# Introduction to Linux

e-Yantra Team
Embedded Real-Time Systems Lab
Indian Institute of Technology Bombay

September 17, 2017

# Table of contents

# Introduction to UNIX

- UNIX is not an open source project
- UNIX requires specialized hardware resources, hence cannot be installed on every machine
- UNIX is mainly used in server systems and main frames
- In 1983, **Richard Stallman** started the GNU project to create free UNIX like operating system

# Major releases of Operating Systems

- **Linus Benedict Torvalds** developed Linux in the year 1991 when he was an undergraduate student
- Considering the versions of Ubuntu there are around 52 major releases which includes the latest version 17.04
- Windows 1.01 - First major version, released on Nov 20, 1985
- After the first release, there are 25 more major releases for windows, which includes windows 10 also

# Why Linux ?

- Open source Multi-user Operating System
  - User can add new features, implement new ideas. This way Linux gives more flexibility to the users
- Can revive older computers - can be fitted from low end to high end systems
- Supports all major programming languages like C, C++, Python, Java, Perl etc
  - No need to worry about setting environment paths

# Why Linux ?

- Variety of distributions
- Best customer support from Linux forums(`askubuntu.com`)
  - We will be assisted with in minutes after posting a query in these forums.
- No need to reboot frequently
  - No need to reboot after installation, remove, update and upgrade of any application.
- Security
  - Every user is assigned with minimum level of privileges over files of other users.
- Pre-installed with drivers

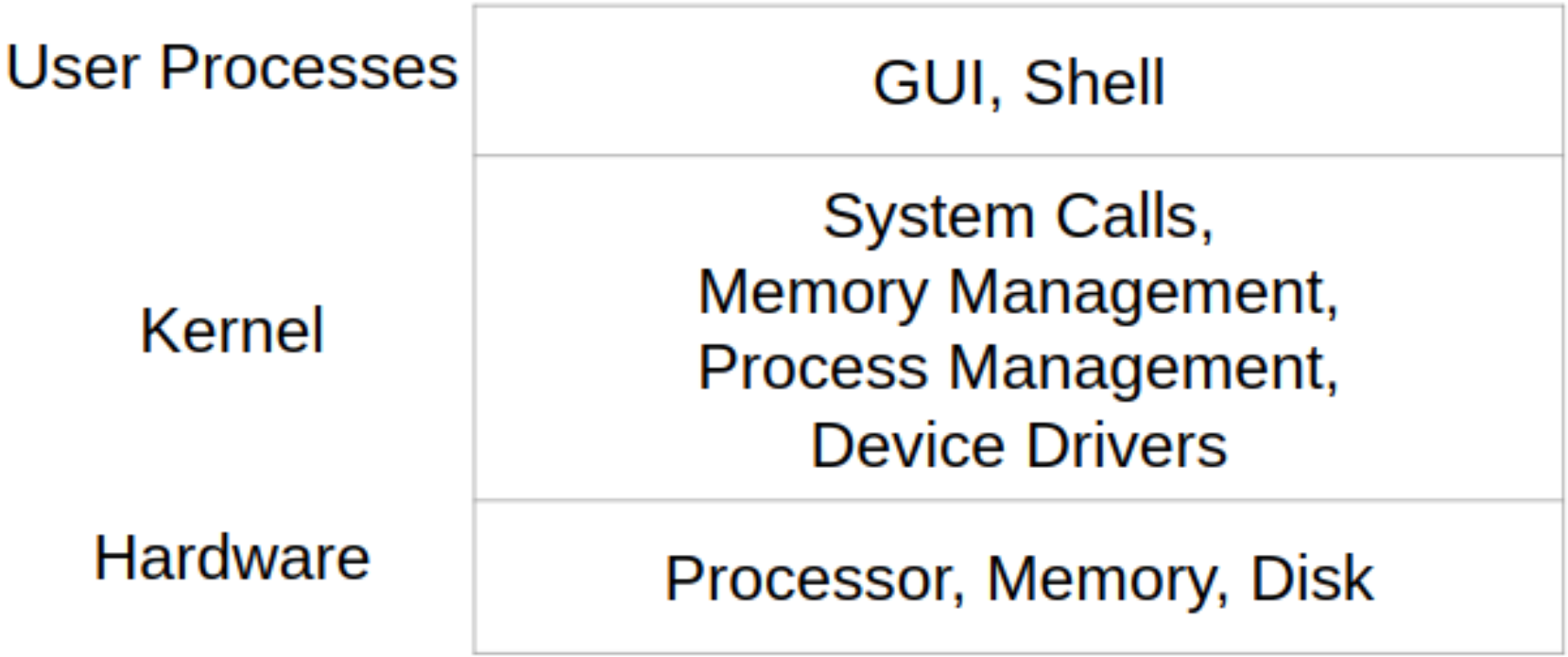
# Operating Systems overview

Operating System is a collection of programs which is used to:

- Manage the hardware resources
  - For example, allocation and deallocation of memory
- Schedule the processes which are in memory
- Synchronization of processes
  - Synchronization between the processes is required for consistency of the data.
- File system Manipulation
  - Create file/directory
  - Read/Write/Execute a file
  - Manage file permissions

and many more

# Levels of Abstraction in Linux



# Process Management

- Process Management describes about start, pause, resume and termination of processes
- Process management also includes scheduling of processes
- Some of the standard scheduling strategies are:
  - FCFS
  - SJF
  - SRT
  - Round Robin
  - Static Priority Based Scheduling
  - Dynamic Priority Based Scheduling

# Memory Management

- **Responsibilities:**
  - Keeps track of which part of memory is being used by whom
  - Deciding which processes memory need to be moved in and out of memory
  - Deciding how much amount of memory needs to be allocated to a particular process
- Each user process will have its own section of memory
- One process may not have the access to private memory of another process

# Device Drivers

- A device is typically accessed in kernel mode i.e., they should run on CPU without any preemption
- Device drivers perform the following operations:
  - Accepts requests from device independent software
  - Interact with the device for I/O and perform required error handling
  - Making sure that execution completed successfully

# System Calls

- System calls perform specific tasks which user process cannot do alone
- For example, opening a file, reading a file, writing to a file etc
- How a process start:
  - fork() - Creates a new process
  - exec() - Starts the process

# Flavours of Linux

Below is the list of Linux flavours most people use:

Ubuntu



Mint

# List of Basic Commands

- **who**
  Displays who is logged on
- **ls -la**
  List the files in the directory with permissions as well as author name
- **mkdir**
  Create directory
- **cd directory_name**
  cd stands for Change Directory. This will make the user to switch the directory

# List of Basic Commands

- **pwd**
  Returns the full path of the current directory
- **touch** < *file_name* >
  Creates an empty file with the mentioned file name
- **rm -r** < *directory_name* >
  Removes the entire directory
- **echo** < *some_text* > » **welcome.txt**
  - Creates a new file welcome if it does not exist
  - Appends the text to the text file welcome.txt

# List of Basic Commands

- **cat -n** $<$ *file_name* $>$
  Dumps the file content on to the terminal with line numbers at left margin

- **wc** $<$ *file_name* $>$
  This command will count the number of lines, number of words and number of characters in the text file

- **less**
  This command dumps the text content of the file onto the screen. The main use of this is to scroll the content.

# List of Basic Commands

- **apt-get**
  This command is used along with one of the options install, remove, upgrade, update, autoremove, clean
- **ifconfig**
  - Stands for Interface Configuration
  - This command will show the ip address(Ethernet, WLAN) as well as the physical(MAC) address of the system
  - Used to view the network configuration of the system
- **man** < *command* >
  Opens the manual page of the command.

# Copy and Move

- **cp** $<$ *source_file_path* $><$ *dest_path* $>$
  Copies file from source and paste into the destination
- **cp -r** $<$ *source_dir_path* $><$ *dest_path* $>$
  Copies directory from source and paste into the destination
- **mv** $<$ *source_file_path* $><$ *dest_path* $>$
  mv command just works like cut and paste. For moving directory use -r option.

# Processes

- **top**
  This command lists the processes which are consuming relatively more resources with their names, IDs and % of memory occupied.

- **ps -A**
  This command lists all the processes.

- **killall** $<$ *process_name* $>$
  This command kills all the threads of the process with the mentioned name

# Grep and Pipe

- **grep -i linux linux_tutorial.txt**
  This command will finds all the lines which contain the linux in the linux_tutorial.txt

- **grep -A 2 -i linux linux_tutorial.txt**
  This command will finds all the lines which contain the linux in the linux_tutorial.txt along with two lines after it.

- **cat demo.txt | grep important**
  This will print all the lines which contain the word important in the text file demo.txt

# Grep and Pipe

- **ps -A | grep chrome**
  This will show all the running threads which contain the word chrome along with the thread IDs.

- We can search for all the patterns on the content which is dumped on to the terminal after running a command can be easily located using pipe symbol along with grep.

# Search in a Directory

- **find** $< path > -name < file\_name >$
  This command returns all the directory paths in which the file exists

- **find** $< path >$ **-name *.jpg**
  This returns all the jpg files in the directory mentioned in the path

- **grep -r abstract ***
  This command will search for the word abstract recursively in all the files in the directory.

# Users and Groups on Linux

- Users
  - Root user or Super user
  - Other users
- Each user will have user ID as well as group ID to which he belongs to
- Set of users can form a group
- Root user can set privileges to individual user separately or user can set privileges to a group of users using group name and group ID

# Create and Delete Users, Groups

- **sudo useradd** $< user\_name >$
  Creates user

- **sudo userdel** $< user\_name >$
  Deletes user

- **sudo groupadd** $< group\_name >$
  Creates group

- **sudo groupdel** $< group\_name >$
  Deletes group

# Create Group and Add Users to Group

- **sudo adduser** $< user\_name >< group\_name >$
  Adds user with the specified name to the group with the specified name.

- Before deleting a group, delete all the users of the group without which it is not possible to delete a group.

- Adding, Deleting of users and groups must be done as a root user.

# File Permissions

- **chmod 754** < *file_name* >
  4 - Read
  2 - Write
  1 - Execute
  0 - No permission
  Each digit is a combination of these numbers. For example 7 is a combination of 4+2+1, 5 is a combination of 4+0+1 and 4 is a combination of 4+0+0.
- **chmod a+wx** < *file_name* >
  Add write and execute permissions to all the users for the file

# File Permissions

- **chmod -R 754** $<$ *file_name* $>$
  Here from the above command sets permission as:
  RWX - For User
  RX - For a users of the group
  R - For other users

# SSH and SCP

- **ssh username@ip_address**
  Used to remote login to a system. After running this command enter the password of the remote system.

- **scp -r** $<$ *local_directory_path* $><$ *dest_user_name* $>$ **@** $<$ *dest_ipaddress* $>:<$ *dest_path* $>$
  This command is used to securely copy the local directory into remote location with the specified user name and ip address

# Archive

- **tar -cvf example.tar example**
  This creates the tar file for the directory example and stores it in the current working directory.

- **tar -xvf example.tar**
  This will untar the file example.tar and stores it in the current working directory

# Archive

- **tar -cvzf images.zip images**
  This will zip the images directory and saves in format images.zip for the directory images

- **tar -xvf images.zip**
  This command will unzip and creates the directory images.
  -c : creates an archive
  -v : verbose
  -f : Allows us to specify file name of the archive
  -x extract files from archive

# Vim Introduction

- Vim is an open source command line editor
- User may not always have access to GUI editors
- For low end flavours of Linux, we don't have access to GUI
- Vim has the powerful features for text navigation
- Vim comes along with tutorial. Just run the command **vimtutor** on your terminal which will show all the commands along with their description
- According to a survey conducted by stackoverflow, 30%(of 55,000 approx) of software employees still use Vim as their primary editor

# Modes

- **Insert Mode**
  - To insert the text
- **Normal Mode**
  - To easily navigate between the text
- In Insert Mode, press Esc to switch to Normal mode
- In Normal Mode, press i to switch to Insert Mode

# Navigate Between the Text

- Keys **h, j, k, l** to move the cursor left, down, up and right respectively.
- Key **w**
  - Moves the cursor to start of next word
- Key **b**
  - Moves the cursor to beginning of the previous word
- Key **e**
  - Moves the cursor to end of the next word
- Key **3w**
  - Moves the cursor to start of 3rd word from current cursor position. Similarly for keys 3b, and 3e also

# Insert Text Multiple Times

- **10i- + Esc**
  - 10 hiphens with just one command
- **10i% + Esc**
  - 10% symbols with just one command
- This way Vim saves lot of time as compared to other text editors

# $N_{th}$ Occurrence of a Character

- **fa**
  - First occurrence of the character from the current cursor position
- **5fa**
  - Fifth occurrence of the character from the current cursor position
- **Shift + %**
  - Jumps to matching parenthesis
- **0**
  - Jumps to beginning of the current line
- **$**
  - Jumps to end of the current line

# $N_{th}$ Occurrence of a Word

- *
  - First occurrence of the word from the current cursor position
- **5***
  - Fifth occurrence of the word from the current cursor position
- #
  - Immediate previous occurrence of the word from the current cursor position
- **5#**
  - 5th previous occurrence of the word from the current cursor position

# Navigate to a Specific Line

- **gg**
  - Move the cursor to beginning of the file
- **G**
  - Move to cursor to end of the file
- **2G**
  - Moves the cursor to the 2nd line of the file

# Search for a Word

- /**between**
  - Search for the word between from the current cursor position
- For further continuation of search for the same word use the key n
- For searching the same word in other direction use the key N

# Insert a New Line

- o
  - Inserts new line above the current line
- O
  - Inserts new line below the current line
- After entering the above key, mode immediately shifts to insert mode

# Delete Characters, Words and Lines

- **x**
  - Deletes the character under the cursor
- **X**
  - Deletes the immediate previous character
- **r**
  - Replace a character under the cursor
- **dw**
  - Deletes the word under the cursor from the current cursor position(may not be complete word)
- **dd**
  - Deletes the current line
- (.)
  - command is used to execute the previously executed command

# Copy and Paste

- **yy**
  - Copies the current line
- **5yy**
  - Copies 5 lines from the current cursor position
- **p**
  - Paste below the current line
- **P**
  - Paste above the current line

# Visual Mode

- **v**
  - Switch to visual mode
  - This mode enables user to perform operations on the selected text
- **ved**
  - Selects the text upto the end of the word and deletes the text
- **v$d**
  - Selects the text upto the end of the current line and deletes it
- **v0d**
  - Selects the text from the start of the current line to the current cursor position and deletes it

# Save and Quit

- The :**w** command is used to save the file
- **:q** is used to quit from Vim
- **:q!** Is used to quit without saving
- **u** is used to undo and Ctrl+R is used to redo