CST 334: Operating Systems

Glenn Bruns

CSUMB

What is a "systems" course?

In a CS systems course, we think about:

physical resources shared by multiple people or applications

What are the resources of a computer system?

CPU

Disk

Memory (RAM)

I/O

How is Burger King like an OS?

Burger King

What are the resources?

Who are the users?

Burger King

How to make the resources easy to use?

How to make sharing of the resources fair?

How to protect the system from the users?

How to measure how well Burger King is working?

CSUMB

Resources?

Users?

CSUMB

How to make the resources easy to use?

How to make sharing of the resources fair?

How to protect the system from the users?

How to measure how well CSUMB is working?

Are resources used efficiently?

Other examples

Freeway:

- users = drivers
- resources?
  - paved road
  - toll collectors and machinery

Gym:

- users = customers
- resources?
  - workout equipment
  - employees
  - physical space

Guiding questions

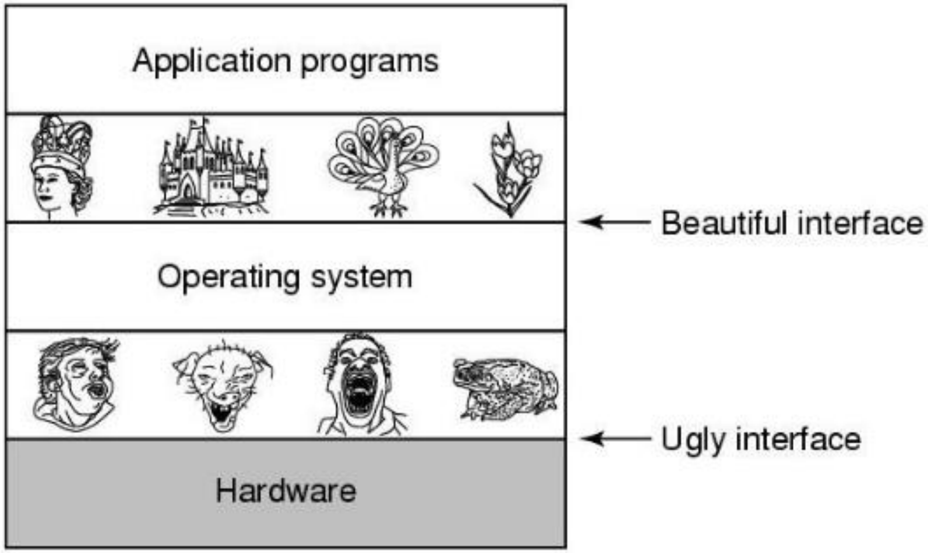
- □ What are the virtual resources or services offered to users? Are they easy to use?
- □ How to ensure fair sharing of resources between users?
- □ How to protect users from each other, and protect the system from users?
- □ What workloads do we use to measure performance?
- □ What metrics do we use to measure performance?
- □ How efficiently are resources managed?

What is an Operating System?

Code that sits above bare hardware.

Main roles of the OS:

- Virtualizer: takes complicated resources, like CPU, memory, disk, and turns them into easy-to-use virtual resources
- API: provides a system call library for programs
- Resource manager: ensures fair and safe sharing of resources by programs/users



Problem Statement

The problem for an operating system: to provide resources to users in an easy-to-use, reliable, fair, efficient, and secure way.

What does it mean for an OS to be:

- easy to use?
- reliable?
- fair?
- efficient?
- secure?

An example of virtualization

- bare hard drive is complicated to use
  - low-level interface
  - data organized into tracks, sectors, …
- OS file system is easy to use
  - files and directories, simple naming scheme
  - data is a sequence of bytes

OS Problem 1:

How to run multiple applications at once?

- Early operating systems didn’t support this
- Basic idea: give each program a split-second of system resources (memory, CPU, disk, network)
- Problem: if OS gives a program control of the CPU, how does it get control back?
- Problem: how to manage programs so time is not wasted while programs are waiting for resources?

OS Problem 2

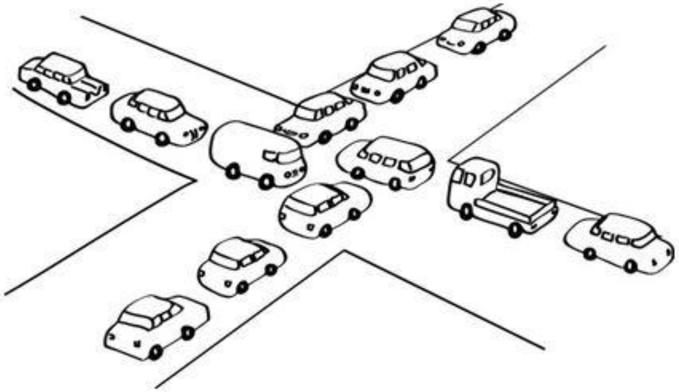
How to provide a simple memory space for programs?

- do you need to think about where your program will sit in memory?
- do you think about how much memory other programs need when you write a program?
- do you think about how much memory a computer has when you write a program?

OS Problem 3

How can concurrent programs coordinate?

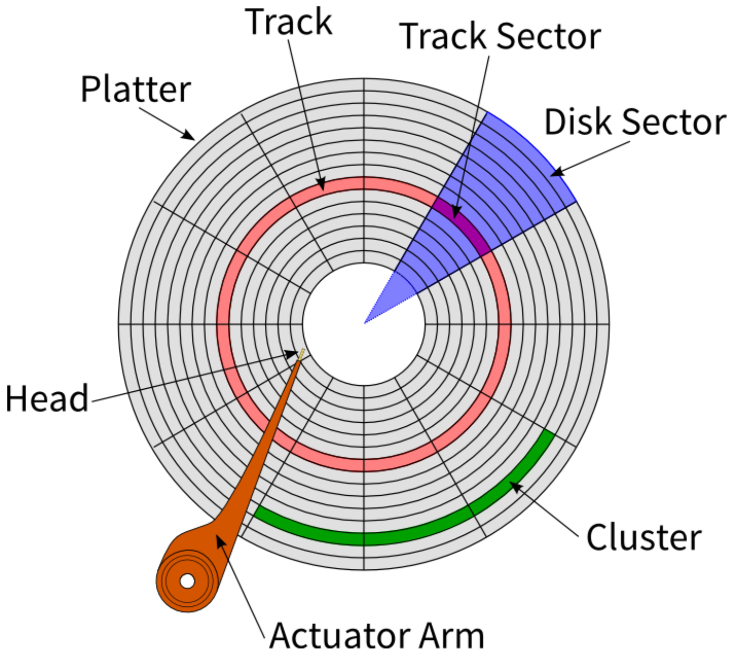
- example problem:
  - □ a computer has two optical drives
  - □ program 1 needs two drives, so requests and gets access to one
  - □ program 2 needs two drives, so requests and gets access to one
  - □ both programs are stuck waiting for the other program to release a drive
- how to avoid it? how to detect it? how to fix it?



OS Problem 4

How to provide a simple file system?

- which disk blocks go with which file?
- where to store a file's attributes?
- how to allow for file sharing?
- how to achieve good performance with large drives?



Linux

Unix -> Minix -> Linux

Unix-derived operating systems are everywhere:

- Linux, Mac OS, Android

We’ll look at Linux because:

- it’s important and widely-used
- Unix/Linux design ideas are good
- more support for power users