CSCI 274 - Intro to Linux OS

Mona Wade (dwade@mines.edu)

Overview

1. Syllabus
2. What is Linux?
3. What is bash?
4. Linux “options”
5. Choose an Editor

INTRO LINUX OS, Sec: A Sprg 20

Jump to Today

CSCI 274 : Intro to Linux OS - Sections A, B, & C - Spring 2020

Course Syllabus

Instructor: Demonna Evelyn Wade

Email: dwade@mymail.mines.edu

Lecture Times:

- Section A - M at 2:00 - 2:50 PM MST - Location: MZ 022
- Section B - W at 1:00 - 1:50 PM MST - Location: MZ 022
- Section C - F at 11:00 - 11:50 AM MST - Location: MZ 022

Office Hours: MEP (1700 Maple St.) -> TODO

Teaching Assistant:

TBA

Course Description :
Introduction to the Linux Operating System will teach students how to become proficient with using a Linux operating system from the command line.

Topics :

- Remote Login (ssh)
- File System Navigation
- File Manipulation Commands
- Editors
- I/O Redirection
- Search and Replace
- Processes
- Pipelines
- File System Interrogation Commands
- Authentication
- Permissions
- Compression
- Privacy
- Bash Scripting

What is Linux?

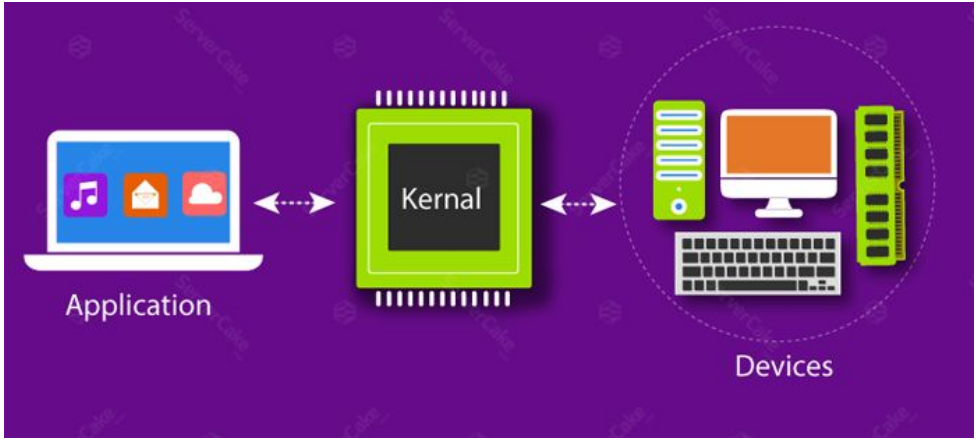
Linus Torvalds created Linux when he was a student at the University of Helsinki in the early 1990s.

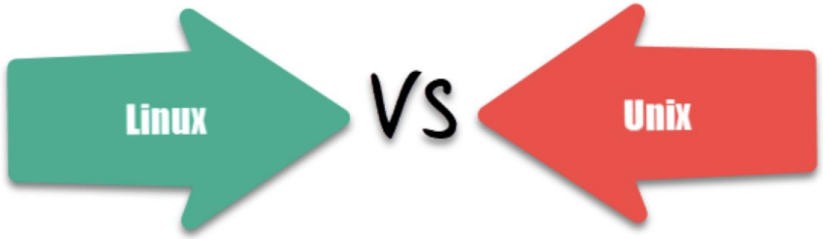
It is not a UNIX-derivative (it was written from scratch). Many of the commands found in Linux are also found in UNIX. UNIX was created by AT&T Bell Labs in the late 1960s.

What is Linux?

A Linux distribution is the Linux kernel and a collection of software together, creating an operating system.

A kernel is the layer between the application and hardware.





Linux	UNIX
Is open source meaning thousands of programmers collaborate online and contribute to its development	Commercial vendors have developed different versions
Can be installed on various types of devices like mobile, tablets, computers, etc.	UNIX is primarily used in internet servers, workstations, and PCs.
Easily portable and can be booted from a USB	Not as portable
Source code is available to the general public	Source code is not available to anyone
Can easily co-exist along with other Operating Systems	Designed for a slow computer system

Common Linux Distributions

What is a bash?

The “Bourne Again Shell” (a derivative of the popular Bourne Shell)

A shell’s job is to interface with the Operating System.

A command language with a syntax derived from natural human language.

Features:

Command execution	Functions	Job control
Conditional logic	String manipulation	Maintenance
arrays/dictionaries	expansion	Sysadmin tasks
iteration/looping	Batch processing	Scripting

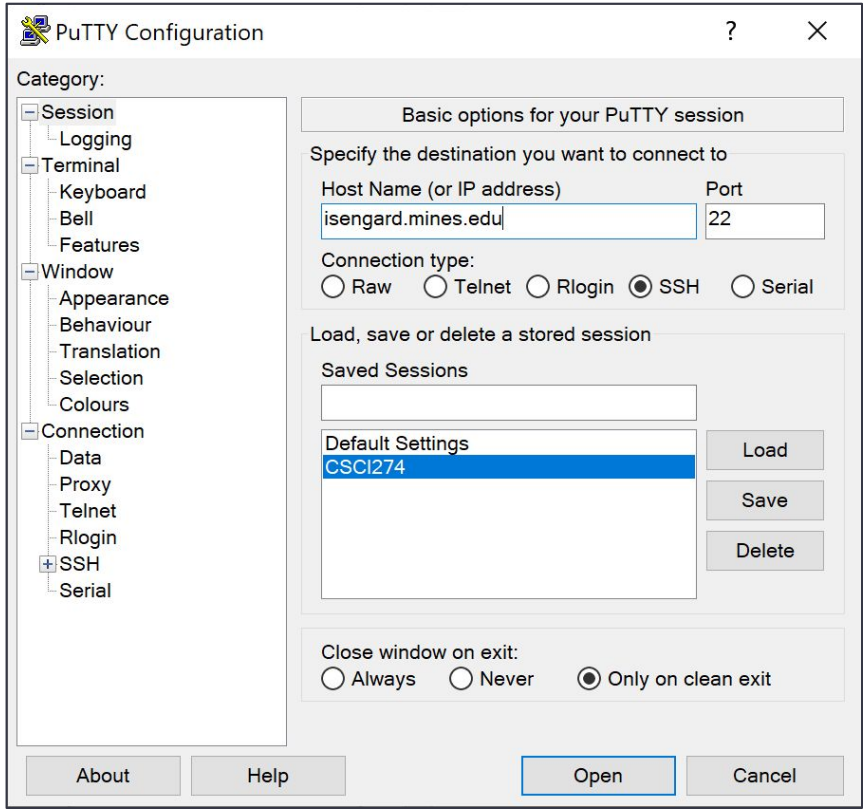
Other Shell Programs

- ksh (KornSHell) - developed by David Korn at AT7T Bell Laboratories in the early 1980s. It is backwards-compatible with the Bourne shell and includes many features of the C shell.
- tcsh (“tee-see-shell”) - based on and compatible with the C shell (csh). It expands on csh with command-line completion, command-line editing, and a few other features. It is backwards compatible with csh.
- Zsh (Z shell) - used as an interactive login shell and as a command interpreter for shell scripting. It extends bash and includes features from ksh and tcsh.

Linux “options”

- **PuTTY**
- **Bash.exe for Windows PowerShell**
- **Windows Subsystem for Linux**
- **Virtual Machine**

PuTTY



Is an SSH and telnet client. Can be installed on most operating systems.

Opens a terminal on isengard.mines.edu server. Login via campus wide username and password.

Terminal emulator is a program that opens a window and lets you interact with the shell.

Nano vs. Emacs vs. Vim

Nano Pros:

- No learning curve.
- Easy to use. Idiot proof.
- Good for simple edits.

Nano Cons:

- Making complicated edits can be difficult and time-consuming.
- No powerful features such as macros, editing multiple files at once, window splitting, vertical block/rectangle selecting/editing, auto-completion, etc.

Who Nano Is For:

- Nano is great for people who are new to the command line or for anyone who needs to make a very simple edit. If you’re a casual Linux user or hobbyist, nano might be all you ever need.

Nano vs. Emacs vs. Vim

Emacs Pros:

- Customizable and extensible.
- Powerful editing capabilities.
- Mature integration with many free software programming tools.
- You never need to leave it because you can edit files, browse the web, and so on.

Emacs Cons:

- Questionable ergonomics. (AKA: Emacs Pinky Syndrome.)
- If you want to customize Emacs you'll need to learn Emacs Lisp which introduces a whole new learning curve.
- Not available everywhere by default. If you need to edit files on a system that you don't have root access to and emacs isn't installed, then you'll end up using vim. Lack of emacs availability is common for server installations.

Who Emacs Is For:

- Emacs is for people who want more than just a text editor as Emacs can be an "environment." It's also for people who have a strong desire or need to customizations.

Nano vs. Emacs vs. Vim

Vim Pros:

- Vim serves one and only one purpose; to efficiently edit text.
- It's astoundingly powerful. Making complicated edits can be quick and easy.
- You can unlock unparalleled efficiency and speed with powerful features such as multiple file/window support, keyboard shortcuts for everything, macros, registers, quick command repetition, auto-completion, text objects, filters, and global substitutions.

Vim Cons:

- The learning curve often scares away new users.
- If you don't know what you're doing you'll look like an idiot. ("Who do I exit out of Vim?!?!?!")
- If you only ever need to perform super simple edits, Vim can be overkill.

Who Vim Is For:

- Vim is a must for Linux system administrators. It's also great for programmers because coding is mainly editing plain text files which Vim excels at. It's also ideal for anyone who works on the command line often or has to log into Linux servers. Really, Vim is for those who work with textual data of any type.