Midterm Preparation

CS3113

Exam Mechanics

- When: Tuesday, October 13th, 3:00-4:15
- Connect to the class Zoom
  - No cameras are required
- Open book; open notes
  - I suggest that you take time to write 1 page of quick-reference notes
  - Scratch paper is allowed
  - All class released class materials are fair game
- No calculating devices, including compilers
- Do not use other network resources
- Accommodations: if you haven't received email yet, please drop me a note

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 that we have been using

Topics

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

Byte-Level Representations and Pointers

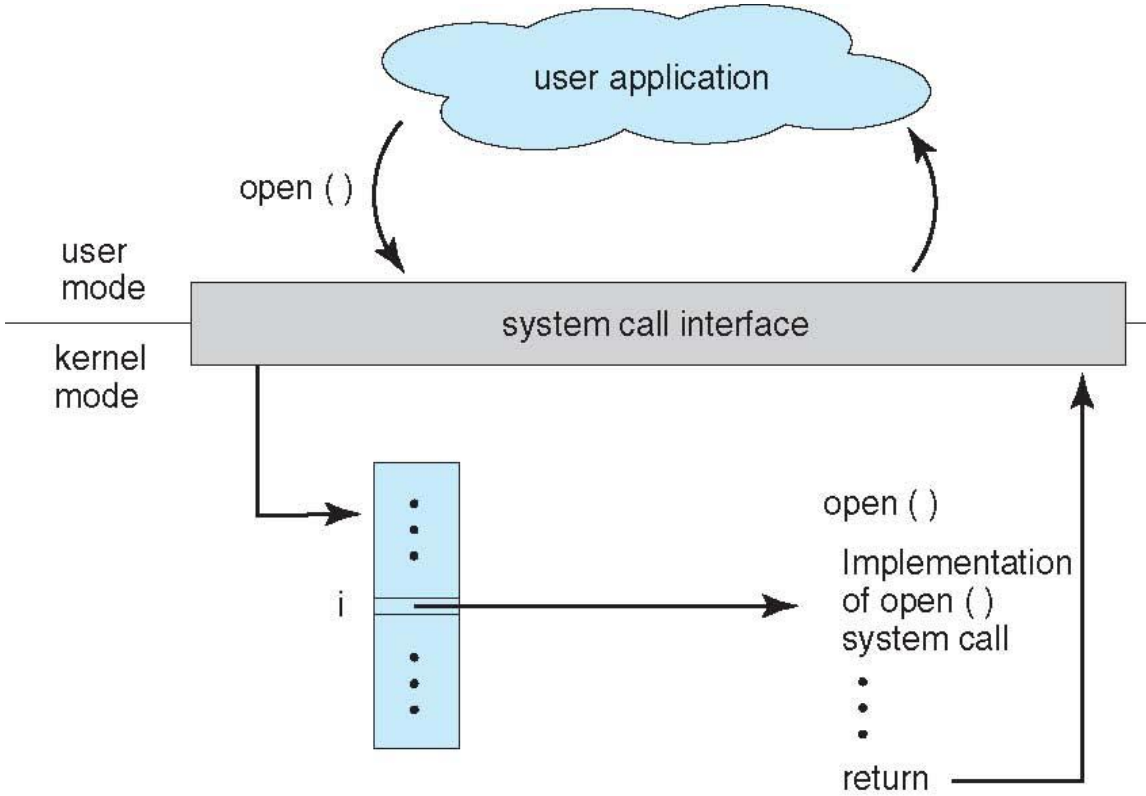
- Variable values vs pointers to values
- Array vs a pointer
- Representation of primitive types
  - char, int, float
- Strings
  - Null termination, strcpy() strcmp(),
- Structs (and pointers to structs)
- memset(), memcpy(), scanf()

Compiler, Linker, Makefile

- Distinction between compiler & linker
  - What files do they take as input & generate as output
- Makefile
  - What do the rules mean?
  - Tracing through a sequence of rules
  - 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)



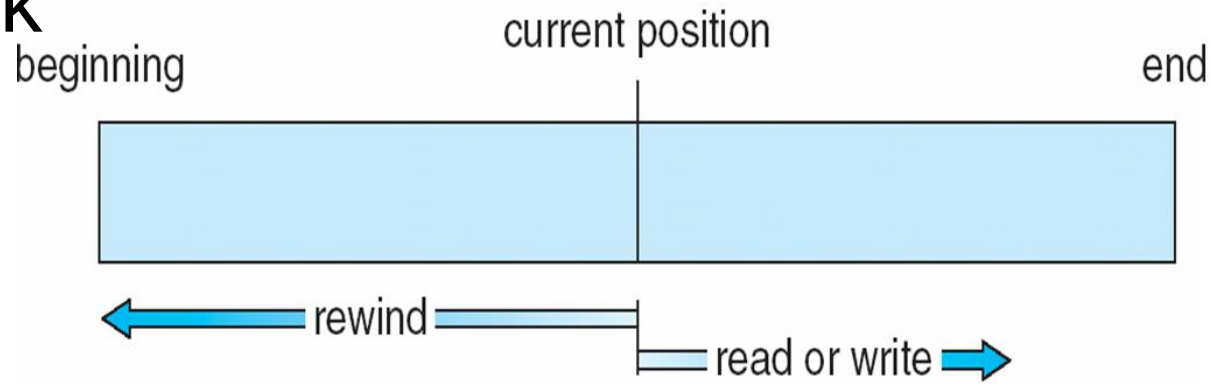
User vs Kernel Space Data

- Process-owned data structures include:
  - Stack
  - Heap
  - Global variables
- Kernel-owned data structures include:
  - Buffers connected to I/O devices and to the file system
  - Buffers for pipes
  - Process Control Block
  - Scheduler queues

Streams

Array of bytes

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

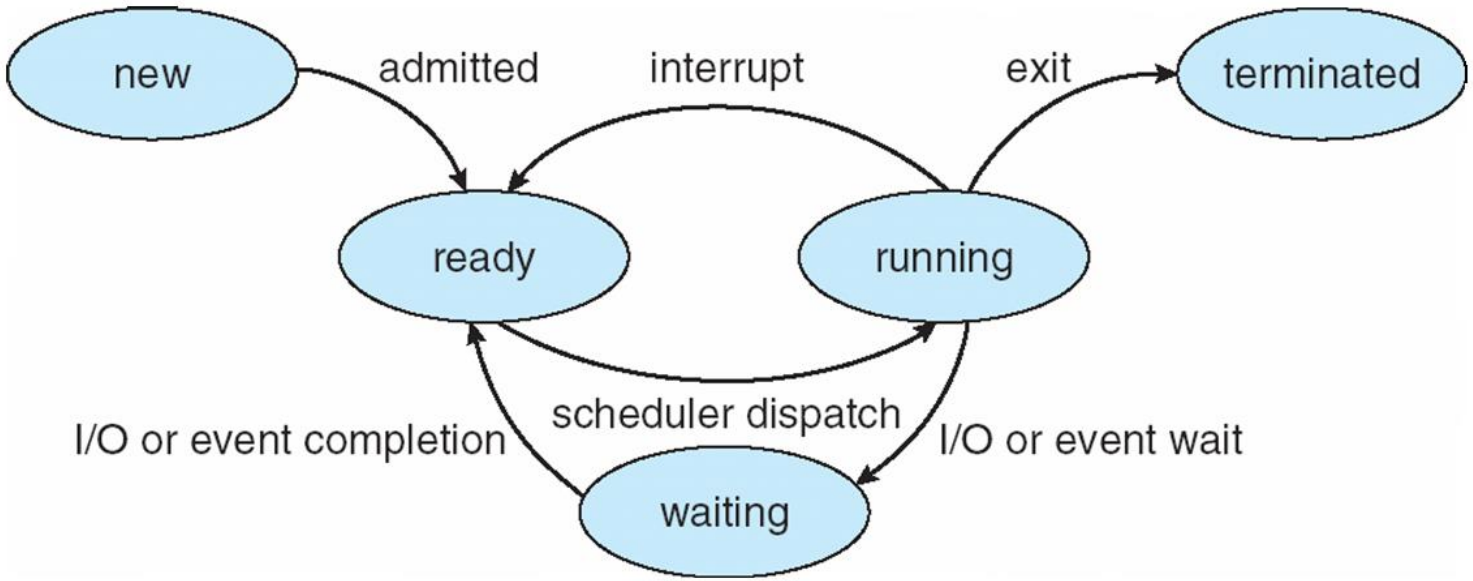


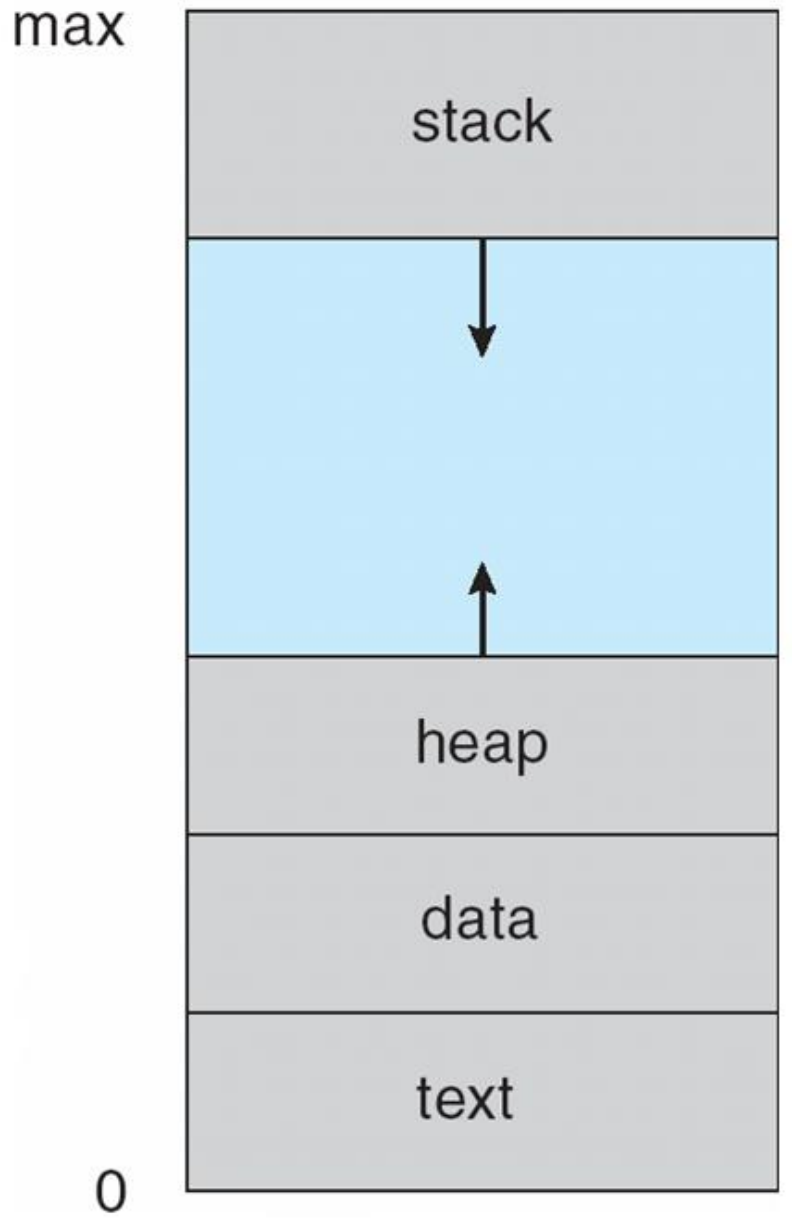
Files and the File System

- 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

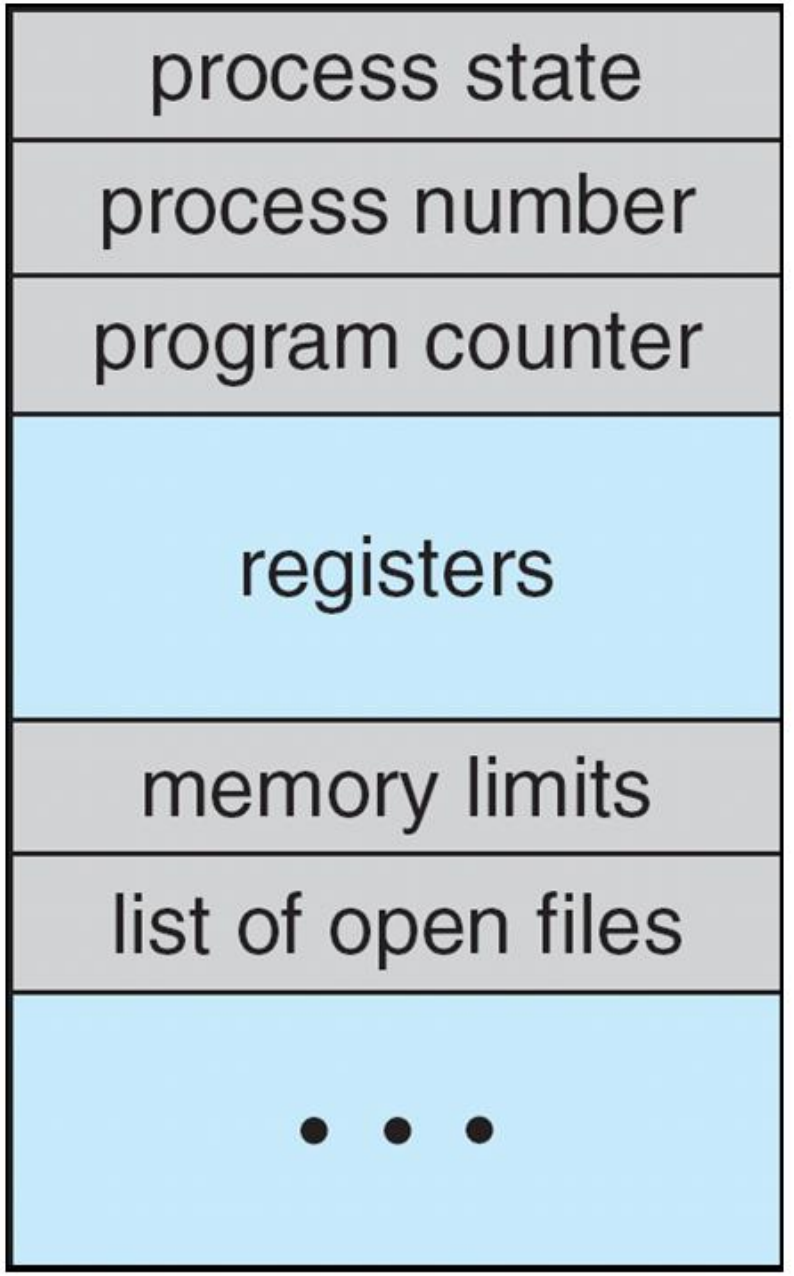




Processes

Process control block:

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



Preparing

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