CSCI 2132: Software Development

Introduction to Unix

Norbert Zeh

Faculty of Computer Science
Dalhousie University

Winter 2019

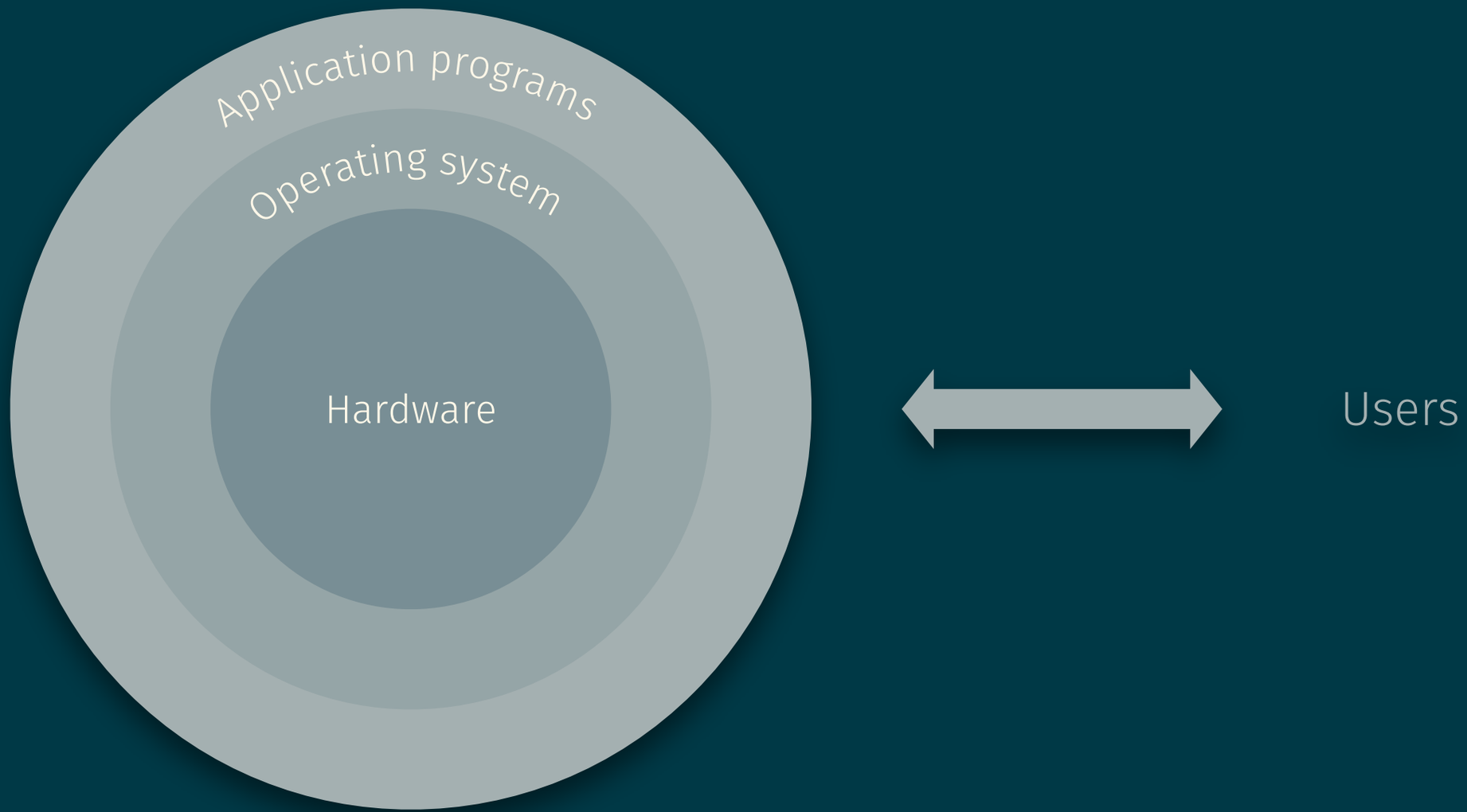
Overview

- Refresh our knowledge of operating systems
- Learn about Unix-style operating systems
- Learn about the Unix shell as a command line interface (CLI)
- Learn about the file system
- Learn about command line tools and
  some software development tools

Some Functions of an Operating System

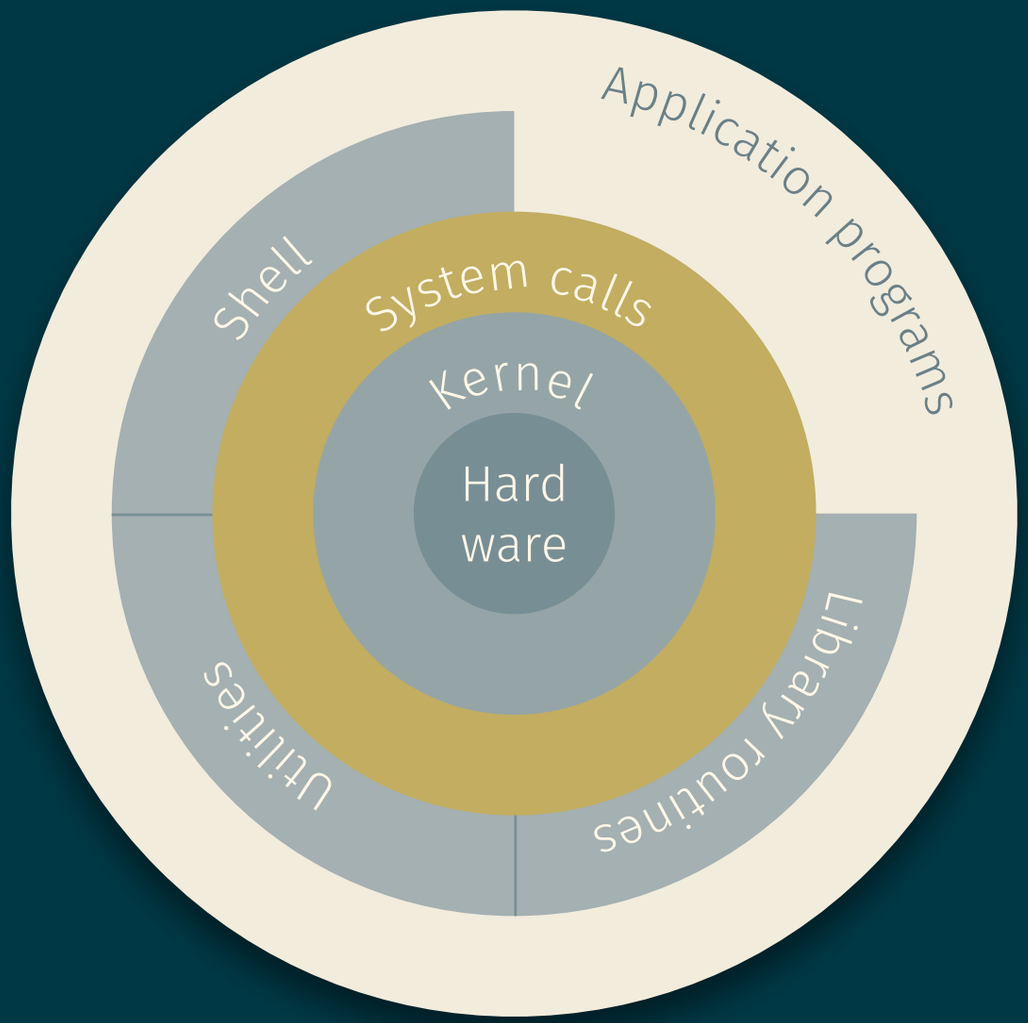
- Provide an **interface between applications and hardware**
  - Read keyboard, write to screen, read/write disk, talk to printer, communicate with network card, ...
  - **Hide complexity of controlling hardware** from applications
    - Protect hardware from user and programming errors
- Manage hardware resources
  - CPU time, memory access, disk space, ...
- **Protect user programs and data from each other** (security)
- Support inter-process communication, sharing
- Provide resource sharing among users, processes

Overview of Unix-Style Architecture



Onion skin model

A More Detailed Onion Skin Model



Adapted from W.R. Stevens and S.A. Rago. *Advanced Programming in the UNIX Environment*.

UNIX's predecessor: Multics

- Multics OS started in 1964
- Developed by Ken Thompson, Dennis Ritchie, and others
- Collaboration between MIT, AT&T (Bell Labs), and GE for GE-645 computer
- Advanced system with many features and idea of "computing utility"
- Hardware did not keep up with software, so it was slow and expensive to run
- AT&T withdrew from the project
- Ken Thompson started to work on a new system

UNIX: 1969–

1969	Implemented for a PDP-9 on a GE system
1970–1973	PDP-11, C language, implemented in C, called it UNIX (Brian Kernighan)
1973–1979	Source code available to universities, PDP-11, very popular very quickly
1980s	Commercialization: System V, BSD, GNU (1985)
1991	Linux (Linus Torvalds), GNU/Linux, new code, distributions

Other UNIX/Linux-based OSs: Chrome, Android, macOS, ...

More About UNIX History

- More in Nemeth et al., *UNIX and Linux System Administration Handbook*, section “A Brief History of System Administration”

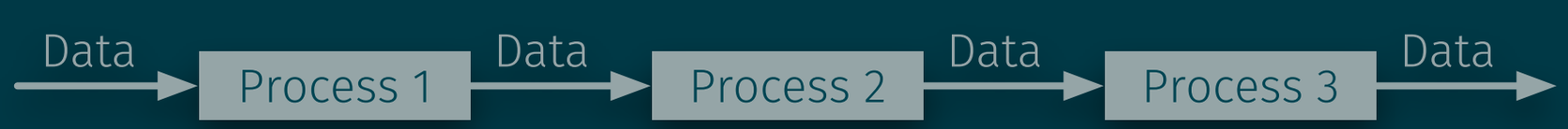
- UNIX had many advanced features from the early days, such as concurrent execution

Unix Philosophy

- Write programs that
  - Process text streams (universal interface)
  - Work together (so they can be easily combined)
  - Do one thing and do it well
- Allows for simple, elegant, and robust solutions
- Programs (utilities) can be sequenced using pipes
- Typical user is a programmer
  - Can decompose problems into subproblems
  - Used to concise syntax
  - Understands data flow

Pipelines

- Pipes specify that the output of one process is to be used as the input to another process:



Example: `who | sort`

- The symbol `|` is called “pipe”
- Pipes can combine any sequence of processes that take text input and produce text output

Another Unix Characteristic

- (Almost) any data that can be manipulated is a **file**:
  - Actual files
  - Terminal I/O (keyboard, screen, …)
  - Hardware control (network cards, peripherals, …)
  - Kernel settings

- That data is manipulated by **processes**:
  - User programs
  - Kernel processes

Some Resources the OS Manages

- Central processing unit (CPU)
- Random access memory (RAM)
- Read-only memory (ROM)
- Disk memory (hard disk, CD drives, ...)
- Graphics card
- Network card
- Peripherals (keyboard, monitor, mouse, ...)

Some Important Unix/OS Concepts

File	Collection of data (Sequence of bytes) • Stored on disk, CD, Amazon S3, … • On Unix, also keyboard and screen
Program	File that stores machine code that can be loaded into memory and run
Process	A running program
Owner of a file or process and **file permissions**	Determine who is allowed to interact with a file/ process and in which way
Hierarchical directory structure	
Location of a file	Place in the directory hierarchy where the file is found
Location of a process (Working directory)	Reference point for file accesses made by the process
System calls	Unix's interface for the creation, modification, and destruction of files and processes

UNIX Directory Hierarchy