used to redirect the output of the current command to the input of the next command

Syntax:

- command1 | command2 | command3

Example:

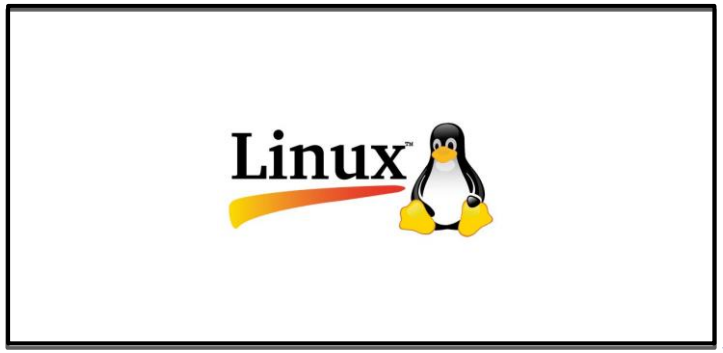
- cat mst.txt | grep “Hello”



The ‘;’ separator

- Can be used in order to run multiple commands in one line
- Example: cat mst1.txt; cat mst2.txt

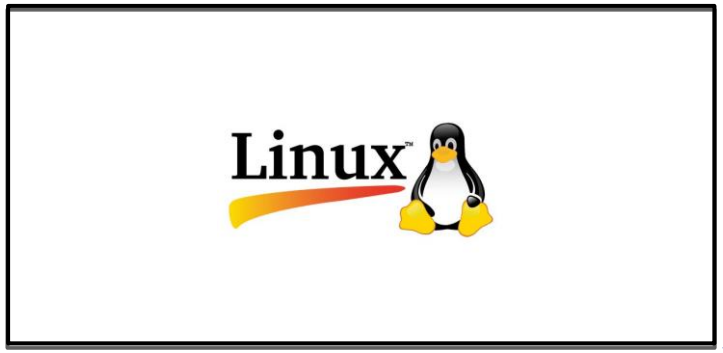
Linux Commands



cp command

- Used to copy files / directories
- cp mst.txt folder/mst2.txt
- -r: recursive

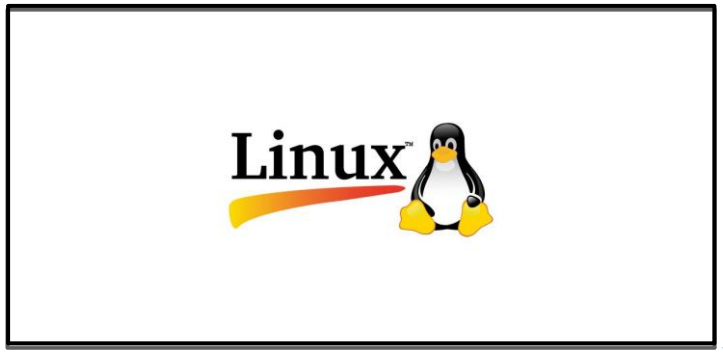
Linux Commands



rm command

- removes files / directories
- rm mst.txt
- -r: recursive

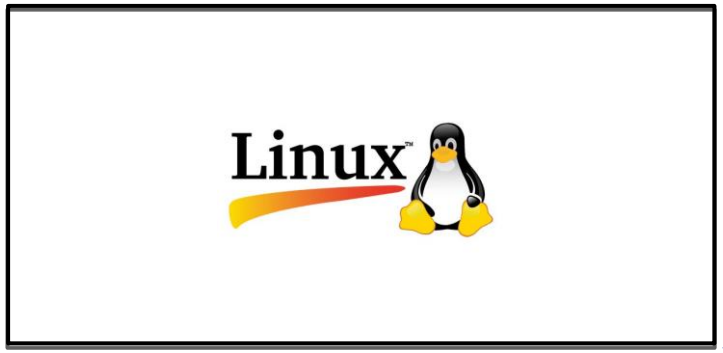
Linux Commands



Process related commands

Linux Commands

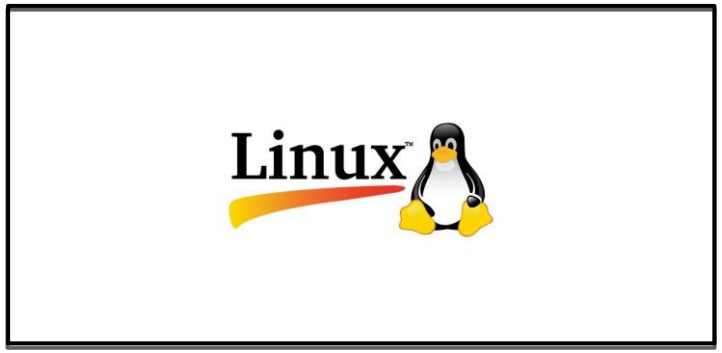
- ps
- top
- kill
- jobs
- fg
- bg



Network & package management commands

Linux Commands

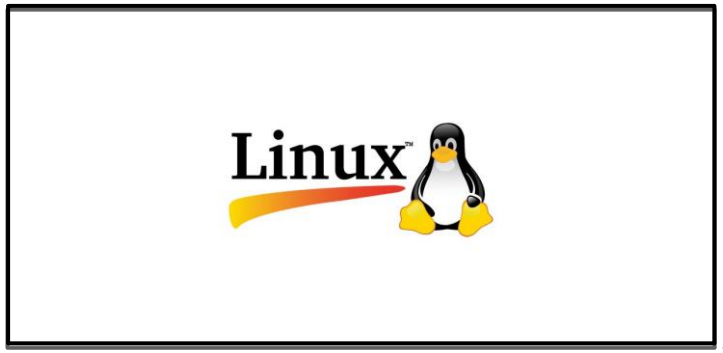
- ping
- curl
- wget
- netstat
- ss
- apt / dnf



vi text editor

Linux Commands

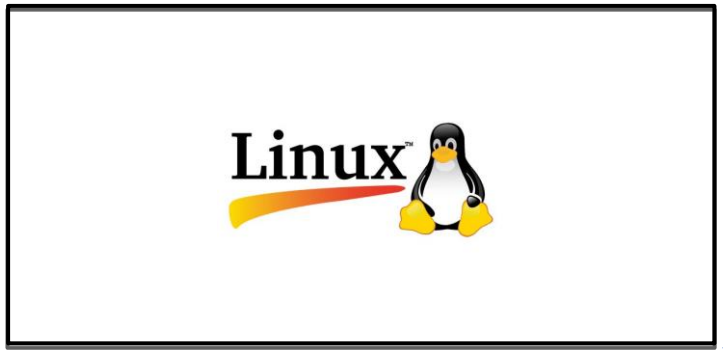
- i – insert
- dd – delete current row
- :q! – quit
- :wq! – write then quit
- / - searches forward in the file
- ? – searches backward in the file



Other commands

- wc –l
- head
- tail
- more

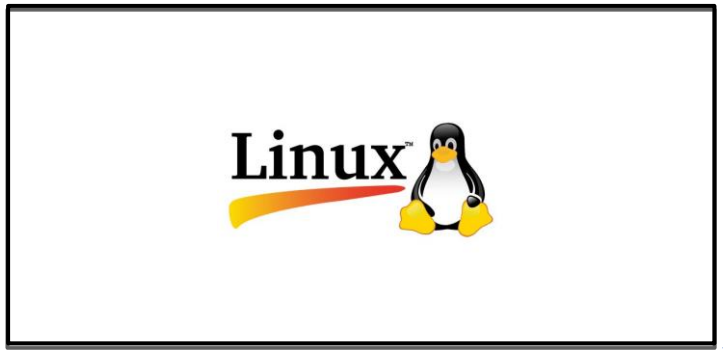
Linux Commands



Bash

BASH

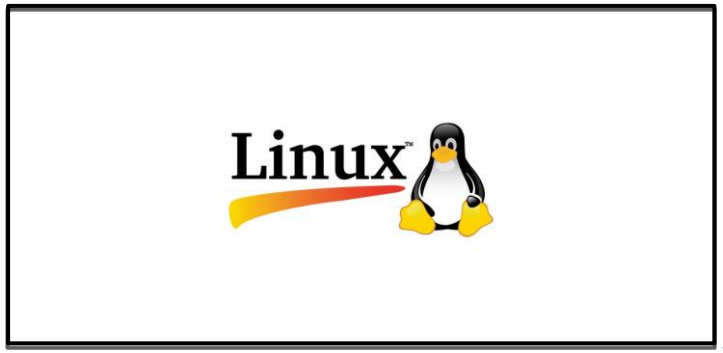
THE BOURNE-AGAIN SHELL



What is Bash?

Bash

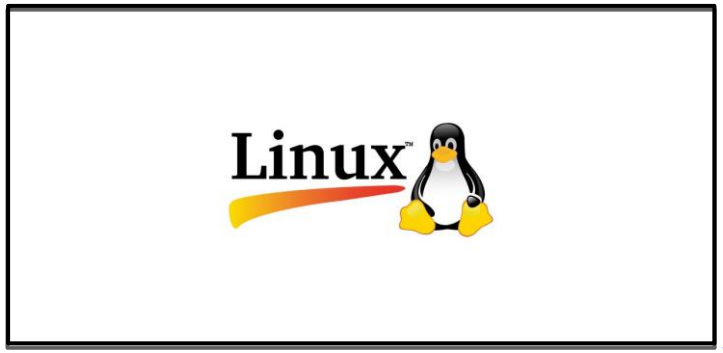
- Short for Bourne Again Shell
- Command-Line Shell & Scripting language
- Widely used in Unix-like OS and macOS
- Default Shell for many Linux distros
- Known for its versatility and Powerful features



Key Features

Bash

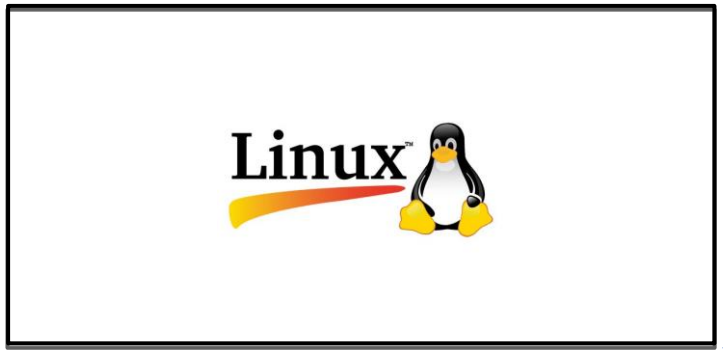
- Command Line Interface
- Scripting Language
- Interactive Features
  - Command History
  - Tab Completion
- Job Control
- Pipes and Redirection |, >, <
- Variables and Control Structures
- Aliases and Functions



Basic Usage

Bash

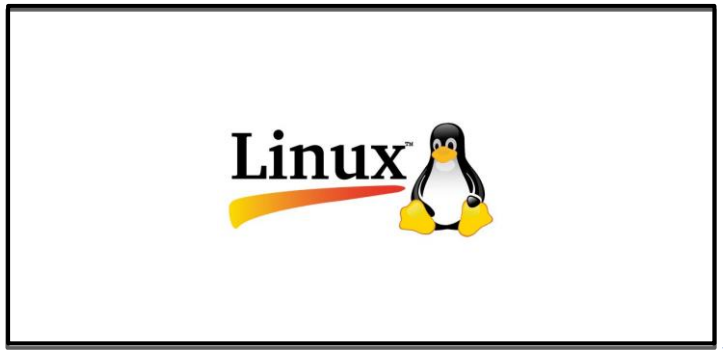
- Create a new file with extension .sh
- Include a shebang (#!/bin/hash) at the top
- Write the commands
- Make the file executable
- Run the file



Key Commands and Concepts

Bash

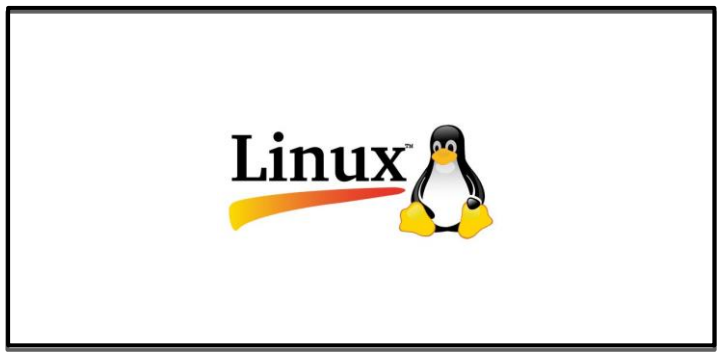
- Viewing Bash Version
- Setting and Using Variables
- Creating Aliases
- Using Job Control
- Scripting (Example)



Script

- Live Example(s)

Bash



Thank You !!