# Midterm Preparation

CS3113

# Exam Mechanics

- When: Tuesday, October 8th, 1:30-2:45
- Wait until you are invited into the room
- Find an unclaimed seat with an exam
- One page of notes is allowed.
  - 8.5x11 paper (double-sided is fine). Typed or handwritten. No Magnification instruments.
- Scratch paper is allowed
- No electronic devices.
  - Including calculators, watches, iwatches, phones, laptops, Tamagotchis, tablets, ...
- Contact drc.ou.edu for appropriate accommodations (drc@ou.edu).

# Exam Mechanics

- Multiple choice
- Coverage will be theory to practical programming
- No generation of code
- But: many questions will be about code
  - Here is code, what does it output?
  - Here is what the code is supposed to do + the code; where is the bug?
  - -> Need to know your API

# Topics

- Byte-level representations and pointers
- Compiler vs linker + Makefiles
- Bit-wise operators in C
- System calls
- Streams
- Files and File Systems
- Processes
- Threads

# Byte-Level Representations and Pointers

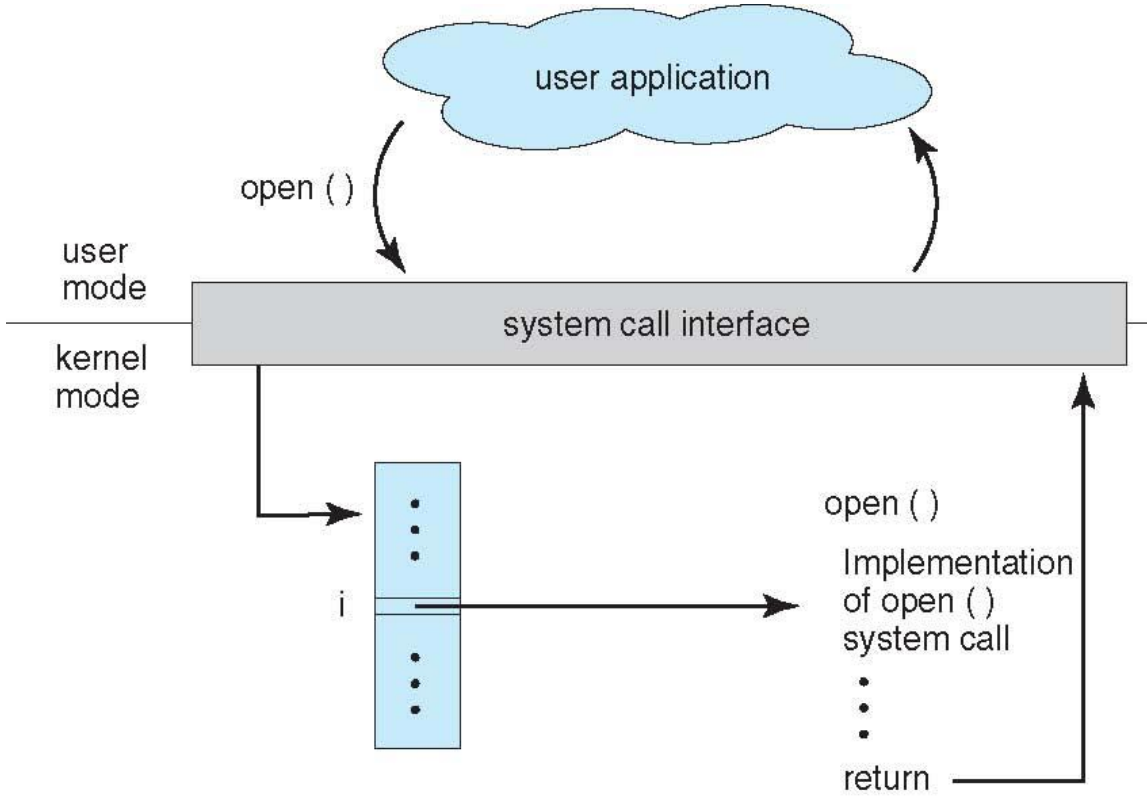
- Variable values vs pointers to values
- Array vs a pointer
- Representation of primitive types
  - char, int, float
- Strings
  - Null termination, strcpy() strcmp(),
- Structs (and pointers to structs)
- memset(), memcpy(), scanf()

# Compiler, Linker, Makefile

- Distinction between compiler & linker
  - What files do they take as input & generate as ouput
- Makefile
  - What do the rules mean?
  - Defining variables inside a Makefile

# System Calls

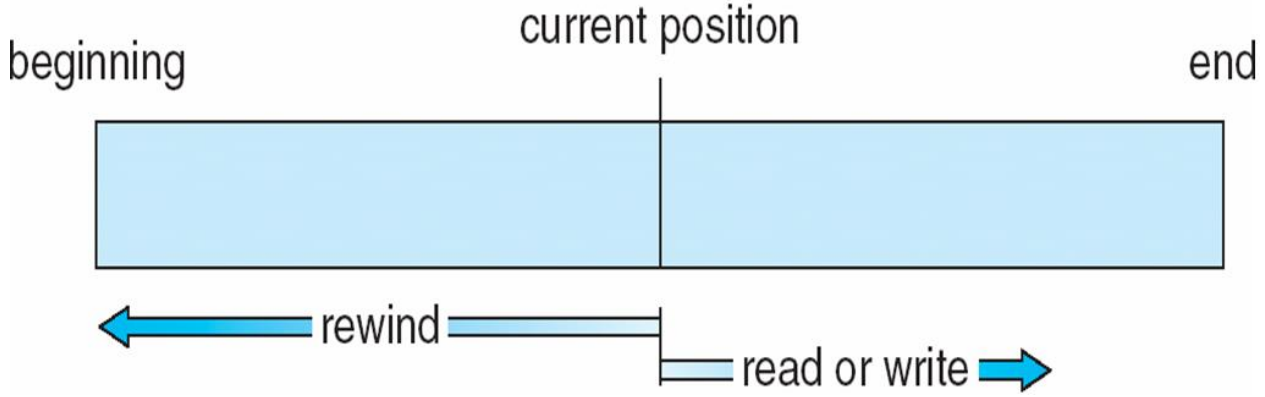
- User mode vs kernel mode
- System calls: allow user program to access kernel-level resources
- Switching from user to kernel mode
  - Table look-up for finding the right kernel-level function  to execute
  - Switching always involves overhead (more than a function call)



# Streams

Array of bytes

- Well defined beginning and ending
- Offset: the current point of access
- Read and write operations
- In some cases: can also seek

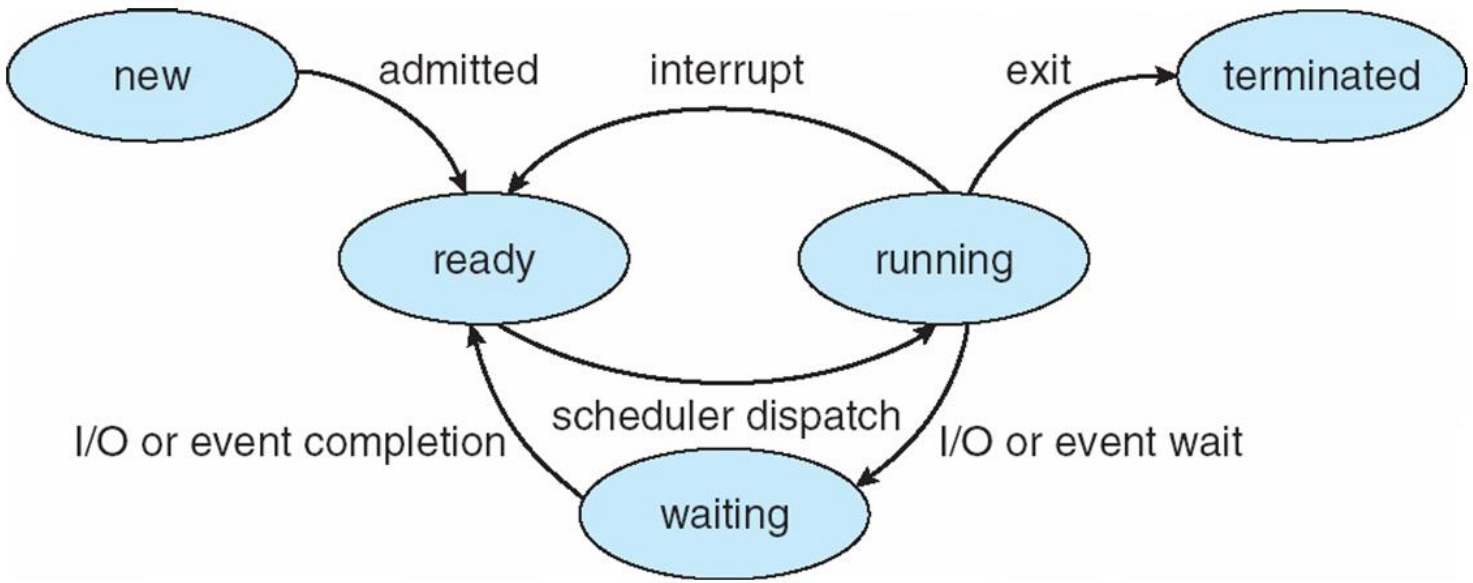


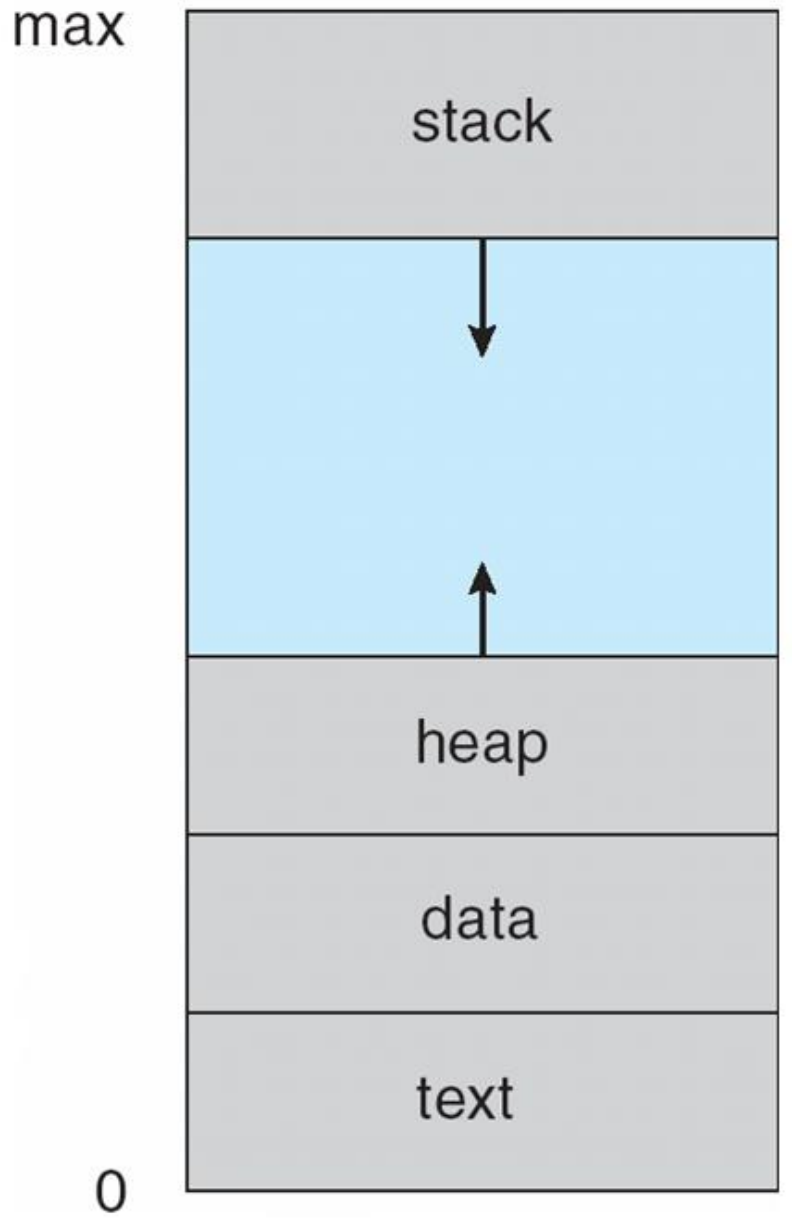
# Files and the File System

- Directory hierarchy
  - Absolute vs relative paths
  - Current working directory
- A file is a stream that lives on a disk (or some other storage)
  - open(), close()
  - read(), write()
  - lseek()
- File attributes
- FILE
  - fopen(), printf(), fprintf(), scanf(), fscanf()

# Processes

- Memory space of a process
  - Heap vs stack
- Process states

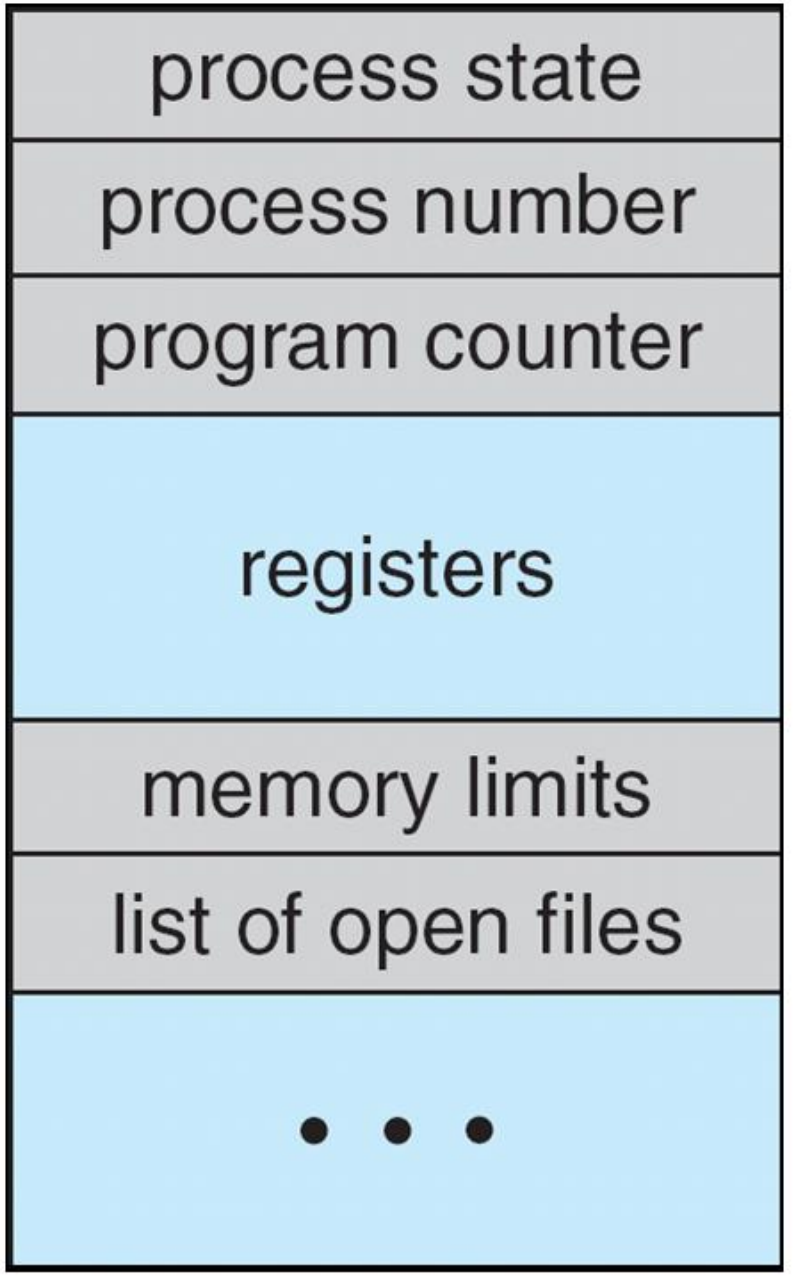


# Processes

Process control block:

- Kernel data structure
- Contains all of the information required to manage the process, including moving it on/off the running state



# Threads

Minimal coverage of threads

- What is the distinction between a process and a thread?
  - Threads share memory (globals, heap), program spaces
  - Threads have their own stack & registers

# Preparing

- Make sure you understand these key concepts
- Lecture notes
- Assigned readings
- Quizzes
- We have also done many coding examples in class
  - Review these: focus on the functionality (but keep careful notes on syntax)
- Prior exams: see the ***prior classes*** section of my home page