CIS 310 Operating Systems

Week 4: Devices and File Systems

Dr. Brian C. Ladd

Wednesday 4th October, 2023

Outline

Computer Architecture

- CPU
  - Registers
  - Program Counter
  - `fetch-decode-execute`
- RAM – An array of bytes
- I/O devices

Computing Resources

- CPU *cycles*

Computing Resources

- CPU *cycles*
- RAM *address spaces*

Computing Resources

- CPU *cycles*
- RAM *address spaces*
- RAM *safe concurrent access*

Computing Resources

- CPU *cycles*
- RAM *address spaces*
- RAM *safe concurrent access*
- Device *interaction*

Computing Resources

- CPU *cycles*
- RAM *address spaces*
- RAM *safe concurrent access*
- Device *interaction*
- Persistent storage *organization*

Side Trip

Definition (Mechanism)

A *low-level* capability. **How** the system works.
For example: *How* does the operating system switch the contexts of two processes?

Definition (Policy)

A *high-level* strategy. **Which** resources go to which process.
For example: *Which* process is next scheduled to have the CPU after we interrupt the current one?

Policies are implemented by making use of **mechanisms**. Most mechanisms could be used by *different* policies.

Operating System

A resource manager for one or more *processes* using the computer hardware concurrently.
Concurrent – Happening at the same time; overlapping *duration*.

Operating System Services

- Multiprocessing

Operating System Services

- Multiprocessing
- Safe Sharing

Operating System Services

- Multiprocessing
- Safe Sharing
- Device Interface

Operating System Services

- Multiprocessing
- Safe Sharing
- Device Interface
- Persistence (files)

Operating System Services

- Multiprocessing
- Safe Sharing
- Device Interface
- Persistence (files)
- Error detection/correction/recovery.

Multiprocessing

- Sharing of physical CPU/physical RAM

Multiprocessing

- Sharing of physical CPU/physical RAM
- Process management – loading/unloading, starting/pausing/stopping

Multiprocessing

- Sharing of physical CPU/physical RAM
- Process management – loading/unloading, starting/pausing/stopping
- Scheduling based on some criteria

Safe Sharing

Uncontrolled access of a single physical resource is inherently unsafe.
Think about a dorm hall with a single *shower*.

Device Interface

Multiple different devices attach to the computer. Many different user programs want to use whatever devices are available.

- A *facade* placed in front of all devices raises the level of abstraction at which they are used.

Think about all the music file formats and then using VLC.

Device Interface

Multiple different devices attach to the computer. Many different user programs want to use whatever devices are available.

- A *facade* placed in front of all devices raises the level of abstraction at which they are used.
- *Driver* code exposes the required interface while hiding the underlying details.

Think about all the music file formats and then using VLC.

A File

An example of a common interface:

- A **sequence of bytes**

Some features can be relaxed. `/dev/null`, `/dev/random`

A File

An example of a common interface:

- A **sequence of bytes**
- Must be **opened** before/**closed** after interaction.

Some features can be relaxed. `/dev/null`, `/dev/random`

A File

An example of a common interface:

- A **sequence of bytes**
- Must be **opened** before/**closed** after interaction.
- A **naming scheme** is provided to find resources.

Some features can be relaxed. `/dev/null`, `/dev/random`

A File

An example of a common interface:

- A **sequence of bytes**
- Must be **opened** before/**closed** after interaction.
- A **naming scheme** is provided to find resources.
- Persists longer than one process execution.

Some features can be relaxed. `/dev/null`, `/dev/random`

What Only Hardware Can Do

- Interrupts (including Timers)
- Privilege Bit
- System Calls
- Address Translation
- Atomic Instructions

System Calls

- `strace`

Virtualization

- `ps aux`
- `htop`

Looking at Memory

- gdb