CS 452/552: Operating Systems

Instructor

Instructor:	Jim Buffenbarger
Office:	CCP-359
Voicemail:	(208) 426-3567
Email:	mailto://buff@cs.boisestate.edu
WWW:	http://csweb.boisestate.edu/~buff

Meetings

Lectures:	TuTh 03:00–04:15	CCP-221
Office hours:	TuTh 11:45–12:45	CCP-359
	by appointment	CCP-359

Our Teaching Assistant is Adam Torek:

`adamtorek@u.boisestate.edu`

TA office hours are at (this "`pub`" directory is explained, below):

`pub/TutorOfficeHours`

I am happy to answer questions by email. Please please click here, or see:

`pub/doc/EmailQuestions.pdf`

Catalog Description

Operating systems structure and design. Process management, concurrency and synchronization, interprocess communication, scheduling, device management, memory management, file systems and security. Case studies of multiple operating systems.

PREREQ: CS 230, CS 253, CS 321, ECE 330, and CS 155.

Goals

Students are introduced to basic concepts of operating systems, including:

- using processes and threads
- operating system organization
- computer organization
- device management
- implementing processes and threads
- scheduling
- synchronization
- interprocess communication
- deadlock
- memory management
- virtual memory
- file management
- security
- networks

Textbook

Operating Systems: Three Easy Pieces, by A. Arpaci-Dusseau and R. Arpaci-Dusseau, 2018, online.

Other Course Material

This syllabus, lecture slides, assignments, and other material is available in what we'll called our "`pub`" directory. It is available in three places: GitHub, Canvas, and the `onyx` cluster of computers. This directory is read-only. So, you might want to copy it, perhaps to your local computer.

The GitHub `pub` directory can be accessed at GitHub, as shown in:

`pub/GitHub`

The Canvas `pub` directory can be accessed from our Canvas website, via the "Files" tab on the left sidebar.

The `onyx` `pub` directory can be accessed directly, by computers in our Computer Science Lab (CCP-240, CCP-241, and CCP-242).

Since `onyx.boisestate.edu` services Secure Shell (SSH) requests, you can also use SSH clients (e.g., `scp` and `sftp`) to access this `pub` directory remotely. However, beware: It contains symbolic links to parent directories, and `scp -r` will unconditionally follow them, thereby looping forever. To avoid this, use `sftp` or `tar`/`scp`, as needed.

In any case, the `onyx` `pub` directory is at:

```
~jbuffenb/classes/452/pub
```

The `onyx` cluster also has the advantage of containing all of the translators we will use.

Grades

At the end of the semester, the Registrar requires a letter grade for each student. Accordingly, during the semester, homework and exams are evaluated, and numeric scores are assigned. Each such artifact is worth a certain number of points, and has a weight. From these scores, a student's overall numeric/raw percentage is computed.

Collectively, a class's raw percentages form a distribution, with a mean and standard deviation, sometimes called a Bell Curve. An algorithm normalizes these values into a "grading" distribution, with a desired mean and standard deviation of 85 and 10 (i.e., the values are "curved"). A normalized value is then mapped to a letter grade, in the conventional way. If you are interested in the gory details, click here.

During the semester all of this evolving data can be found in our `pub` directory, on `onyx`, its nodes, Canvas, and GitHub.

Homework is due at 11:59PM, Mountain Time, on the day it is due. Late work is not accepted. To submit your solution to an assignment, login to a lab computer, change to the directory containing the files you want to submit, and execute:

```
submit jbuffenb class assignment
```

For example:

```
submit jbuffenb cs452 hw1
```

The `submit` program has a nice `man` page.

When you submit a program, include: the source code, sample input data, and its corresponding results.

Scores are posted in our `pub/scores` directory, as they become available. You will receive a code, by email, indicating your row in the score sheet. You are encouraged to check your scores to ensure they are recorded properly. If you feel that a grading mistake has been made, contact me as soon as possible.

The weights of homework and exams is shown in the table below:

Activity	*Weight*
Homework	80%
Exam	10%
Final	10%

Homework

Several homework problems are assigned during the semester. Each asks you to develop software in the C programming language. Students work on these individually.

The rubric that will be used to grade each assignment is distributed with the assignment. Try to focus on the assignments, rather than the rubrics.

Exams

An exam and a final are administered. These are in-class, open-note, open-textbook, and open-laptop tests. Students work on these individually.

Graduate-Student Grades

This is a dual-listed section: containing undergraduate and graduate students.

Graduate students are assigned extra work: additional assignments, or additional parts of existing assignments. It may take the form of software development, or research and in-class presentation. Details are determined by the

number of graduate students enrolled. In any event, rubrics will reflect this extra work.

For graduate students, the "grading" distribution, discussed above, is different. It has a mean and standard deviation of 80 and 10. The effect is that a graduate student is expected to "do better" than an undergraduate student.

Attendance

In-person lecture attendance is an important part of course participation. Attendance is taken at each lecture: starting five minutes before the scheduled start time, and ending fifteen minutes after the scheduled start time. Attendance is not taken during the first week of classes, holidays, or finals week.

Attendance can affect your grade. Each absence results in a one-percent reduction of your overall normalized percentage. Since a few absences are expected, completion of BSU's on-line end-of-semester course evaluation will erase up to five absences.

Attendance is administered wirelessly, via the iClicker app, available for free, from your smartphone's app store. For more information, click here.

Source-Code Documentation

Good documentation and programming style is very important. Your programs must demonstrate these qualities for full credit. Good documentation and programming style includes:

- heading comments giving: author, date, class, and description
- function/procedure comments giving description of: purpose, parameters, and return value
- other comments where clarification of source code is needed
- proper and consistent indentation
- proper structure and modularity

For more information, and examples, click here.

Academic Integrity

The University's goal is to foster an intellectual atmosphere that produces educated, literate people. Because cheating and plagiarism are at odds with that goal, those actions shall not be tolerated in any form. Academic dishonesty includes assisting a student to cheat, plagiarize, or commit any act of academic dishonesty. Plagiarism occurs when a person tries to represent another person's work as his or her own or borrows directly from another person's work without proper documentation.

If a student engages in academic dishonesty, the student may be dismissed from the class and may receive a failing grade. Other penalties may include suspension or expulsion from the University.

For much more information about academic integrity, including examples of academic dishonesty, please click here. If you are unsure about a particular behavior, ask your instructor.

Labs and Safety

Each student receives an account on the cluster of computers in the Computer Science Labs: CCP-240, CCP-241, and CCP-242. The cluster comprises a server named `onyx.boisestate.edu` and a set of nodes with shared home directories. It is remotely accessible, via SSH. The cluster runs the Linux operating system.

Physical access requires building and room access. After-hours building access, and all-hours room access, require an authenticated proximity-type student-identification card.

You are responsible for understanding and obeying lab rules.

Schedule

Week	*Date*	*Topic*	*Assigned*	*Due*	*Reading*
1	Jan 09 Tue	Introduction			1-2
	Jan 11 Thu	Processes and Processors	HW1		3-11
2	Jan 16 Tue				
	Jan 18 Thu			HW1	
3	Jan 23 Tue		HW2		
	Jan 25 Thu				
4	Jan 30 Tue				
	Feb 01 Thu				
5	Feb 06 Tue	Memory	HW3	HW2	12-24
	Feb 08 Thu				
6	Feb 13 Tue				
	Feb 15 Thu				
7	Feb 20 Tue	Concurrency		HW3	25-34
	Feb 22 Thu				
8	Feb 27 Tue		HW4		
	Feb 29 Thu				
9	Mar 05 Tue	Exam			
	Mar 07 Thu				
10	Mar 12 Tue				
	Mar 14 Thu				
11	Mar 19 Tue	Spring Break			
	Mar 21 Thu	Spring Break			
12	Mar 26 Tue			HW4	
	Mar 28 Thu				
13	Apr 02 Tue	Input/Output Devices	HW5		35-42
	Apr 04 Thu				
14	Apr 09 Tue				
	Apr 11 Thu				
15	Apr 16 Tue				
	Apr 18 Thu			HW5	
16	Apr 23 Tue				
	Apr 25 Thu				
17	May 02 Thu	Final: 12:00-2:00			