

csci 3411: Operating Systems

Reflections

Gabriel Parmer

...Back Where We Started

- **What is an Operating System?**
- What you thought on the first day:
 - Intermediary between human and hardware/assembly
 - Man/machine interface
 - Way user accesses capabilities of the computer
 - Layer of interaction between peripherals and programs
 - Multiple layers
 - Provides security
 - Graphical User Interface
 - UI to hardware

What Have We Learned?

- **What is an Operating System?**
- What are the general *themes* of the class
 - Not the specific policies or mechanisms
- What is an OS, and what *isn't an OS*?

Persistent Topics

- Concurrency
 - Synchronization, processes, threads, deadlocks,
- Scheduling
 - CPU, disk requests, critical sections
- Naming
 - VASes, directory structures, capabilities,

Persistent Topics II

- Protection/Fault Isolation/Security
 - Caps/ACLs, processes, VASes
- Caching
 - Virtual memory/swapping, buffer cache
- Trade-offs – ...which is the best...
 - Space/time – memory allocation
 - fault tolerance/performance – thread models
 - complexity/performance – FIFO vs. *
 - fairness/throughput – FIFO vs *

Persistent Topics III

- **Hardware/Resource Management**
 - CPU – Fairly/efficiently multiplex processor(s) amongst multiple competing interests
 - Memory – Efficient allocation w/ little wasted memory, VAS
 - Disk – Maximize throughput of disk via concurrency, data layout, and request scheduling

Persistent Topics IV

- **Abstraction – Kernel/OS interface**
 - interrupts → threads → processes → semaphores
→ nodes
 - physical mem → virtual mem → malloc/free
 - disk → files → directories → open → read/write
 - single namespace → VAS → caps/ACLs → RBAC

Deeper Down the Rabbit Hole...

- Special topics classes
 - Data centers and cloud management
 - If full, talk to professor Wood
 - Foundations of Security
 - Theory of security