

CSCI 2132
Software Development

Lecture 2:
Introduction to UNIX and Unix-like Operating Systems

Instructor: Vlado Keselj

Faculty of Computer Science

Dalhousie University

Previous Lecture

- Course Introduction
 - logistics and administrivia
 - important dates, course description
 - evaluation scheme and criteria
 - textbooks
 - lectures, exams, assignments, lab work
- Academic integrity policy
- Culture of respect
- Main learning objectives
- Motivation: Why UNIX, Why C
- Tentative list of course topics

Part 1: Unix Operating System

- Reading: Unix book, Chapter 1
- In the first part we will refresh our general knowledge about operating system,
- learn more details about the Unix-style operating systems
- learn about shell as a command-line interface
- learn about the file system
- learn about utilities and some tools used in software development

What is an Operating System?

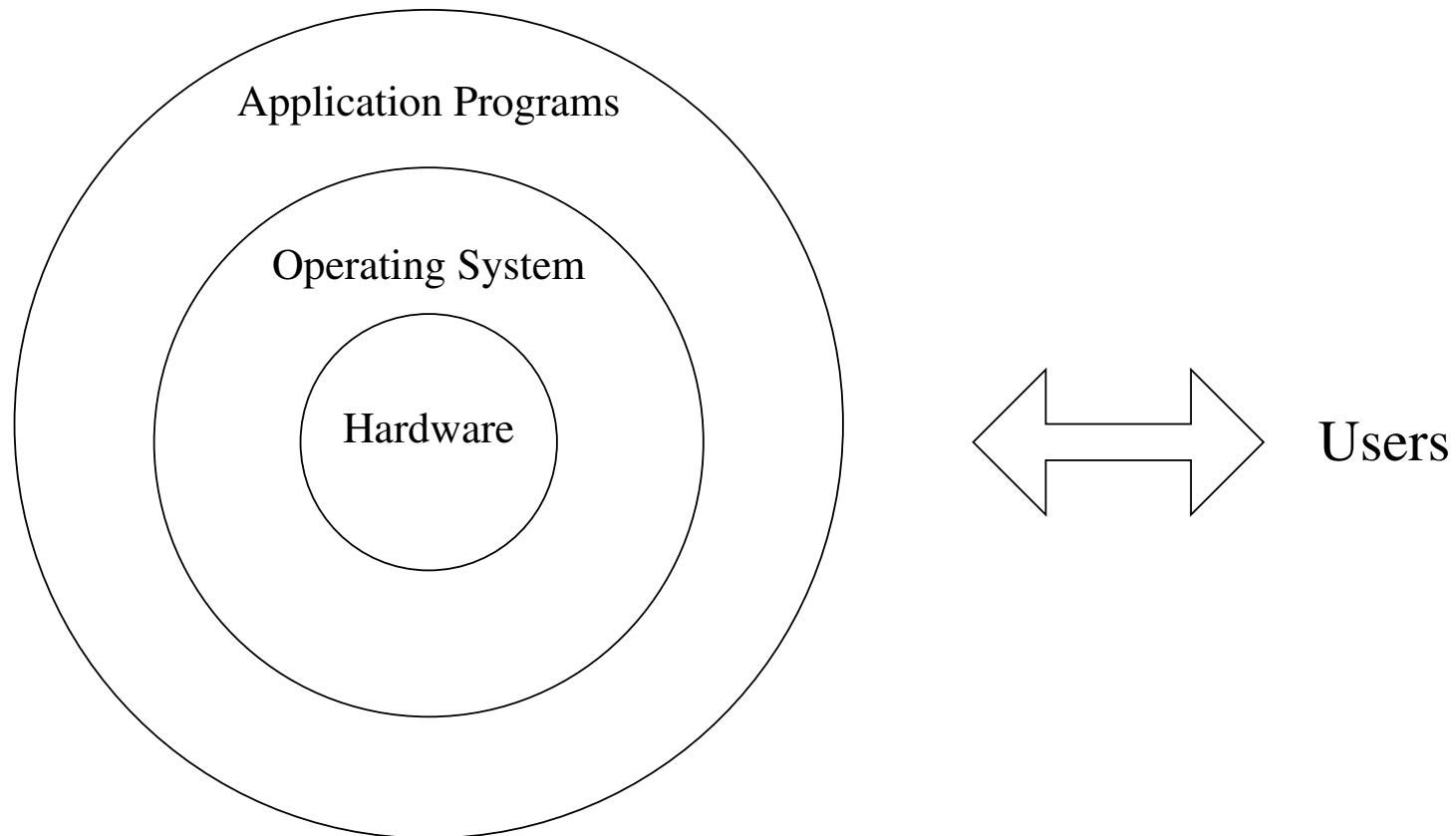
- What are some main functions of an operating system?

Some Functions of an Operating System

- Provides an interface between application programs and the hardware
 - E.g., reads keyboard, writes to screen, writes and reads from disks, sends data to printer, communicates with network card, . . .
 - Hides the complexity of hardware interfaces from application programs
 - Protects the hardware from user mistakes and programming errors (to prevent crashes)
- Manages the hardware resources of the computer systems
 - CPU time, disk space, memory access, . . .
- Protects user's programs and data from each other (security issues)
- Supports inter-process communications and sharing
- Provides resource sharing among users and processes

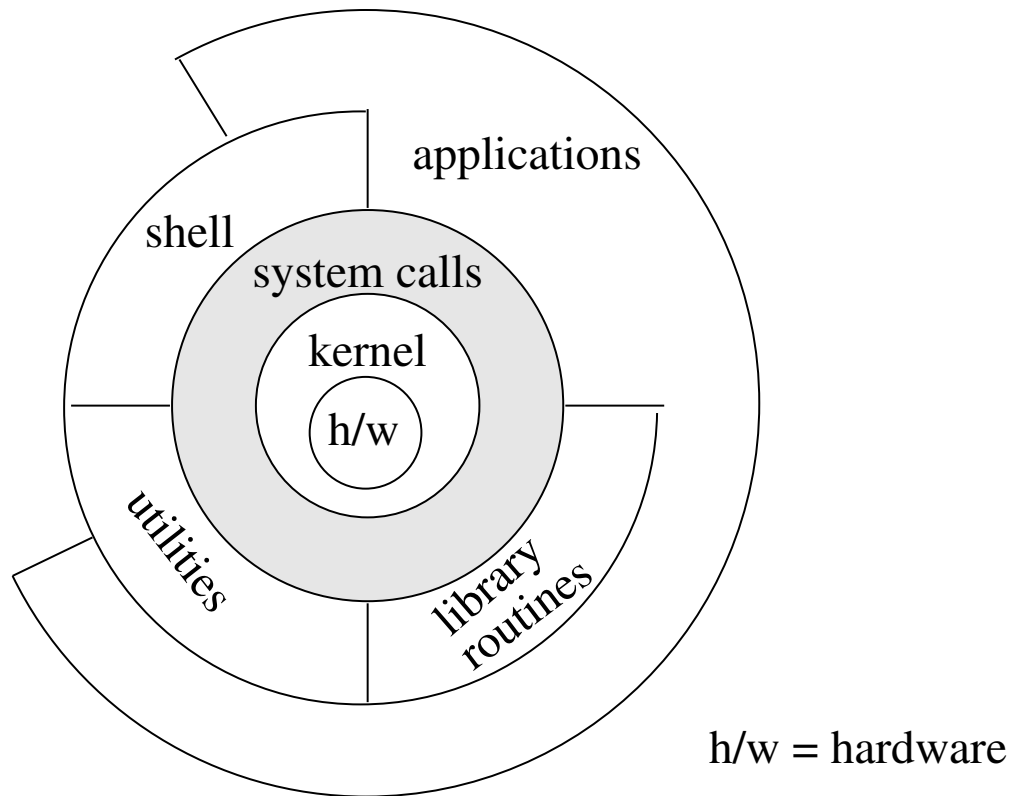
Overview of Unix-style Architecture

- The “Onion Skin Model” of Operating System



A more-detailed Onion-skin Model

Adapted from “Advanced Programming in the UNIX Environment” by W. Richard Stevens and Stephen A. Rago:



A Brief History of Unix: Multics

- Multics OS started in 1964, 5 years before Unix
- Developed by Ken Thompson, Dennis Ritchie, and others
- Collaboration of MIT, AT&T (Bell Labs), and GE for GE-645 computer
- Advances system with many features and an 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 and Ken Thompson started working on a new system

UNIX: Created in 1969 by Ken Thompson



Ken Thompson; Dennis Ritchie and Ken Thompson (sitting) at PDP-11

UNIX: 1969– Development

- 1969: Implemented for an old PDP-7 in assembly language on a GE system
- 1970-3: PDP-11, C language, reimplemented in C, pipes, called it UNIX (Brian Kernighan)
- 1973-9: Source code available to universities, PDP-11 machines, very popular, very quickly
- 1980s: Commercialization, System V, BSD, GNU (1985)
- 1991: Linux (by, Linus Torvalds), or GNU/Linux, new code, distributions
- Other Unix/Linux-based OS's: Chrome, Android, MacOS, etc.

More Reading about UNIX History

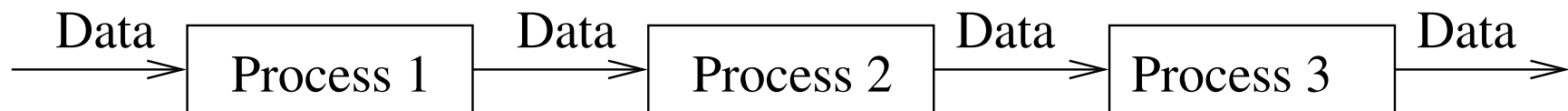
- You can read a bit more about UNIX history in the book by Nemeth *et al.*, *Unix and Linux System Administration Handbook* in the section 'A Brief History of System Administration'
- UNIX had many advanced features from early days, such as concurrent execution

Unix Philosophy

- Write programs that. . .
 - handle text streams; because that is a universal interface
 - work together; because then they can be easily combined
 - do one thing and do it well
- This allows for simple, elegant, and robust solutions
- Programs (utilities) can be combined into pipes
- A typical user is a programmer
 - can decompose problems into subproblems, used to concise syntax, understands data flow

The Concept of Pipeline

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



- Example:
`who | sort`
- Symbol ‘|’ is called ‘pipe’
- Related to every process having three default I/O channels: stdin, stdout, and stderr (standard input, standard output, standard error output)