# COSC 361 -- Operating Systems -- Fall 2022

## Class Time: 10:20 am - 11:10 pm MWF
## Classroom: MKB 524

---

### Instructor: Micah Beck

Office: Min Kao 433
Office Hours: By appointment
Email: mbeck@utk.edu

### TA: Yi Wu

TA Office Hours: Make an appointment by email.
TA email: ywu83@vols.utk.edu

---

### Textbooks

- **(Required) Operating Systems: Three Easy Pieces (OSTEP)**
  Authors: Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
  Edition: 1.0
  This book is available for free at http://pages.cs.wisc.edu/~remzi/OSTEP/

- **(Optional) zyBooks: Operating Systems**
  - Click any zyBooks assignment link in your learning management system (Do not go to the zyBooks website and create a new account)
  - Subscribe

### OSTEP Reading Assignments
**(see Canvas for optional zyBooks assignments)**

| Date | Assignment |
|---|---|
| 8/24 | OSTEP Ch. 2 |
| 9/2 | OSTEP Ch. 4 |
| 9/7 | OSTEP Chs. 5 and 6 |
| 9/12 | OSTEP Ch. 7 |
| 9/21 | OSTEP Ch. 9 |
| 9/26 | Unix Pipes and Signals |
| 9/28 | OSTEP Chs. 13 and 14 |
| 9/30 | OSTEP Chs. 15 and 18 |
| 10/3 | OSTEP Ch. 19 |
| 10/6 | OSTEP Ch. 20 |
| 10/14 | OSTEP Chs. 21 and 22 |

| | |
|---|---|
| 10/19 | Notes on `vfork()` |
| 10/24 | OSTEP Ch. 26 |
| 10/31 | OSTEP Chs. 27 and 28 |
| 10/4 | OSTEP Ch. 29 |
| 10/9 | OSTEP Ch. 30 |
| 10/11 | OSTEP Ch. 31 |
| 11/28 | OSTEP Ch. 39 |
| 11/30 | OSTEP Ch. 40 |

## Homework Assignments & Projects: see Canvas

The project consists of an installation step and three programming parts:

- Students should follow the installation instructions found here (from a class at NYU): https://gcallah.github.io/OperatingSystems/xv6Install.html to install the xv6 simulator toolchain on your laptop.
- The first project will be the Lottery Scheduling project from the Operating Systems in Three Easy Pieces site: https://github.com/remzi-arpacidusseau/ostep-projects/tree/master/scheduling-xv6-lottery
- The second project will be the Virtual Memory project from the Operating Systems in Three Easy Pieces site: https://github.com/remzi-arpacidusseau/ostep-projects/blob/master/vm-xv6-intro
- The third project will be the Kernel Threads project from the Operating Systems in Three Easy Pieces site: https://github.com/remzi-arpacidusseau/ostep-projects/blob/master/concurrency-xv6-threads

## Annotated Course Syllabus (Under revision!)

- Processes [2 (code)]
- Virtualization [4] [5 (code)]
  - What are the reasons for using libraries?
  - What is a protected resource?
  - What is a manager?
  - How is virtualization used to manage resources?
- Multitasking [6]
  - What is a process? How does it differ from a task or a thread?
  - What are the elements of process context?
  - How is the process/task control block used in operating systems?
  - What are the steps in kernel/user and user/user context switches?
  - How are runnable processes managed, and how is the next to run chosen?
  - How are processes created?
- Scheduling [7] [8] [9 (code)] [10]
  - What is the CPU-I/O burst cycle?
  - What is the function of a CPU scheduler? When are scheduling decisions made?
  - What are the usual criteria used in making scheduling decisions?
  - What are the criteria used in making preemptive scheduling decisions?
  - What is the ideal or optimal scheduler? Why can't it be implemented?
  - What is the definition and important attributes of these CPU scheduling algorithms: FCFS, SJF, priority, RR and multilevel?
- Memory [13] (code)] [14]
- Virtual Memory [15 [16] [17]
  - What are the definitions and uses for swapping, paging and segmentation?

- Paging [18] [19] [20]
  - Describe a straightforward hardware implementation of paging.
  - What is a Translation Look Aside Buffer, and why is it important in paging?
  - How does a multi-level page table work? A hashed page table? An inverted page page table?
  - How is page management hardware used to support demand paging? copy-on-write?
  - Define and explain one strength and one weakness of random, FIFO, optimal, LRU and LRU approximation page replacement algorithms.
  - Explain one explain one LRU algorithm and the second chance LRU approximation algorithms, and compare their use of resources.
  - What are the causes of thrashing and what can be done to avoid it?
  - Explain three advantages of memory-mapped files.
- Threads [26 (code)] [27 (code)]
  - What are the major reasons for programming with threads?
  - What is the difference between a thread and a (heavyweight) process?
  - What are the different modes of support that an OS kernel may provide for threads?
  - What are the strengths of preemptive vs. non-preemptive threads?
  - How are threads used in the programming of multicore architectures?
- Mutual Exclusion [28 (code)] [29]
  - Why is mutual exclusion important in preemptive and parallel systems?
  - What are the requirements for a solution to the critical section problem?
- Concurrency [30 (code)] [31 (code)] [32]
  - What are Semaphores and Monitors and how are they used for process synchronization?
  - What are Deadlock and Priority Inversion?
  - Explain the Bounded Buffer and Dining Philosophers problems and their solutions.
- Storage [36] [37]
- File Systems [39] [40] [41]
  - How is the inode used to implement sequential files?
  - Why does the use of a memory cache raise the possibility of inconsistent file system structures on disk in the case of a system crash?
  - What is read-ahead and why is it central to file system read performance?
  - Why are Unix directories a performance bottleneck while storing large numbers of larges?

---

## Course Requirements and Grading

- There will be two midterm exams and a final exam.
- There will be programmng projects.
- Homework will be assigned on a regular basis.

The final course grade will be calculated as

$$10\%(H) + 25\%(P) + 12.5\%(M1) + 12.5\%(M2) + 40\%(F)$$

where

- H = MAX(homework grade, zyBooks participation)
- P = MAX(project grade, final exam grade)
- M1 = MAX(midterm 1 grade, final exam grade)
- M2 = MAX(midterm 2 grade, final exam grade)
- F = final exam grade

For example, a student who scores 65 on midterm 1, 90 on midterm 2, 85 on the projects, 84 on the final, and 75 on the homework will have an overall score of

$$10\%(75) + 25\%(MAX(84,85)) + 12.5\%(MAX(84, 65)) + 12.5\%(MAX(84, 90)) + 40\%(84) = 84.15$$

The intention of this grading scheme is that students have two chances to show their mastery of the material covered in the midterm: on the midterm (or projects) and on the cummulative final. The 90% of the course grade that is awarded on the basis of exams (and projects) is available to every student at the time they take the final.

---

**NOTE:** The "total grade" calculated and automatically displayed on Canvas does **not** correspond to the above formula and does **not** represent your total grade that you will be assigned in this course. If you have any questions about the above formula you should contact Dr. Beck directly.

---

While it is possible for a student to skip the midterm exam and rely solely on the homework, project and final for their course grade, students are strongly advised against this approach. Here are some reasons for this advice:

- Taking the midterm exam is a no-risk proposition: it can only improve a student's final course grade.
- The midterm is more focused, and so it should be easier to study the material covered deeply and get a good grade.
- There is less presure on students when taking midterms than during the final exam period.
- Taking the midterm is the best preparation for the final exam.