

# Web Systems Notes

## Chapter 1: Operating Systems

*What is an operating system?*

- A piece of software that sits between all programs and the computer's hardware.

- Functions:

- Manages your computer
- Runs programs
- Interface between user and HW
- Provides services to programs + users
- Protects users and programs from each other

*Common OS*

Large systems:

- Mainframes: 1960

- IBM runs mainframes
- handles 1000s of users simultaneously, runs Linux
- Bulk data processing

- Minicomputers

- OpenVMS, Unix (Solaris, Mac OSX)

- Personal computers

- Linux
- Microsoft Windows

- Embedded systems

- Military, telecommunications

*OS Architecture: The Unix Onion Model*

[INSERT MODEL]

Top Layers: userland, application layer

- Hardware:

- Central Processing Unit (CPU): e.g intel core 2 duo
- Memory
- I/O
- Storage

- Kernel

- Controls the HW directly (drivers, firmware)
- Provision of resources and services to applications e.g CPU, memory
- Manages access to privileged resources

- User Interface

- GUI: user friendly interface on top of the operating system
- Often runs the "shell" commands transparently

- Shell

- Command line interface (CLI) or interpreter, command prompt, terminal
- Makes a set of commands available to the user

- Services

- programs that run "behind the scenes": provide system support e.g security/networking

*User Interfaces*

- How to design a user interface

- Pick intended audience
- good workflow

- polish
- consistency
- psychology
- Computer interfaces
  - CLI: interact through the keyboard and monitor
  - GUI: Interacts via windows, icons, mouse, menus
- CLI
  - sh 1969 (BASH), CPM 1973 (DOS), cmd.exe windows shell
- Which is better?
  - Neither, each has an important role and has strengths and weaknesses
- Why have multiple interfaces?
  - Customization
  - Automation
  - Understanding

#### *Graphical User Interface: Strengths and weaknesses*

- Strengths
  - Little/no experience required
  - Good for graphics e.g. artwork, desktop publishing
  - User friendly, intuitive
  - Hides complexity from users
- Weaknesses
  - Can't do everything
  - Can crash the system
  - User is unsure of what the OS is doing
  - Slows the computer down
  - Needs better HW
  - Hides complexity from users

#### *CLI: Strengths and weaknesses*

- Strengths
  - Greater flexibility
  - Fine tuning -> parameters
  - essential for system administration
  - faster, less overhead
  - runs on simple hw
  - can run remotely
  - robust
- Weaknesses
  - Hard to learn: Cryptic commands
  - multiple options
  - output cryptic
  - Inconsistent commands
  - No graphics
  - No safety net -> "expert mode"

#### *Batch files and scripting languages*

- Automate CLIs
- Put a sequence of commands into an executable file : CLI treats the file as a command
- Programming features included: Scripting language

#### *Example scripting language*

- Bash, Korn Shell, C Shell, Z shell
- DOS batch language

#### *Characteristics*

- Variables are usually untyped (loosely bound)

- Language syntax is often inconsistent
- Often designed and created by one person to get a particular job done
- Usually run through an interpreter not a compiler

#### *Evolution of scripting languages*

- Gain extra features through evolving
- Perl: Started as a scripting language, now a generic prog language
- Windows shell: replaced by powershell
- Bash: Linux default CLI, now includes arrays, data types

## **Chapter 2: UNIX Operating Systems**

- Many versions of Unix. Most are based from 2 original versions:
  - o System V - the original version from AT&T
  - o BSD - from the University of California at Berkeley
- Adhoc development: grew in an unregulated fashion
  - o Thus more powerful and versatile, but confusing
- IEEE: Standardizing UNIX attempts
  - o POSIX
  - o Defined commands, utilities, system interfaces, scripting language
  - o High cost, too complex
  - o Inconsistency and difficulty in transferring code between systems
  - o Single Unix Specification (SUS) agreed
- IBM VM/CMS:
  - o Virtual machine/conversational monitor system
  - o Used on mainframes
- What is UNIX?
  - o Set of ideas
  - o Any person or group is free to implement these ideas
  - o Based on simple concepts:
    - Files, processes, permissions, users
    - HW devices are represented as files
    - Simplified picture of Unix
    - Unix can implement new ideas and tech easily
  - o Portable
    - Prog language C
    - Not tied to any CPU
    - Free
    - Efficient, stable, secure
    - Designed for security for multi-user systems
    - Set of tools approach
    - Simple commands, pipes, i/o redirection
- File system
  - o Part of the operating system that manages data storage and access
  - o Disk physical structure:
    - Organized into:
      - Tracks: concentric rings on the platter
      - Heads: reads data from a platter
      - Cylinders: collection of all tracks on platters which are horizontally in the same position
      - Sectors: part of a track for data
-