

- What is Ubuntu?
 - What is Ubuntu?
 - A little bit of Ubuntu's background
 - The Ubuntu vision
 - Ubuntu flavors
 - Ubuntu has 7 official flavors:
 - Some Ubuntu distributions are:
- Notes Lecture 02 | Installing Ubuntu
 - What is Virtualization?
 - Server-side virtualization
 - Virtual Desktop Infrastructure(VDI)
 - Client-side virtualization
 - Benefits of Virtualization
 - Installing Ubuntu in a virtual machine
- Notes Lecture 03 | Learning the Bash Shell
 - Desktop Environments
 - GUI
 - KDE DE
 - XFCE DE
 - Mate DE
 - Cinnamon DE
 - What is a Shell
 - The Bash Shell
 - Managing Software
 - Debian Package Management System
 - The Linux Filesystem
- Notes Lecture 04 | Manipulating files and directories
 - Creating directories
 - Creating files
 - Deleting files and directories
 - rm command
 - Moving files and directories
 - The mv command
 - Copying files and directories
 - The cp command
 - Getting Help
 - Working with Wildcards
 - The * Wildcard
 - The ? Wildcard
 -  Wildcard
 - Brace Expansion
- Notes Lecture 05 | Command Line Text Editors
 - The basics of Nano
 - Opening and Creating files
 - Saving and exiting
 - Cutting and pasting

- Searching for text
- The basics of VIM
 - How to start and quit vim
 - Vim modes
 - Editing a file with vim
 - Searching words in vim
 - Moving to Lines
- Notes Lecture 06 | Managing Data and File permissions
 - Managing Data
 - Archiving utilities
 - File Compression
 - Useful commands
 - Zip, 7Zip, and rar
 - File Permissions
 - Files vs Directories
 - Files
 - chmod command
- Managing Users and groups
 - Managing User Accounts
 - Creating a user with useradd
 - Viewing a user's account info
 - Maintaining Passwords
 - Deleting User Accounts
 - Managing Groups
 - Delete me
 - Querying Users
- Shell Scripting
 - Creating a basic script
 - Displaying text
 - Working with variables
 - Exit Status Codes
 - Using Structured Commands
 - Conditions



What is Ubuntu?

What is Ubuntu?

Ubuntu is a complete Linux operating system, freely available with both community and professional support. Ubuntu is suitable for both desktop and server use. The current Ubuntu release supports Intel x86 (IBM-compatible PC), AMD64 (x86-64), ARMv7, ARMv8 (ARM64), IBM POWER8/POWER9 (ppc64el), IBM Z zEC12/zEC13/z14 and IBM LinuxONE Rockhopper I+II/Emporer I+II (s390x). Ubuntu includes thousands of pieces of software, starting with the Linux kernel version 5.4 and GNOME 3.28, and covering every standard desktop application from word processing and spreadsheet applications to internet access applications, web server software, email software, programming languages and tools and of course several games.

A little bit of Ubuntu's background

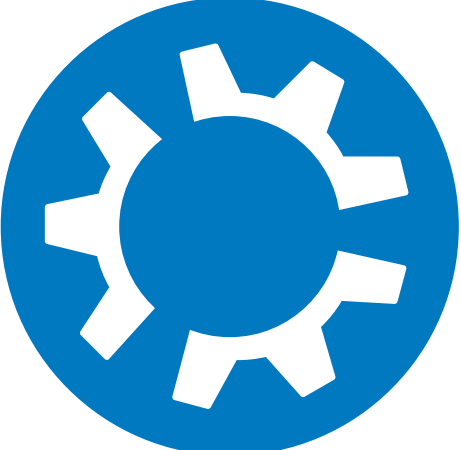
The Ubuntu vision

- In April 2004 Mark Shuttleworth brought together a group of developers from the Debian project, GNOME, and GNU. The goal of the meeting was to answer the following questions:
 - Is a better type of operating system possible?
 - What would this OS look like?
 - Can you describe the community that would build this project? The team called itself the Warthogs and set on to build the Ubuntu operating system.
- Before Ubuntu, installing Linux was a project for computer enthusiasts. Linux on the desktop was considered a computer nerd only operating system because it was difficult to install and maintain for a regular computer user.
- Ubuntu 4.10 (Warty Warthog), released on 20 October 2004. It was Canonical's first release of Ubuntu and it looked like this:  Ubuntu 6 Desktop *Ubuntu 6.06 (Dapper Drake), released on 1 June 2006, was Canonical's fourth release and the first long-term support (LTS) release. It looked like this:  Ubuntu 6 Desktop



Ubuntu flavors

An ubuntu flavor is an operating system based on ubuntu that uses a different desktop environment than the default Ubuntu's desktop environment (GNOME). Ubuntu flavors offer a unique way to experience Ubuntu, each with their own choice of default applications and settings. Ubuntu flavors are backed by the full Ubuntu archive for packages and updates.

Ubuntu has 7 official flavors:

logo	Flavor	Download Link
	Kubuntu	https://kubuntu.org/getkubuntu/

logo	Flavor	Download Link
	Lubuntu	https://lubuntu.net/downloads/
	Ubuntu Budgie	https://ubuntubudgie.org/downloads
	Ubuntu Kylin	https://www.ubuntukylin.com/downloads/show.php?id=451&lang=en
	Ubuntu Mate	https://ubuntu-mate.org/download/

logo	Flavor	Download Link
	Ubuntu Studio	https://ubuntustudio.org/download/
	Xubuntu	https://xubuntu.org/download/

Ubuntu also serves as a base for a large number of Linux distributions. Some people call those Linux distributions Ubuntu distributions because they heavily rely on Ubuntu's active development cycle.

Some Ubuntu distributions are:

- Linux Mint
- Pop!_Os
- elementaryOS
- Zorin OS
- Linux Lite
- Peppermint OS
- BackBox Linux

Notes Lecture 02 | Installing Ubuntu

What is Virtualization?

- Replication of hardware to simulate a virtual machine inside a physical machine.
- Two general types of virtualization:
 - server-side virtualization
 - client-side virtualization
- The difference between the two is where the virtualizing takes place

Server-side virtualization

Virtual Desktop Infrastructure(VDI)

- Thick client or fat client
- Think client

- Zero client

Client-side virtualization

- Software installed on a computer to manage virtual machines
- Each VM has its own operating system installed
- For client-side virtualization, the computer needs:
 - A hypervisor (software that allows the management of virtual machines)
 - Hardware support
- Capable CPU
- Enough RAM
- Enough Storage

Benefits of Virtualization

- Allows running multiple OSs on one machine
- Allows applications to be tested before installing them on a host machine
- Reduces cost by decreasing the physical hardware that must be purchased for a network
- Offers the chance to experiment with untested programs without infecting host machines with viruses or other malware

Installing Ubuntu in a virtual machine

- Download the iso file
- Click on new and enter a name for the virtual machine
- Select an OS type
- Select the desired memory size
- Select storage type
- Select file location and size
- Start!

Notes Lecture 03 | Learning the Bash Shell

Desktop Environments

GUI

- A graphical user interface is a set of programs that allows a user to interact with the computer system via icons, windows, and various other visual elements. ###DE
- A desktop environment is an implementation of the desktop of the desktop metaphor made of a bundle of programs running on top of a computer operating system, which shares a common GUI, sometimes described as a graphical shell. ##GNOME DE
- Default desktop in Ubuntu is GNOME 3. It is used not only by Ubuntu but also several other linux distributions such as, Debian, Fedora, Red Hat, Enterprise Linux, and Oracle Linux. The official GUI for GNOME 3 is called GNOME Shell.
- GNOME was an acronym for HNU Network Object Model Environment, but the acronym was dropped because it no longer reflected the vision of the GNOME project.

- GNOME was started August 15,1997, by Miguel de Icaza and Federico Mena as a free software project to develop a desktop environment and applications for it.

KDE DE

- The Kool Desktop environment got its start in 1996, with its first version released in 1998
- Through time the name KDE was no longer just referring to a desktop environment, but insread, it specified the project's organization and the strong community that supported it.

XFCE DE

- XFCE is a lightweight DE that aims to be fast and low on system resources, while still being visually appealing and user friendly
- The XFCE project was started by Oliver Fourdan in 1996.

Mate DE

- The Mate DE is the continuation of GNOME 2.
- Mate has forked applications from GNOME core
- Mate Applications include:
 - File manager
 - Text editor
 - Document viewer
 - Archive manager
 - Terminal emulator
 - Window manager

Cinnamon DE

- Cinnamon is a free and open-source DE for the X Windows System that drives from GNOME 3 but follows traditional desktop metaphor conventions.
- Cinnamon is the principal DE of the Linux Mint distribution
- The development of Cinnamon started as a response to the release GNOME 3 and the GNOME Foundation decision of dropping support for GNOME 2

What is a Shell

Shells make large-scale IT possible. They're a necessary component to modern computing. But it might not have turned out that way without a lot of hard work from a developer at the Free Software Foundation named Brian Fox. The Bash shell is shipped with almost every computer in the world.

The Bash Shell

- The GNU bash shell is a program that provides interactive access to the Linux system.
- It runs as a regular program and is normally started whenever a user logs in into a terminal.
- Most Linux distributions use the bash shell as the default shell. however, other shells exist like
 - Tcsh Shell
 - Csh Shell
 - Ksh Shell

- Zsh Shell
- Fish Shell

Managing Software

Debian Package Management System

- The DPMS is the foundation for managing software on all Debian distributions
- Debian package names end with .deb extension and are called ".deb files"
- At the core of the DPMS is the dpkg (Debian Package) application
- Dpkg works in the back-end of the system. dpkg can directly manipulate .deb files.

The Linux Filesystem

- Linux organizes its files in what is called a hierarchical directory structure (tree like pattern of folders).
- Directory and folder mean the same thing
- the first directory in the file system is called the root directory. the root directory contains files and subdirectories.
- Filesystem Hierarchy Standard (FHS) Specifies requirements and guidelines for file and directory placement in UNIX-like operating systems.

Notes Lecture 04 | Manipulating files and directories

Creating directories

- mkdir is used for creating a single directory or multiple directories
- to create multiple directories, separate each directory name with a space
- You can create directories in the present working directory or in a different directory by using an absolute path or relative path

Creating files

- touch command is used for creating files
- Create a file called list - touch list

Deleting files and directories

rm command

- rm removes files
- rm by default does not remove directories. To remove a directory use rm with -r option
- In linux and other Nix systems you cannot remove non empty directories
- To remove empty directories use the rmdir command
- To remove non-empty directories use rm -r + directory name or directory absolute path

Moving files and directories

The mv command

- mv moves and renames directories
- The basic formula of the mv command is:
 - mv + source + destination
- Where source is the file or directory that you want to move and destination is where the directory or file is going
- Both source and destination can be an absolute path or relative path

Copying files and directories

The cp command

- cp copies files/directories from source to a destination
- The cp command uses the same structure as the mv command
 - cp + files to copy + destination
- To copy directories you must use the -r option
 - cp -r + directory to copy + destination

Getting Help

- Man pages are documentation files that describe Linux shell commands, executable programs, system calls, special files, and so forth
- Man Pages are quick references
- To view a man page type: man + command
- To exit the man page press letter "q"
- Most commands have a help option built in. Normally that option is -h or --h or --help
- Some commands may not have a man page but an info page
- You can use the whatis command to display a simple description of what a command does

Working with Wildcards

- Wildcard represent letters and characters used to specify a filename for searches
- File globbing is the processing of pattern matching using wildcards
- The wildcards are officially called metacharater wildcards

The * Wildcard

- The main wildcard is a star, or asterisk (*) character
- A star alone matches anything and nothing and matches any number of characters

The ? Wildcard

- The ? wildcard metacharacter matches precisely one character
- In addition, the question mark proves very useful when working with hidden files
- If you want to list all hidden files you can use: ls ..??* which will match all files that start with a . or .. and have any character after it.

☐ Wildcard

- The brackets wildcard match a single character in a range
- The brackets wildcard use the exclamation mark to reverse the match. For example, match everything except the vowels [!aeiou] or any other character except numbers [!0-9]

Brace Expansion

- Brace expansion {} is not a wildcard but another feature of bash that allows you to generate arbitrary strings to use with commands

Notes Lecture 05 | Command Line Text Editors

The basics of Nano

Opening and Creating files

- For opening and creating files
 - nano + filename

Saving and exiting

- If you want to save the changes you've made ctrl + O
- To exit nano, type ctrl + X
- If you ask nano to exit from a modified file, it will ask you if you want to save it, just press N for no and Y for yes. It will then ask for a filename and press enter

Cutting and pasting

- To cut a single line, you can use ctrl + k the line disappears
- To paste it, use ctrl + U

Searching for text

- Searching for string is as easy as long as you think "Whereis" instead of "Search" Simply hit ctrl + W type in your string and press enter

The basics of VIM

- The vi command is now linked to the vim command, so even when you issue the vi command, you're actually starting the vim editor
- To install vim
 - sudo apt install vim

How to start and quit vim

- To start vim type vim
- To quit vim press esc and type :qa!
 - : prefix for entering command line mode
 - q short for quit
 - a short for all buffers

- ! force
- :qa! quit all now
- To set line numbers :set number

Vim modes

- Insert mode: used for writing texts
- Normal mode: used for manipulating text
- Command mode: used for entering vim commands
- Visual mode: used for navigation and manipulation of text selections
- Select mode: similar to visual mode
- Ex-mode: Similar to the command-line mode but optimized for batch processing

Editing a file with vim

- You can tell vim that you want to edit another file by using the e command
- :e new.txt will now open new.txt and allow you to edit
- ctrl + g will show the file that you are currently editing in the status line
- You can also use :f in command mode to see the file that you are currently working on

Searching words in vim

- Use / and the word you are looking for to search forward
- Letter n will repeat the search for the next word
- ? To search backward
- *will search for the next occurrence of the word under the cursor
- #will search backward for the previous occurrence of the word under the cursor

Moving to Lines

- To move to a specific line use : plus the line number
 - :8 will move you to line 8
 - Additionally use 8G
- \$ will move to the end of the line
- 0 will move to the beginning of the line
- ◦ executes any vim command from the shell prompt

Notes Lecture 06 | Managing Data and File permissions

Managing Data

- Backup- Copies files and directories to an archive
- System backup- Use to restore data in case of a system failure or data loss and corruption
- Archive- File containing many other files, each of which is still identified by its filename, owner, permissions, and timestamp

Archiving utilities

- Tar(Tape archive)- creates archives by combining files and directories into a single file
- CPIO- Creates an archive, restores files from an archive, or copies a directory hierarchy. The CPIO utility has three modes of operation:
 - Create mode places multiple files into a single archive file
 - Extract mode restores files from an archive
 - Pass-through mode copies a directory hierarchy
- Ar- Creates, modifies, and extracts from archives

File Compression

- Gzip, Bzip2, and xz commands are used for compression
- Gzip, Bzip2, and xz compress files in place meaning the original file is deleted after compression
- bzip offers better compression ratios in comparison to gzip
- xz produces better compression ratios than gzip and bzip2

Useful commands

- gunzip= gzip -d
- bunzip= bzip -d
- unxz= xz -d

Zip, 7Zip, and rar

- Zip is an archiving and compression utility
- 7zip is an open source, cross-platform and fully-featured file archiver with a high compression ratio
- To use 7zip on linux you need the package: p7zip-full
- RAR is a proprietary archive file format developed by Eugene Roshal. The command unrar allows Linux users to extract rar archives.

File Permissions

- A file can be owned only by one user and one group
- ls -l shows you the file user owner and group owner
- The /etc/passwd file contains a list of all the users in Linux
- The /etc/group file contains a list of all the groups in Linux
- The chown command is used for changing group owner

Files vs Directories

Files

- R(read)
 - Gives users permission to open and view a files contents
- W(write)
 - Gives users permission to open a file and edit it contents
- X(execute)
 - Allows users to run the file
- Directories
 - R(read)

- Allows users to list a directory's contents with commands such as ls
- W(write)
 - Allows users to add or remove files and subdirectories
- X(execute)
 - Allows users to switch to the directory with the cd command

chmod command

- The chmod command is used to change permissions on files and directories
- The permissions argument is the information used to change permissions
- The file/directory argument specifies the file or directory you want to change
- You can use the chmod command in two ways to change file permissions:
 - Symbolic Notation
 - Numeric Notation

Managing Users and groups

Managing User Accounts

- Managing user accounts involves adding, modifying and deleting user accounts and account's information
- Add user accounts use useradd or adduser
- modify use usermod
- delete use userdel
- make sure to use sudo
- The /etc/login.defs file contains directives for use in various shadow password suite commands
- The /etc/default/useradd file stores the system default configuration for creating new users with the useradd utility
- to view the default parameters use either
 - useradd -D
 - cat /etc/default/useradd
- The /etc/passwd file stores information about every user account in a Linux system
- The /etc/shadow file contains information about the users' passwords

Creating a user with useradd

- The useradd utility, in Ubuntu, is considered a low-level utility and should not be used. This utility is the standard in most linux distros. The adduser utility uses the useradd utility in the "backend".
- -md options needed for adding a home directory to a user
- /home/student is the new users home directory
- -s used for specifying the users login shell
- /bin/bash is the new users login shell
- students is the users username

Viewing a user's account info

- The grep command is used to see the students user information - grep student /etc/passwd/

- The second one is used to see password information about the new user. ! means there is no password set - `sudo grep student etc/shadow/`
- The third shows the content of the students user home directory - `ls -A /home/student/`
- the fourth shows the content of the skel directory which must be the same as the new user's home directory - `sudo ls -A /etc/skel/`
- The getent utility can be used to view information about a user's account and password - `getent passwd student - sudo getent shadow student`

Maintaining Passwords

- The useradd does not create a password for the users, the passwd utility does
- To change password of another user - `passwd + username`
- to change password of the current user - `passwd` with no argument
- The passwd utility can also lock and unlock account with the -l and -u options
- -d delete
- -e expire
- -i inactive
- -l lock
- -n minimum
- -s status
- -u unlock
- -w warning or warndays
- -x maximum or maxdays
- Chage is also another utility to work with passwords
 - It can modify and display user information

Deleting User Accounts

- userdel utility and the most common option to use is -r
- This option will delete the accounts home directory tree and any files within it
- Delete the sampleuser account `sudo userdel -r sampleuser`

Managing Groups

- `sudo group user` shows the group name that the user belongs to.
- to create a new group - `sudo groupadd groupname`
- To add a user to a new group add, the group must preexist - `sudo getent group groupname`
- `sudo usermod -aG groupname user` allows you to add the DAdams account as a member of the project42 group but the -a switch is important because it preserves only previous DAdams account group memberships

Delete me

- Groups can also have passwords but is not a good idea
- The `/etc/shadow/` file is where group passwords are stored
- If you need to modify a group use `groupmod` - `sudo groupmod` then whatever variable you need.

Querying Users

- Whoami displays what user account you are currently using
- Who provides a little more data than whoami utility. You can view information concerning your own account or look at every current user on the system.
- W shows who is logged on and what they are doing
- Id utility allows you to pull out various data concerning the current process. it also displays information for any account whose identification you pass to id as an argument
- Last pulls information from the /var/log/wtmp file and displays a list of accounts showing the last time they logged in/out the system or if they are still logged on. It also shows when the system reboot occurs and when the wtmp file was started

Shell Scripting

- Source code vs machine code
 - Source code is the human readable code written in a programming language like python
 - Machine code consists of binary 1s and 0s and is the language a computer's CPU understands
- Compiler: a program used for converting a complete source code into machine code
- Interpreter: a program that converts source code into machine code line by line
- Compiled program vs Interpreted program
 - a compiled is a binary file produced by a compiler
 - an interpreted program is a program that requires an interpreter to interpret and run each instruction
- Shell scripts are examples of interpreted programs, and the interpreter used is the BASH shell

Creating a basic script

- Start vim, enable line numbers, and enter insert mode
- type:
 - `#!/bin/bash`
 - `echo "this is a script that displays information about your linux system"`
 - `uname -a`
- Save the file and name it script1.sh
- Type `chmod u+x script1.sh` to make the file executable
- to run the script type : `./script1.sh`

Displaying text

- To display a line of text use the echo command
 - Ex. `echo "this is a message"`
- `echo -n` does not output a new line

Working with variables

- Variable: placeholder for data
- Environment variable: is the placeholder for data that can change: typically, it gets its value automatically from the OS startup or the shell being used.
- The HOME environment variable stores the absolute pathname to a user's home directory, so it varies for each user

- The env command allows you to see all environment variables
- You can also use the echo command to see the value of each environment variable
 - echo \$HOME
 - echo \$HOST

Exit Status Codes

- Exit Status code: a number sent to the shell when you run a command
- Successful commands usually return the code 0, and failures return a value greater than 0

Using Structured Commands

Conditions

- The if statement is used to carry out certain commands based on testing a condition and the exit status of the command
- if-statement- Starts the condition being tested
- then statement- Starts the portion of code specifying what to do if the condition evaluates to true
- else statement- Starts the portion of code specifying what to do if the condition evaluates to false.
- The fi statement indicates the end of the if statement

Table 5-7 File attribute operators in the BASH shell

File attribute operator	Description
-a	Checks whether the file exists
-d	Checks whether the file is a directory
-f	Checks whether the file is a regular file
-r	Checks whether the user has read permission for the file
-s	Checks whether the file contains data
-w	Checks whether the user has write permission for the file
-x	Checks whether the user has execute permission for the file
-O	Checks whether the user is the owner of the file
-G	Checks whether the user belongs to the group owner of the file
file1 -nt file2	Checks whether file1 is newer than file2
file1 -ot file2	Checks whether file1 is older than file2

Numeric Comparison

Comparison	Description	Example
<code>n1 -eq n2</code>	Checks if n1 is equal to n2	If [\$n1 -eq \$n2]
<code>n1 -ge n2</code>	Checks if n1 is greater than or equal to n2	If [\$n1 -ge \$n2]
<code>n1 -gt n2</code>	Checks if n1 is greater than n2	If [\$n1 -gt \$n2]
<code>n1 -le n2</code>	Checks if n1 is less than or equal to n2	If [\$n1 -le \$n2]
<code>n1 -lt n2</code>	Checks if n1 is less than n2	If [\$n1 -lt \$n2]
<code>n1 -ne n2</code>	Checks if n1 is not equal to n2	If [\$n1 -ne \$n2]

String Comparison

Comparison	Description	Example
<code>str1 = str2</code>	Checks if str1 is the same as string str2	If [\$str1 = \$str2]
<code>str1 != str2</code>	Checks if str1 is not the same as str2	If [\$str1 != \$str2]
<code>str1 < str2</code>	Checks if str1 is less than str2	If [\$str1 < \$str2]
<code>str1 \> str2</code>	Checks if str1 is greater than str2	If [\$str1 > \$str2]
<code>-n str1</code>	Checks if str1 has a length greater than zero	If [\$str1 -n]
<code>-z str1</code>	Checks if str1 has a length of zero	If [\$str1 -z]