Course Name: Operating Systems I
Course Number: CS 344
Credits: 4
Instructor Name: Ryan Gambord
Instructor Email: Ryan.Gambord@oregonstate.edu

Course description

Introduction to operating systems using UNIX as the case study. System calls and utilities, fundamentals of processes and interprocess communication.

Prerequisites

CS 261 [C] and (CS 271 [C] or ECE 271 [C])

Communication

This course uses:

- Canvas to host static course content and the gradebook. (No canvas inbox, discussions, etc.)
- Gradescope for submitting/grading assignments
- Ed discussions for the online classroom environment, announcements, etc.
- Ed Chat for informal instant messaging (not used for "official" announcements, etc)
- MS Teams for office hours with TAs
- Email for private or personal matters, including grading results and concerns.

Learning resources

All course materials are freely available online or through the school library.

- C99 standards (Freely available working draft ISO/IEC 9899:TC3) pdf [wiki (recommended)]
- POSIX.1-2008 pdf [html (recommended)]
Measurable student learning outcomes

Upon completion of this course, students should be able to:

1. **Justify** the need for a multi-programmed OS and explain the general structure of such systems.
2. **Select** system calls for appropriate uses.
3. **Compare** and **contrast** the process and thread abstractions and select the correct abstraction when needed.
4. **Assess** and **solve** possible issues related to concurrent execution.
5. **Explain** the file abstraction and system level I/O.
6. **Compare** and **choose** mechanisms for inter-process communication.
7. **Write** software by applying appropriate system programming principles and techniques.

Course content

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 0</td>
<td>Course Introduction - Software Configuration</td>
</tr>
<tr>
<td>Week 1</td>
<td>Introduction to OS, *nix and C</td>
</tr>
<tr>
<td>Week 2</td>
<td>Introduction to C</td>
</tr>
<tr>
<td>Week 3</td>
<td>Files and Directories</td>
</tr>
<tr>
<td>Week 4</td>
<td>Processes I</td>
</tr>
<tr>
<td>Week 5</td>
<td>Processes II</td>
</tr>
<tr>
<td>Week 6</td>
<td>Concurrency and Threads</td>
</tr>
<tr>
<td>Week 7</td>
<td>Inter-process Communication</td>
</tr>
<tr>
<td>Week 8</td>
<td>Network Programming</td>
</tr>
<tr>
<td>Week 9</td>
<td>Introduction to Rust (Optional content; not on final)</td>
</tr>
<tr>
<td>Week 10</td>
<td>Rust and Safe System Programming (Optional Content; not on final)</td>
</tr>
<tr>
<td>Week 11</td>
<td>Final Exam (asynchronous, unproctored, open book, open notes)</td>
</tr>
</tbody>
</table>
Evaluation of student performance

This course consists of five assignments (80%) and a final exam (20%). The final is unproctored, open book, but closed to any "interactive" resource that generates answers in response to queries (chatGPT, phone a friend, etc).

Letter grade

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93 - 100+</td>
</tr>
<tr>
<td>A-</td>
<td>90 - 92</td>
</tr>
<tr>
<td>B+</td>
<td>87 - 89</td>
</tr>
<tr>
<td>B</td>
<td>83 - 86</td>
</tr>
<tr>
<td>B-</td>
<td>80 - 82</td>
</tr>
<tr>
<td>C+</td>
<td>77 - 79</td>
</tr>
<tr>
<td>C</td>
<td>73 - 76</td>
</tr>
<tr>
<td>C-</td>
<td>70 - 72</td>
</tr>
<tr>
<td>D+</td>
<td>67 - 69</td>
</tr>
<tr>
<td>D</td>
<td>63 - 66</td>
</tr>
<tr>
<td>D-</td>
<td>60 - 62</td>
</tr>
<tr>
<td>F</td>
<td>0 - 59</td>
</tr>
</tbody>
</table>

Late work policy

Late work is not accepted. Extensions for events which substantially disrupt your ability to complete your work are given on a case-by-case basis.

Test scripts and example materials

Test scripts and other example materials may be provided to assist with understanding the expectations of an assignment. These materials should never be taken as sufficient evidence that an assignment is correctly implemented, or that it will receive any credit. Grading is carried out according to each assignment's specifications and rubric. You are responsible for verifying your work is correct.

Rubrics

Each rubric item is graded on a pass/fail basis with no partial credit within an item. Rubric items may list an example test or command, which is not intended to be interpreted as the exact test or command used in grading, but only as an example to clarify the type of test that might be performed.
Unfinished/Incomplete/Draft work

Submissions that fail to compile receive no credit. Submissions that produce unspecified output (debug statements, flourishes, greeting messages, extra whitespace, etc) may lose significant points on rubric items and will not be regraded. Submissions that fail to produce specified output may likewise fail rubric items and will not be regraded.

Unimplemented parts of assignments should be filled in with placeholder code so that your submission fails gracefully (exit with error, silently ignore input, etc.) without crashing. Crashing may result in significant point deductions.

Course Academic Integrity Policy

Rule of thumb: If you are copying and pasting (or manually retyping), you’re probably cheating.

- ALL submissions must be FULLY your own work.
- Submissions are reviewed for plagiarism using sophisticated detection techniques. These are the same tools used in forensic copyright infringement investigations.
- Confirmed violations will result in a 0 on the assignment in question, or an F in the course.
- Copying ANY code from unapproved sources (not course materials) is considered plagiarism.
  - Code citations are not a defense against plagiarism, and are often taken as an admission of guilt.
  - Attempts to obfuscate copied code are considered, additionally, cheating.
- You may be required to meet with the instructor to explain your code, if a violation is suspected.
- Sharing part or all of work with other students without authorization is considered cheating.
- Viewing solutions to assignments, whether directly or incidentally related to this course, is considered cheating.

ChatGPT and other AI tools

Using chatGPT and other AI tools to generate code is considered cheating. You may however use any of these tools to assist with reviewing your finished code for mistakes or other issues, to ask for help explaining a concept, etc. Please be cautious, as these tools are often wildly incorrect and often give wrong advice.

General exceptions

We understand that learning from others and from examples can be immeasurably valuable, and we want to support your learning and promote collaborative learning as much as possible, while also taking a firm stance on academic dishonesty.

You may use (directly copy) any code from course documents, examples on os1 man pages, and directly provided by instructional staff in your work. You may copy code from outside sources that are general in nature and idiomatic. Idiomatic code is the widely agreed on "standard" way of performing
certain routine actions. Idiomatic code is rarely more than 2-3 statements and performs a single discrete task that is generic, routine, and common to a variety of programs.

Some examples of idiomatic code would be:

- Calling a library function/system call and checking the return value for errors.
- Performing a looping read/write operation to flush a buffer.
- Checking the length of a formatted string, allocating memory for it, and then writing the formatted string to the allocated buffer.

Examples of non-idiomatic code that would be considered plagiarism if copied, whole or in part:

- A string search and replace function.
- A function that parses a file path into its constituent path components.
- Anything that might be described as an "algorithm"

You may share useful resources, general programming advice, high-level/structural approaches to assignments (no directly copied assignment code; edited minimal-working examples of bugs are allowed), input/output examples, etc. with other students. This specifically does not authorize you to copy any code directly from other students.

Consider that the course follows the same policy of most programming discussion and Q/A boards: You may ask and answer questions, including snippets of code as necessary to illustrate a point or question, but we are not here to do your homework for you.

Exceptions for students retaking the course

If you are retaking this course, you may re-use your code from a previous attempt (but you may not re-use code from other courses). Please make sure that your reused code complies with this term's academic integrity policies and assignment specifications, since both may have changed. Also, ensure that your code from a previous term has not been shared in any way with others (such as posting to a public github) -- you can be found guilty of assisting if another student plagiarizes your code. Contact the instructor(s) if you have any questions about this policy.

The unseen impacts of cheating and plagiarism

Far from a victimless offense, cheating directly harms everyone around you. Innocent students have been embroiled in academic integrity investigations because someone else plagiarized their work without their knowledge or consent. Cheating jeopardizes our ability to share solutions/walkthroughs to assignments in future terms, robbing students of a valuable learning resource. It also forces us to change assignments regularly, which reduces our time available to update other course content to benefit other students and can force us to switch to assignments with less instructional value. Investigating and reporting cheating also takes a considerable amount of instructional staff's time, which we wish we could apply towards teaching and improving content.
Communication and Conduct Policy

All university policies around student conduct apply in this course. Additionally, communication which is hostile or insults, demeans, bullies, or belittles the course, instructional staff, or other students is prohibited. This includes communication which insinuates or implies such things. You are expected to communicate professionally at all times -- if your communication would not be acceptable in a professional work environment, it is not acceptable here.

This policy is enforced strictly and swiftly.

Harassment

Everyone has the right to feel safe and respected in their workplace. This includes instructional staff. Do not harass instructional staff. Examples of harassment include:

- Flooding, or transmitting a similar message over multiple channels or repeatedly in the same channel.
- Engaging with a confrontational, overly demanding, argumentative, or hostile tone
- Repeated grading appeals that lack substance or effort (grade grubbing)
- Demands of a grade adjustment
- Public negative, hostile, or derogatory comments about the instructor or staff.
- Emotional/Trauma dumping (oversharing distressing personal information). We care about you, but we are not trained therapists and we cannot provide this level of emotional support to students. Share only what is necessary for us to complete our professional duties. We can always ask for more information if we need it.

Stop and think before you send: Is this something you would feel comfortable receiving from several dozen students on a daily basis? Would it be appropriate for them to do so?

Sanctions

First offenses will generally be given a warning. Repeat offenses may face sanctions including being banned from any or all course communication channels. Especially serious violations will be reported to the university without being given a warning.
Expectations for Student Conduct

Student conduct is governed by the university’s policies, as explained in the Student Conduct Code (https://beav.es/codeofconduct). Students are expected to conduct themselves in the course (e.g., on discussion boards, email postings) in compliance with the university's regulations regarding civility.

Academic Integrity

Integrity is a character-driven commitment to honesty, doing what is right, and guiding others to do what is right. Oregon State University Ecampus students and faculty have a responsibility to act with integrity in all of our educational work, and that integrity enables this community of learners to interact in the spirit of trust, honesty, and fairness across the globe.

Academic misconduct, or violations of academic integrity, can fall into seven broad areas, including but not limited to: cheating; plagiarism; falsification; assisting; tampering; multiple submissions of work; and unauthorized recording and use.

It is important that you understand what student actions are defined as academic misconduct at Oregon State University. The OSU Libraries offer a tutorial on academic misconduct, and you can also refer to the OSU Student Code of Conduct and the Office of Student Conduct and Community Standard’s website for more information. More importantly, if you are unsure if something will violate our academic integrity policy, ask your professors, GTAs, academic advisors, or academic integrity officers.

Statement Regarding Students with Disabilities

Accommodations for students with disabilities are determined and approved by Disability Access Services (DAS). If you, as a student, believe you are eligible for accommodations but have not obtained approval, please contact DAS immediately at 541-737-4098 or at http://ds.oregonstate.edu. DAS notifies students and faculty members of approved academic accommodations and coordinates implementation of those accommodations. While not required, students and faculty members are encouraged to discuss details of the implementation of individual accommodations.

Academic Calendar

All students are subject to the registration and refund deadlines as stated in the Academic Calendar: https://registrar.oregonstate.edu/osu-academic-calendar.

Student Bill of Rights

OSU has twelve established student rights. They include due process in all university disciplinary processes, an equal opportunity to learn, and grading in accordance with the course syllabus: https://asosu.oregonstate.edu/advocacy/rights.
Ecampus Reach Out for Success

University students encounter setbacks from time to time. If you encounter difficulties and need assistance, it’s important to reach out. Consider discussing the situation with an instructor or academic advisor. Learn about resources that assist with wellness and academic success.

Ecampus students are always encouraged to discuss issues that impact your academic success with the Ecampus Success Team. Email ecampus.success@oregonstate.edu to identify strategies and resources that can support you in your educational goals.

- **For mental health:**
  Learn about counseling and psychological resources for Ecampus students. If you are in immediate crisis, please contact the Crisis Text Line by texting OREGON to 741-741 or call the National Suicide Prevention Lifeline at 1-800-273-TALK (8255).

- **For financial hardship:**
  Any student whose academic performance is impacted due to financial stress or the inability to afford groceries, housing, and other necessities for any reason is urged to contact the Director of Care for support (541-737-8748).