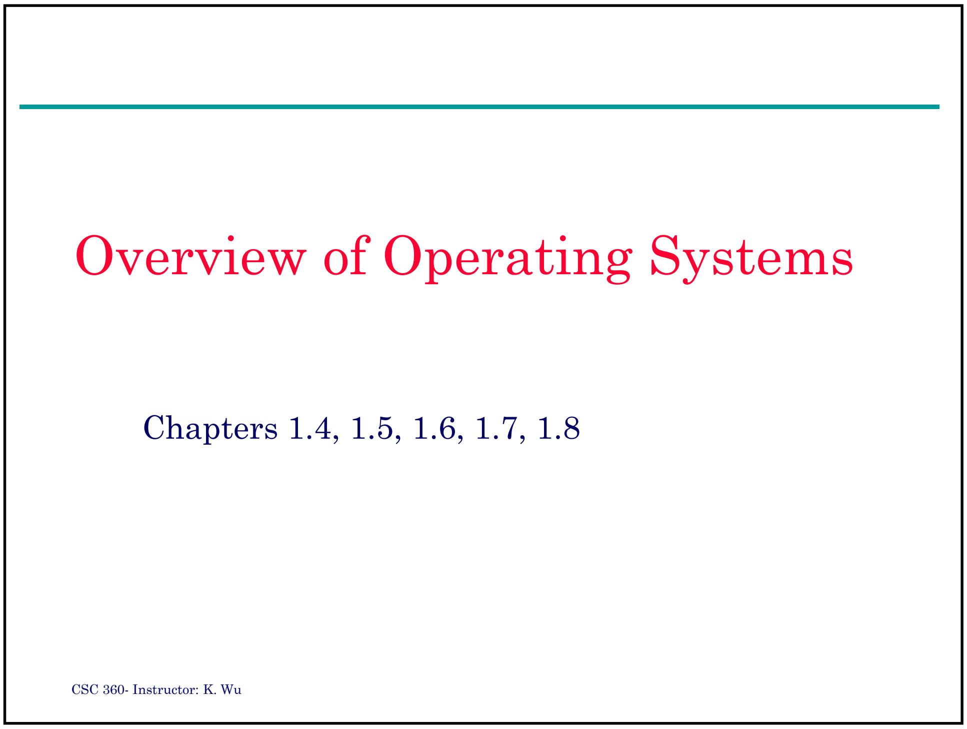
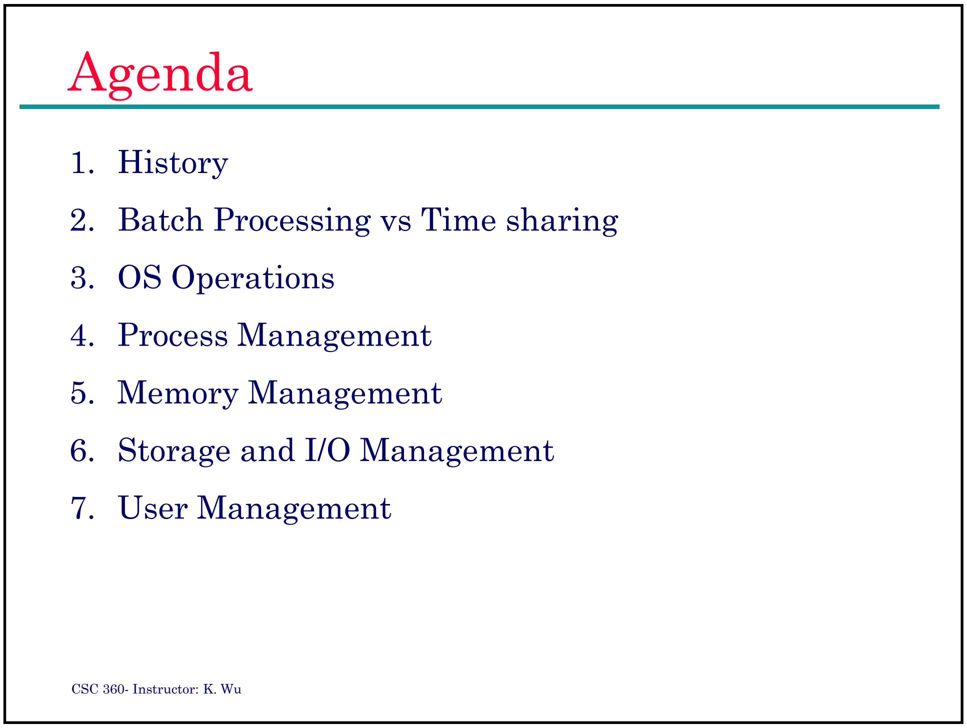


Overview of Operating Systems

Chapters 1.4, 1.5, 1.6, 1.7, 1.8



Agenda

1. History
2. Batch Processing vs Time sharing
3. OS Operations
4. Process Management
5. Memory Management
6. Storage and I/O Management
7. User Management

1. History: generations

Uniprogramming

- “One program at a time”
  - – start, execute, {wait, execute}*, finish
  - – wait for: input/output, other programs, etc
  - – CPU may be idle most of the time

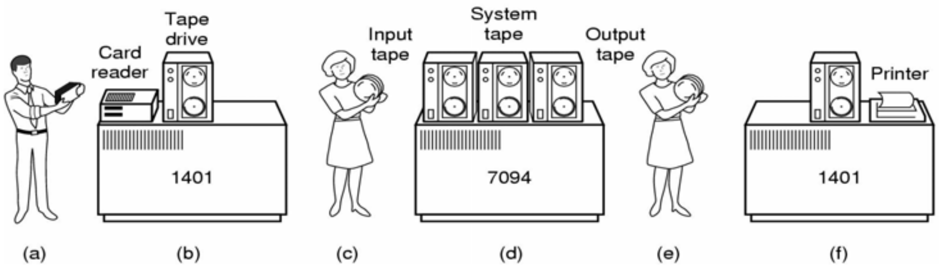
Multiprogramming

- “Many programs at a time”
  - – try to keep CPU always busy
  - – handle multiple programs at the same time
  - – “share” (a) CPU

2. Batch Processing vs. Time Sharing (1)

Batch Processing:

- Load a pool of jobs
- Execute one job *until* it is blocked
- Pick another one to execute



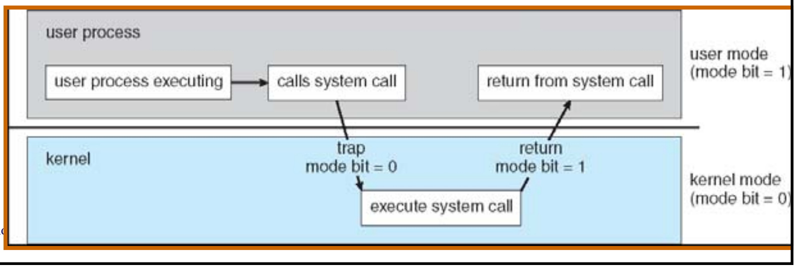
2. Batch Processing vs. Time Sharing (2)

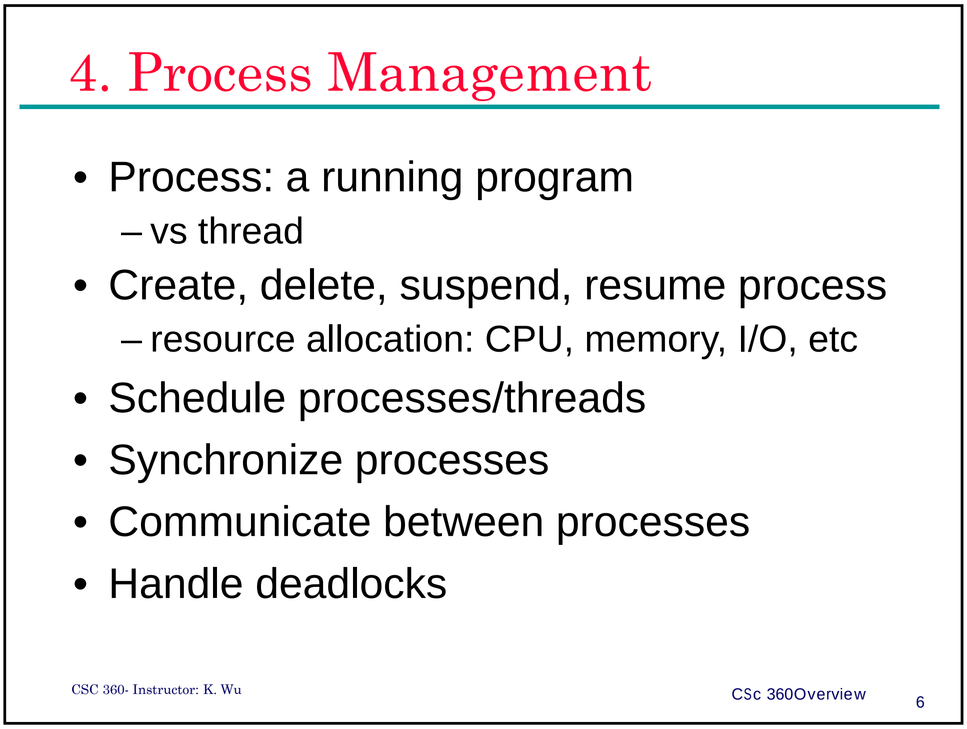
Time Sharing:

- Execute one job up to a certain time
  - e.g., hardware timer with counter
- Switch to another one to execute
  - job scheduling, memory swapping
- Seem to execute *many* jobs at the same time
- Batch processing vs time sharing
  - job responsiveness
  - switching overhead

3. OS Operations

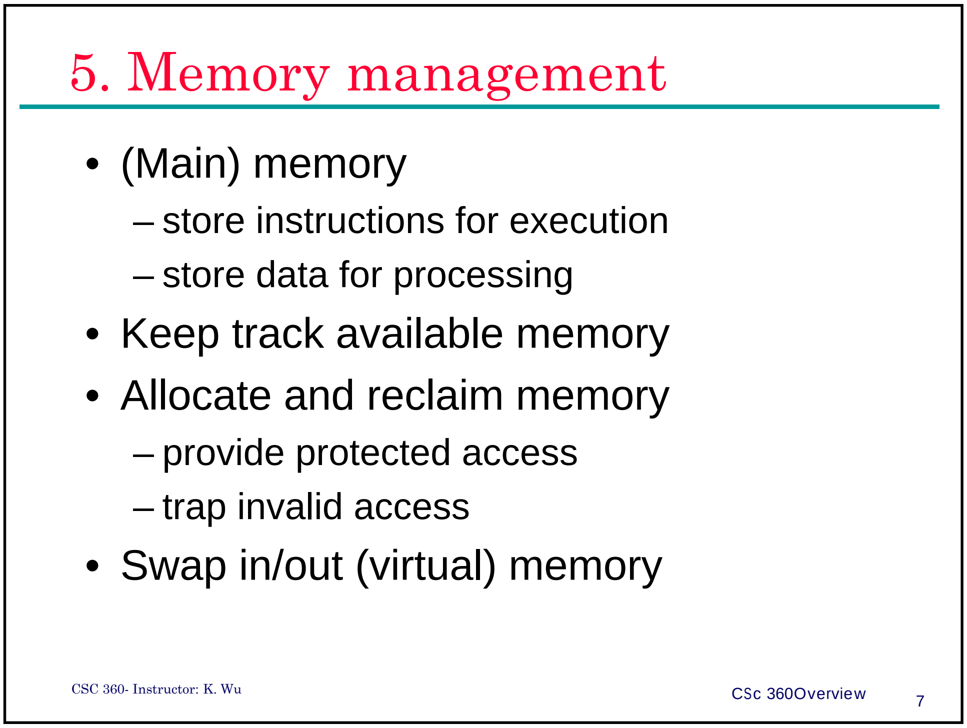
- Interrupt the current job
  - *yield*: system call trap (e.g., I/O)
  - *yank*: hardware timer interrupt
  - how about an “abusive” job?
- Dual-mode operation
  - user mode for regular applications
  - kernel mode with privileged instructions
  - trap: user to kernel entry





4. Process Management

- Process: a running program
  - vs thread
- Create, delete, suspend, resume process
  - resource allocation: CPU, memory, I/O, etc
- Schedule processes/threads
- Synchronize processes
- Communicate between processes
- Handle deadlocks



5. Memory management

- (Main) memory
  - store instructions for execution
  - store data for processing
- Keep track available memory
- Allocate and reclaim memory
  - provide protected access
  - trap invalid access
- Swap in/out (virtual) memory

6. Storage and I/O Management

- "In Unix, everything is a file"
  - a logical interface: open, read, write
- Create and delete files and directories
  - directory is a special file
  - file system hierarchy
- Manipulate files and directories
  - provide protected access
  - handle device-specific issues (disks, etc)

7. User Management

- Authentication
  - who's who
  - user credentials (e.g., password, token)
- Authorization
  - what can do what
  - access control (e.g., read, write, execute)
- Accounting
  - what has been done (e.g., logging)