CS 854: Advanced Topics in Operating Systems

Instructor: Ali Mashtizadeh

Overview

Website: https://cs.uwaterloo.ca/~mashti/cs854-f17/

Piazza: https://piazza.com/class/j7aoc9b46mc6q6

Email: mashti@uwaterloo.ca

All handouts and lecture notes will be online

Please print them yourselves

Prerequirements

Assume familiarity with C, UNIX, etc.

Have taken an undergraduate course in operating systems

Grading (tentative)

25% Class Participation

25% Paper Summaries/Presentation

50% Project/Peer Reviews

Expectations:

- Everyone must present papers in this class
- Everyone must read and be ready to discuss every paper

Paper Reading

Everyone must read every paper before class

Write ~1 page response due by email or in class

- Any key ideas and tools you could use in other work?
- Anything you agree/disagree with? Or would do differently?
- Any assumptions you think do not apply today?

Group Project

Choose a small sized research project that you complete in the term with a group of 2-3 people

Write a one page proposal with your teammates

You could build a prototype system, expand an open source project, or conduct an evaluation of an existing system

Course Topics

Synchronization & Multi-core

Language-driven Design

Security

Storage and File Systems

Virtual Machines

What is an operating system?

Provides abstractions for the programmer

Hides the details of hardware

- Manages hardware resources
- Providing higher level interfaces

Provides protection

- Prevent one process/user damaging another process/user’s stuff

Why study operating systems?

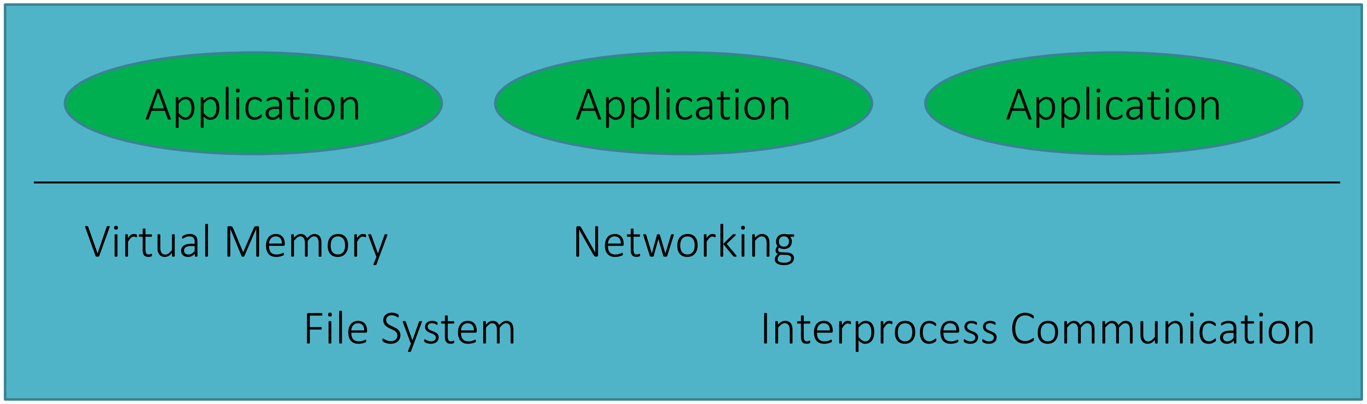
Operating systems are a maturing field

Most people use a handful of mature OSes: Windows, Mac OS, Linux, FreeBSD, etc.

Improving performance, power utilization, and security are all OS issues

Virtual machines, embedded devices, unikernels provide opportunities for rethinking operating systems again

Typical operating system



Applications run as user-level processes

Kernel manages hardware resources and security

Applications typically call the kernel through “system calls”

Application/Kernel transition is called a “context switch”