

# LECTURE 14: SCRIPTING FOR SYSTEM ADMINISTRATION

# OUTLINE

- Introduction to computer systems and operating systems
- System administration (sysadm) and sysadm background skills
- Learning the shell

# INTRODUCTION TO COMPUTER SYSTEMS AND OPERATING SYSTEMS

# COMPONENTS OF A COMPUTER SYSTEM

- Different types of computer systems
  - Tablets, laptops, smartphones
  - Desktops, workstations
  - Embedded systems: ATM, camera, copier, washing machine, car, medical devices
  - Servers
  - Mainframes, supercomputers
- A computer consists of both hardware and software

# HARDWARE COMPONENTS

- Processor
- Main memory
- Secondary memory
  - hard disk, optical disks, tapes
- Input devices
  - keyboard, mouse, microphone, touchpad
- Output devices
  - monitor, printer, speaker

# SOFTWARE COMPONENTS

- Include programs and data
- Programs: instructions for the processor
  - Application programs: the apps
  - Systems programs
- The most important systems program is the operating system (OS)
  - OS is always present when computer is running
  - Some embedded systems may not have an OS

# WHAT IS AN OPERATING SYSTEM?

- A large resource management software
- It manages hardware resources
  - processor(s)
  - main and secondary memory
  - I/O devices
- Also software resources
  - Files, processes, file descriptors, etc.
- The OS provides orderly, controlled allocation and use of the resources by the users (jobs)

# OS COMPONENTS AND SIGNIFICANCE

- An operating system generally consists of the following major components
  - Process management
  - Memory management
  - (Disk) storage management
  - File systems
  - I/O (device) management
  - Networking
  - Protection
  - User Interface
- OS hides complexity of the underlying hardware and provides user an abstraction of the computer



## SYSTEM ADMINISTRATION (SYSADM) AND SYSADM BACKGROUND SKILLS

- Essential system administration tasks
- Some UNIX/Linux experience
- Command-line interface(CLI)
- Text editing at command-line

## ESSENTIAL SYSADM TASKS

- Managing user accounts
- Adding/removing hardware
- Installing/upgrading software
- Performing backups
- Monitoring the system(performance, security)
- Troubleshooting
- Maintaining local documentation, helping users

## SOME LINUX OR UNIX EXPERIENCES

- Unix is a multi-user, multitasking operating system widely used in servers, workstations, and mobile devices
  - Example Unix: Solaris, HP-UX, AIX, Mac OS X
- Linux is a Unix-like operating system, with the defining component being the Linux kernel
  - Linux kernel is a reimplementation of UNIX kernel, originally developed in 1991 by Linus Torvalds
  - Many developed software outside the kernel
- Linux is compatible with most existing UNIX software
- Linux is free, open source, collaboratively developed
- There are other free Unix-like OS than Linux(FreeBSD, OpenBSD, OpenSolaris ...)

# WE WILL USE CLI ON UBUNTU LINUX

- As a general purpose operating system,
  - Linux is not too different from Windows or Mac OS, especially in terms of the desktop (graphical interface)
  - Logs in first, daily computer tasks: browsing the web, creating/editing files and directories, running other apps
  - For entertainment, news, book-keeping, school and work
- Hundreds of Linux distributions
  - Same Linux kernel, different ancillary materials
  - RedHat, CentOS, Fedora, Ubuntu, Debian, Mint Linux
- We use the CLI on Ubuntu Linux via ssh

# HOW TO CONNECT

- Mac/Linux:
  - No need to connect, just use your own command line(Terminal)
  - `ssh -p 130 username@server`
  - Enter password when prompted
- Windows:
  - Command Prompt
    - Not identical
  - We provide access to our UNIX terminal
    - Connect via an SSH client, such as PuTTY
    - <https://www.putty.org/>

# TEXT EDITING

- A text editor is a type of program used to edit plain text files
- Often provided with the OS
- Very different from the word processors
  - No formatting information
- A text editor can be invoked in a CLI
  - Nano
  - Vim
  - Emacs

# LEARNING THE SHELL

- What is shell and shell commands examples
- Logging in and logging out
- Commands and file name completion
- Invoking different shells

# WHAT IS SHELL

- Shell is a program
  - It takes commands issued from the keyboard as input, relays them to the OS for execution, and displays the output from the execution if there is any
- What's its first task?
  - Prints a command-line prompt e.g., #, \$, >
- There are many shell programs
- Common ones: sh, bash, csh, tcsh, ksh



# SHELL COMMANDS BY EXAMPLES

- What's my default shell?
  - `echo $SHELL`
- We start with bash, change later if you like others
- Let's issue some commands
  - `pwd`
  - `ls`
  - `less datafile.txt`
  - `mkdir CSE337`
  - `vim/vi notes1`
  - `cp notes1 tasks`
  - `mv notes1 tasks2`

## WHAT DOES THESE COMMANDS DO?

- pwd: show current directory I am in
- ls: list content of my current directory
- less: see content of file: “datafile.txt”
- mkdir: create a directory called
- cd: go into a directory
- vi create/edit a file
- cp copy a file to another file

## SOME OTHER SHELL COMMANDS

- rm: remove file(s) (DANGEROUS with -R flag)
- man rm: display manual page
- scp: securely copy files
- mutt: read emails
- cal: calendar
- wc: word, character, line counts
- gzip: gnu zip
- gunzip: gnu unzip
- yppasswd to change password

**FIN!**