# CS 333

Introduction to Operating Systems

Introduction to the principles of operating systems and concurrent programming. Operating system services, file systems, resource management, synchronization. The concept of a process; process cooperation and interference.

## CS 333 – Introduction to Operating Systems

**Location**: In Zoom
**Lecture**: Monday and Wednesday, 11:30m – 1:30pm.
**This is an "Online Scheduled Meetings"** class, as defined by Portland State University's course delivery methods. The description for an Online Scheduled Meetings class is "Online courses that include required meeting times." **For this class, attendance to the lecture periods is required**. Class lecture material will be presented via Zoom. I do not plan to record the lecture for later viewing. In addition, there will be in-class activities which will be provided during the lecture time and will be due at the end of the lecture time. This is not a fully online class with all recorded content.

**Instructor**: R. Jesse Chaney
**Office**: FAB 120-13 (Fourth Avenue Building)
**Email**: rchaney@pdx.edu
**Phone**: 503-725-9815
**Office hours**:
- MW: 2:00pm to 4:30pm (**on Zoom**, I will be in my office on Wednesdays)
- TTh: 4:00pm-5:00pm (in person, FAB 120-13)
- Tuesday: Code Party 6:00pm to 11:pm location: TBD
- By appointment (email me).
- Occasionally, Office Hours are superseded by other university meetings.

**Course Web site:** Canvas ( https://canvas.pdx.edu/ )
**Prerequisites**: CS 201 and CS 202. You must have passed both CS201 and CS202.
**Recommended Text:**
**THE LINUX PROGRAMMING INTERFACE** by Michael Kerrisk ISBN 978-1-59327-220-3.

The Linux Programming Interface (**TLPI**) is the definitive guide to the Linux and UNIX programming interface—the interface employed by nearly every application that runs on a Linux or UNIX system.
- You will see this referenced as **TLPI** in the recommended reading assignments.
- No Starch Press link to book.
- Amazon link to book.
- PSU Bookstore link to book.
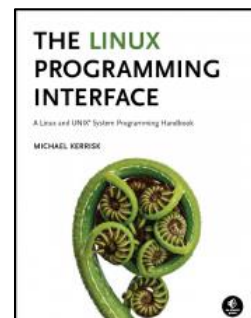
There may be supplemental/supporting reading material, freely available online.

### Rationale & Primary Learning Outcomes

To expose students to the fundamental architectural and algorithmic concepts involved in the design and implementation of modern operating systems. This will include the design and coding of concurrent programs.

Upon the successful completion of this class, students will be able to:
1. Read technical material that refers to operating system concepts.
2. Write system-level code.
3. Apply synchronization-related concerns when implementing code.
4. Continue on to study advanced operating systems.

5. Describe observed application runtime behavior.

**The perquisites should have given you an introduction to programming in C**. This includes syntax, flow control constructs, basic function usage, and pointers. **Basic UNIX/Linux knowledge is also assumed**. Please see resources page for helpful information if you feel lacking in these areas.

## Reading

Reading assignments extend the lessons explored in class and immerse you in the world of Unix/Linux development. **You are encouraged to have completed the recommended reading for a class in advance of the class meeting time so that you can actively participate in the classroom discussions**. Your active participation will greatly enhance learning the material.

## Communication

Any email sent to the instructor about this course must originate with a Portland State University supplied email account. Failure to comply with this will result in lengthy delay for the response to your email.

The best form of communication outside of class is for students to come into office hours. I have established a number of office hours during the week.

My preferred communications, outside of class, are:
1. office hours,
2. appointment,
3. email.

## Grading

- Quizzes/Activities           15%
- Video Assignments            15%
- Final Exam                   25%
- Programming Assignments      45%

| PSU Grading Scale | | | |
|---|---|---|---|
| 93-100 | A | 73-76 | C |
| 90-92 | A- | 70-72 | C- |
| 87-89 | B+ | 67-69 | D+ |
| 83-86 | B | 63-66 | D |
| 80-82 | B- | 60-62 | D- |
| 77-79 | C+ | 0-59 | F |

## Attendance

This is an on-campus and in-person class. Attendance to class is required. Attendance and participation are important factors that contribute to what you will get out of the class.

## Quizzes/Activities

There will be a number of quizzes in Canvas for this class. The Canvas quizzes have a due date associated with them. **If you submit the quiz after the due date, I will treat it as a zero for that quiz**. Once you have taken a quiz, don't take it again after the due date. The quiz score used to calculate your class grade will be the highest score for each quiz, taken before the due date. Therefore, there is no penalty for taking a Canvas quiz more than once, I will use the highest score for each (on time) Canvas quiz.

The Activities are short work sheets you'll be given during class (typically the last 20 to 30 minutes of class). You'll work on these in groups of 2 or 3. The worksheet will generally have 5 to 10 questions which will relate to recent class material. You'll need to complete and turn in the worksheet before you leave the classroom for the day. The names of the students who worked on your assignment should be on the paper turned in.

**Video Assignments**

Video Assignments fill the role of live coding examples during class. Most Video Assignments will be recorded videos. Some of the Video Assignments may actually be live coding sessions in class. For the videos, I will explain each step of step the code. Some of the Video Assignments are made from more than a single video. What you need to do to receive 100% on an Video Assignment is follow along, write the code, make sure it works as expected, and submit it into Canvas. You should ask questions as they arise. Asking questions is an important part of learning the material. Submitting an Video Assignment that does not compile will typically receive a grade of zero. Go ahead, **compile it and test it**.

Video assignments are due as posted in Canvas. Video Assignments that are late are penalized 20% per day. Any Video Assignment submitted more than 2 days after the due date will automatically receive a grade of zero (0).

An objective of the Video Assignments is to present small, isolated examples of concepts we cover in class. We then take those examples and combine and expand them in the larger Programming Assignments. They are intended to be low stress. The code for a Video Assignment does need to be exactly the same as presented in the videos. However, the behavior of the code you submit and the code from the video must be exactly the same.

**Midterm Exam**
**There is not a midterm for this class**. Because there are so many quizzes and Video Assignments, I removed the midterm. I also found that there is a high correlation between midterm grades and final exam grades, so I eliminated the midterm.

**Final Exam**
**The final exam is scheduled for Thursday June 15th from 12:30pm to 14:20pm** (week 11 of the term). The final exam will be given through Canvas. As we get closer to the date of the final exam, more information will be available.

A makeup examination can be arranged only when a student has an emergency (e.g., medical emergency or urgent family matter) or an excused absence from the university (e.g., athletic events). The student must provide the instructor with an appropriate document, such as a doctor's note or official university document.

**Programming Assignments**
To complement the project-based learning, students will complete **individual** programming labs to focus on UNIX/LINUX development and understanding. You should expect to spend a significant amount of time working on the programming projects for this class. If you do not submit the programming projects for this class, you will fail this class.

**Programming Assignments that are late are penalized 20% per day**. Any programming assignment submitted more than 2 days after the due date will automatically receive a grade of zero (0). Don't make the mistake of submitting your assignment late just trying and get the last few points by making it perfect. Perfection is the enemy of done. You want to be done.

**Incompletes**

I hate incompletes. Incomplete (I) grades will be granted **only** in emergency cases (usually only for a death in the family, major illness or injury, or birth of a child), and if the student has turned in the majority of the points possible (in other words, usually everything but the final exam).

To be clear, I hate incompletes because they so frequently turn into an F. Too often, I've seen students get an incomplete only to take a full load of classes the following terms and never complete the work from the incomplete class. The incomplete is always put off until it finally turns into an F. I've had to change several incompletes into F's and I've also turned a couple into A's; mostly F's.

Do not skip all the programming assignments, come to me at the end of the term and ask for an incomplete. You'll receive an F.

If you are having any difficulty that might prevent you completing the coursework, please don't wait until the end of the term; let me know right away.

**Class Structure**

This course is not based on passive learning and the classic lecture format. The class will be heavily discussion oriented, giving you time to participate in the discussion and work on the InClass assignments. Our classroom discussions should be lively, but must also be courteous and respectful.

I expect you to keep up with your email and postings and changes to the class Canvas site, in addition to attending and participating in class lectures.

**Code Party**

I have a group study session during the week in the evening. I prefer to call the study sessions **Code Party** (indicating that they are fun). These will occur on Tuesday evening from 6pm to 11pm. The location will be communicated during class. Code Party will be a little different this term, as I have a class overlaps the Code Party time.

Attendance to code parties is completely optional. However, it is a time when you can find me and your classmates working on the class material. There may be more than one class in the room during a code party.

**Course Schedule**

This schedule is an **estimate** and is subject to change according to the actual progress of the course.

| Week | Topics |
|------|--------|
| 1 | Syllabus, man pages, editors, C programming, argc/argv, getopt, shell |
| 2 | malloc, make and Makefiles, TLPI chapters 1, 2, 4 |
| 3 | TLPI chapters 5, 10, 20, 21 |
| 4 | TLPI chapters 24, 25, 26, 27, 43, 44 |

| 5 | TLPI chapters 3, 8, 9, 29, 30, LLNL Threads tutorial |
|---|---|
| 6 | TLPI chapters 31, 32, 53, 54 |
| 7 | TLPI chapters 56, 57, 58, 59, 60 |
| 8 | TLPI chapters 14, 19, 37, 52 |
| 9 | bash |
| 10 | TLPI 12, gprof, curses (maybe) |
| 11 | **Final Exam** |

**Collaboration and Academic Integrity**

Students are prohibited from handing in work as their own which they did not create. This includes handing in assignments in which a substantial amount of the material was done by someone else. Students need to be especially careful that in the process of discussing problems with other students they do not inadvertently end up using someone else's work. Similarly, failing to cite a source that contributed substantially to the solution of a problem is also considered to be cheating. It is not necessary to cite the textbook for the course on your homework, other than for direct quotes. All other sources should be referenced precisely. In the event a case of cheating is discovered, the student will automatically receive a score of zero (0) for that assignment or exam. Additional penalties may be applied by the Department, College, or University.

**In this course you can:**
- Study with other students.
- Describe your algorithm, problem solving strategy, or ideas.
- Help each other with debugging commands.
- Diagram and discuss your proposed design for assignments.
    - Especially helpful for making sure you've covered all the requirements.
- Help each other on slack in the **public** channel.
    - Do not use private channels or DMs.
- Help each other understand error output in the test suite, how to grok the output files.
- Consult blog posts, tutorials, etc. and you must cite them in your code.
- A "references" section at the top of your file with a list of sources is perfectly fine.
- Freely discuss style and formatting with examples (as long as those examples are not from your homework)

**Examples of Academic Misconduct**

Please note the following examples of what is considered inappropriate.
- Viewing another student's quiz, test, paper, or code while working on your own.
- Directly providing another student a copy, electronic or otherwise, of your work.
- Accepting a copy, electronic or otherwise, of another student's work.
- Copying and pasting any component of another student's work into your own.

- Copying solutions found online or otherwise, pasting it into your own work without proper citation.

These scenarios will be considered as academic misconduct except when involving an assigned project partner.

### CS Tutors
The tutors are an excellent resource for debugging, understanding core concepts, explaining man pages, version control, tools (GDB, CLI), and related issues. They may not help you with your assignment directly.

### Computer Action Team (CAT)
The CAT can help you with the CS Linux server, MCECS accounts, and connectivity issues.

### Accommodations
Accommodations for students with disabilities are determined and approved by Disability Resource Center (DRC). If you, as a student, believe you are eligible for accommodations but have not obtained approval please contact DRC immediately at 503-725-4150, drc@pdx.edu, or https://www.pdx.edu/disability-resource-center. DRC notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations.

I want to make this class an open and welcoming environment for all. Your success is my goal.

### Religious Observance
Portland State University strives to respect all religious practices. If you have religious holidays that conflict with any of the requirements of this class, please see me immediately so that we can make alternative arrangements.

### Reach Out for Success
As a student you may experience a range of issues that can cause barriers to learning, such as strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may lead to diminished academic performance or reduce a student's ability to participate in daily activities. Portland State University is committed to advancing the mental health and well-being of its students. If you or someone you know is feeling overwhelmed, depressed, and/or in need of support, services are available. You can learn more about the broad range of confidential mental health services available on campus via https://www.pdx.edu/health-counseling/.

SHAC also has resources for physical health, including flu shots. You can check out their COVID-19 resources page here: https://www.pdx.edu/health-counseling/covid-19-resources (including testing).

PSU Student Health and Counseling is staffed with folks who care and can help with a wide range of challenges. Here at PSU there is never a need to tough things out alone.

Get Food Now Here at PSU there is never a need to tough things out alone.  Those who can, give, so those who need, have.

Housing / financial crisis help. Here at PSU there is never a need to tough things out alone. Emergency Housing, etc.

**Title IX**

As an instructor, students frequently come to me for assistance in matters that are not related to the course material. Please be aware that PSU's policies require instructors to report any instance of sexual harassment, sexual and relationship violence and/or other forms of prohibited discrimination to University Officials, who keep the information private. If you would rather share information about these experiences with a PSU staff member who does not have these reporting responsibilities and can keep the information confidential, please contact one of the following campus resources.

- Confidential Advocates: 503.894.7982, or by scheduling on-line (for matters regarding sexual harassment and sexual and relationship violence)

- Center for Student Health and Counseling (SHAC): 1880 SW 6th Ave, 503.725.2800

- Student Legal Services: 1825 SW Broadway, (SMSU) M343, 503.725.4556 For more information, please complete the required student module Understanding Sexual Misconduct and Resources in Canvas.

PSU Sexual Misconduct Response website gives you comprehensive information about how to support and/or report an incident.

Please complete the required student module Understanding Sexual Misconduct and Resources in Canvas, which provides information about PSU policy and resources.

You may also report sexual and relationship violence to law enforcement on campus with Campus Public Safety Office (CPSO).

Or you may file an anonymous report with Campus Public Safety Office or a Bias Incident report with the Bias Review Team (BRT). PSU does not typically investigate the reports that are made through these two avenues. These reports help PSU understand what students and employees are experiencing on and around campus and provide support where needed.