Course Review

ECE595
Apr 21

Y. Charlie Hu

Plan

- Final exam warm up!

- Reminder: course evaluation due Apr 30
- Final exam: ME 3006, May 3, 8:00-10:00am

- 2nd half semester review
- Quiz3

- Movie quiz
- Course summary (annotated slides from week1)

Course review – 2nd half semester

Movie “Paycheck”

- Which ECE595 concept does the movie storyline remind you of?

A: Priority CPU scheduling

B. Virtual memory

C. Journaling file system

Course Summary
(annotated slides from Lecture 1)

[lec1] What is an OS?

“Code” that *sits between*:

- programs & hardware
- different programs
- different users

But what does it do/achieve?

[lec1] What is an OS?

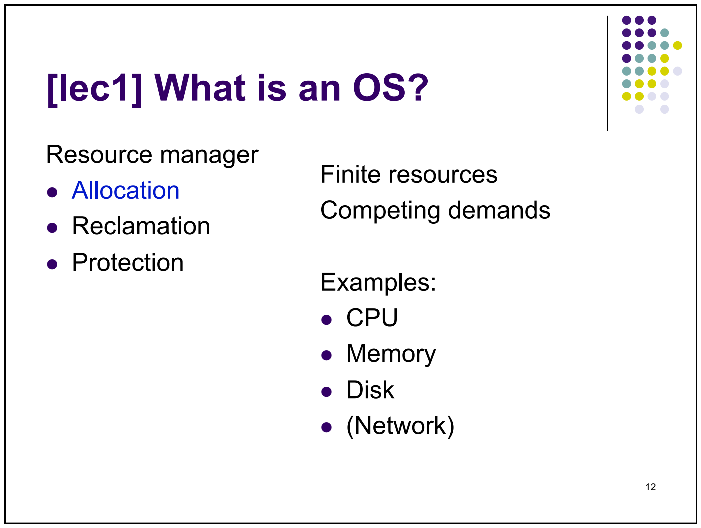
- Resource manager
- Extended (abstract) machine
- A giant interrupt handler!

Makes computers efficient and simple to use

[lec1] What is an OS?

Resource manager (answer1)

- Allocation
- Reclamation
- Protection

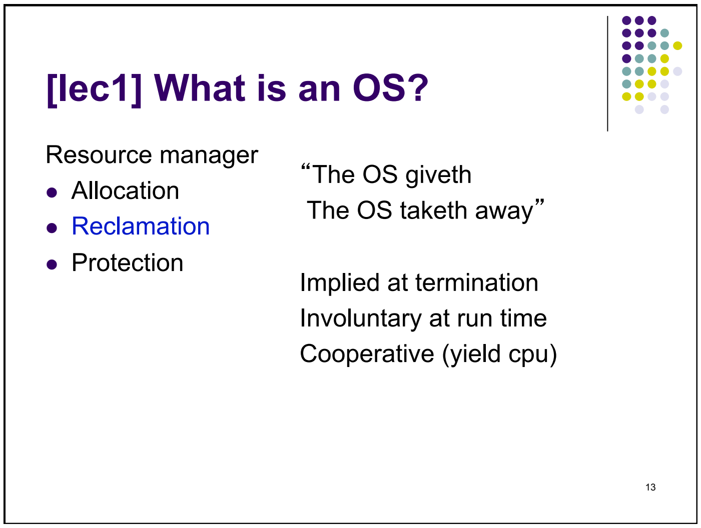


[lec1] What is an OS?

Resource manager
- Allocation
- Reclamation
- Protection

Finite resources
Competing demands

Examples:
- CPU
- Memory
- Disk
- (Network)

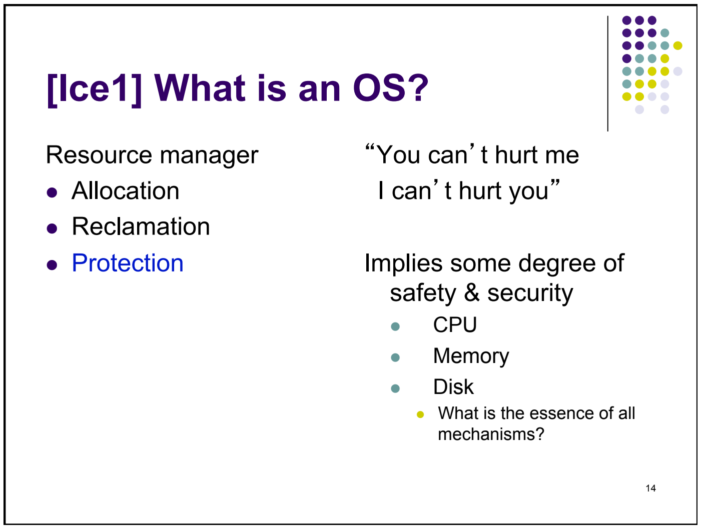


[lec1] What is an OS?

Resource manager
- Allocation
- Reclamation
- Protection

"The OS giveth
The OS taketh away"

Implied at termination
Involuntary at run time
Cooperative (yield cpu)

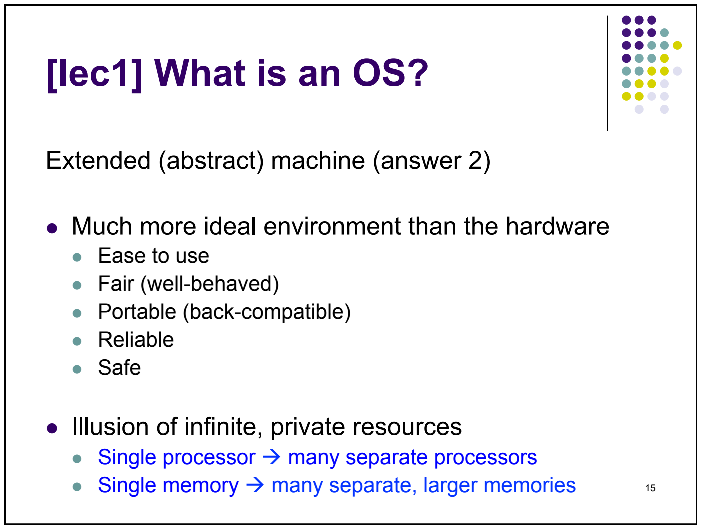


[lce1] What is an OS?

Resource manager
- Allocation
- Reclamation
- Protection

"You can't hurt me
I can't hurt you"

Implies some degree of safety & security
- CPU
- Memory
- Disk
  - What is the essence of all mechanisms?



[lec1] What is an OS?

Extended (abstract) machine (answer 2)

- Much more ideal environment than the hardware
  - Ease to use
  - Fair (well-behaved)
  - Portable (back-compatible)
  - Reliable
  - Safe
- Illusion of infinite, private resources
  - Single processor → many separate processors
  - Single memory → many separate, larger memories

[lec1] Separating Policy from Mechanism

Policy – decisions on how to use tool

Examples:

- CPU scheduling policies
- Page replacement policies
- Buffer cache replacement policies
- Disk allocation policies

Mechanism – tool to achieve some effect

Examples:

- Priority scheduling vs. lottery scheduling
- FIFO w/ 2nd chance vs. Clock: a simple FIFO w/ 2nd chance

Separation leads to flexibility

[Ice1] Is there a perfect OS?

Portability
Security
Fairness
Robustness
Efficiency
Interfaces

- Conflicting goals
  - Fairness vs efficiency
    - SJF vs. RR
    - FIFO vs. SCAN
  - Efficiency vs robustness
    - Buffer caching
- Don't know future
  - CPU scheduling
  - Page replacement
  - Disk scheduling

[lec1] There is no magic in OS design

This is Engineering

- Imperfection
  - Don't know future
- Tradeoffs
  - Segmentation vs. paging
  - Read/write API vs. mmap
- Constraints
  - hardware, cost, time
    - FIFO w/ 2nd chance
    - Enhanced version
    - Approx. LRU
- Optimizations
  - After functionality
  - 1-level paging -> 2-level
  - Basic FS → Buffer caching

Nothing's Permanent

- High rate of change
  - Killer-app: Databases/web servers
  - Arch: uniprocessor -> Multi-core
- Cost / benefit analyses
  - motivation for mmap
  - Semaphore impl on multiprocessor
- One good news:
  - Lots of inertia
    - Principle of locality
      - TLB
      - Demand paging
      - Buffer caching
    - Extra level of indirection
      - Dynamic memory relocation
      - 1-level paging -> 2-level paging
      - UNIX multi-level indexed files

[lec1] About this course…

Principles of OS design

- Some theory
  - SJF optimal
  - Working set modeling
- Some rational
  - Optimize the common case
    - Sequential file access - prefetching
  - Locality -> caching
  - Why mmap()?
- Lots of practice
  - Locality exists/how much?
  - Dist. of file size (UFS inode)
  - Buffer cache size vs. VM size?

Goals

- Understand OS design decisions
- Basis for future learning

To achieve the goals:

- Learn concepts in class
- Get hands dirty in labs

Great ideas in Computer System Design (1)

- *"All computer science problems can be solved with an extra level of indirection"*

  -- David Wheeler

1. Dynamic memory relocation
   - Base&bound, segmentation, paging
2. One-level paging → Two-level paging
3. UFS multi-level indexed files
4. Boot block → stores bulk of the bootstrap program
5. NFS: transparency via VFS

Great ideas in Computer System Design (2)

- Principle of locality → Caching

1. TLB
2. Demand paging (VM)
3. Buffer cache in FS
4. On-disk cache
5. Client caching in NFS

(6. Hardware cache, L1, L2, etc.)

My best wishes

- Be well
- Do good work
- Keep in touch

from Garrison Keillor, *A prairie home companion radio show*