CIS 3207: Introduction to Systems Programming & Operating Systems
Fall, 2024

Professor: Dr. Gene Kwatny
Office: SERC 324
Phone: 215-204-1679
EMAIL: gkwatny@temple.edu

Office Hours: Great opportunity for 1 on 1 discussion
Tuesday, Thursday 2:30 PM – 3:30 PM
(other times available by request)
Check Canvas for up-to-date office hour schedule [appointments are encouraged]

Course Assistants:

3207.001 Gavin Mai (gavin.mai@temple.edu)
Office Hours

3207.002 Katerina Orlovskiy (Katerina.orlovskiy@temple.edu)
Office Hours

3207.004 Magdalena Rogalska (m.rogalska@temple.edu)
Office Hours

Course Meeting Times: 3207 Sections 1, 2 and 4:

LECTURE: Tuesday & Thursday	12:30 PM - 1:50 PM	Beury 160	[Prof. Kwatny]
Laboratory Section 1: Wednesday	9:00 AM - 10:50 AM	BioSci 137	[Gavin Mai]
Laboratory Section 2: Wednesday	9:00 AM – 10:50 AM	Tuttleman 9	[Katerina Orlovskiy]
Laboratory Section 4: Wednesday	1:00 PM - 2:50 PM	BioSci 137	Magdalena Rogalska]

Prerequisites:
CIS 1166 Math Concepts in Computing I or MATH 2196 Basic Concepts in Math; CIS 2107 Computer Systems and Low-level Programming; CIS 2168 Data Structures and Algorithms (minimum grade of C- in each of the required prerequisite courses). The course will require substantial programming assignments; students should have experience programming in C.

Required Course Materials:
The course requires the use of two textbooks

- "OPERATING SYSTEMS - Three Easy Pieces", Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, free online: http://pages.cs.wisc.edu/~remzi/OSTEP/
  The digital version of individual chapters of this book are available for free. Optionally, you may purchase a PDF containing all chapters for $10.00, a softcover for $21.00, or a hardcopy for $36.00.
- "Computer Systems - a Programmer's Perspective",3e, Bryant & O'Hallaron, 2016, Pearson
  This is the same textbook used in CIS 2107. You may have a copy or purchase a used copy.

Textbook Content

Reading and lecture material from “Operating Systems Three Easy Pieces”: Chapters 1 - 8, 12 - 16, 18 – 22, 25 – 28, 30 - 32, 35 – 37, 39, 40.
Reading and lecture material from Bryant & O'Hallaron ('Computer Systems') Chapters 1, 6.1 - 6.3, 8, 11, 12 [you may find Chapter 9 helpful in understanding virtual memory]

Approximate Reading Schedule:

Lecture materials and supplemental documents and texts are available through Canvas.

Additional material will be drawn from selected research publications. You may also find the following optional reference texts useful to you in tackling the programming assignments for this course:

- The Linux Programming Interface: A Linux and UNIX System Programming Handbook by Michael Kerrisk
- UNIX Network Programming, Vol. 1: Networking APIs: Sockets and XTI by W. Richard Stevens
- Advanced Programming in the UNIX Environment, 3rd edition, by W. Richard Stevens and Stephen Rago
- “Operating Systems Internals” by Stallings,
- “Interprocess Communications in Linux” by Gray
- [Some of these texts are available in Canvas in ‘Additional Text Book References’]

COURSE TOPICS

- Operating system principles and computer architecture
- Processes and threads
- Concurrency
- CPU Scheduling and dispatching
- Memory management and virtual memory
- Device management
- File systems

LEARNING OBJECTIVES

- Operating systems general knowledge
  - Students will be able to explain the objectives and functions of modern operating systems.
  - Students will be able to analyze and express the tradeoffs inherent in operating system design.
  - Students will be able to explain the benefits of building abstract layers in hierarchical fashion.
  - Students will be able to describe the value of and demonstrate the use of APIs and middleware.
  - Students will be able to contrast kernel and user mode in an operating system
  - Students will be able to analyze interrupt processing and demonstrate how it functions.
- Concurrency
  - Students will be able to express the need for concurrency
  - Students will be able to describe and demonstrate how concurrency can be achieved within the framework of an operating system

        - Students will be able to demonstrate knowledge of the different states that a task may pass through and the data structures needed to support the management of many tasks
    - Students will be able to demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks
    - Students will be able to apply techniques for achieving synchronization in an operating system
        - Students will be able to analyze and express the reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system.
- Task Scheduling and Dispatching
    - Students will be able to apply and analyze several algorithms for preemptive and non-preemptive scheduling of tasks in operating system
    - Students will be able to discuss the similarities and difference between processes and threads and apply them in the implementation of systems
- Memory management
    - Students will be able to explain the rationale for hierarchical memory
    - Students will be able to understand, analyze, and demonstrate principles of virtual memory, particularly for caching and paging
- File systems
    - Students will be able to analyze design choices and tradeoffs in file systems
    - Students will be able to analyze fault tolerant approaches for the design of file systems

GRADING POLICY:
Course grade will be determined by :

Final Exam (30%) [the final exam will be comprehensive, covering all course material]
Midterm Exam (25%) [covers all material up to the date of the midterm exam]
Quizzes (20%) [There will be an in-class quiz once per week. Unless announced otherwise, quizzes are to be completed individually, in person; you may not work together on a quiz. The grade for a missed quiz is 0. The lowest 3 quiz grades will be dropped. No Makeup Quizzes]
Laboratory Projects (25%) [We will have 4 to 5 multi-week projects with weekly deliverables]

NOTE ABOUT GRADING IN CANVAS: I use Canvas to list graded items. I do not use Canvas to compute grades. Each exam will be scored and mapped to a letter grade. Projects will be graded individually and "the project group" will be given a letter grade. Quizzes will be graded individually, and the group of quizzes will be given a letter grade.
For the course grade, I apply the above percentages to each category for a final grade at the end of the semester. Ignore the weightings and assessments done by Canvas.

DATES of IMPORTANCE:

- Tuesday, August 26: First Class
- Wednesday, August 27: First lab
- Monday, September 9: Last day to add or drop a Full Term 16-week course (tuition refund)
- Friday, October 11: Fall Wellness Day
- Monday, November 25 – Wednesday November 27:Fall break (no classes)
- Thursday, November 28 – Sunday December 1: Thanksgiving Holiday (no classes)

- Wednesday December 4: Last Lab Session
- Thursday, December 5: Last Lecture Session
- Monday, December 9: Full Term 16-week Courses end
- Monday, December 9: Last Day to Withdraw
- Study Day: Tuesday, December 10
- Final Exam: 3207 Sections 1, 2 and 4 - Thursday, December 12, 10:30 AM – 12:30 PM, Beury 160

  Students who miss the final exam and do not make alternative arrangements with me before the exam, will receive a grade of F.

PROGRAMMING ASSIGNMENT POLICIES

All programs must compile; programs that do not compile earn a grade of zero. Coding style and documentation are important and will be reflected in your programming assignment grade. You must provide useful comments in your code, eliminate unnecessary commented code, partition your code into reasonably sized functions/methods, and use appropriate variable/function naming conventions. We recommend that you use Google's C and C++ coding style guide.

You are **not allowed** to look at other solutions for the programming assignments (or previous iterations of the homework), including those of other students or complete/partial solutions that may be available online. We will use a plagiarism detection tool, such as MOSS, on programming assignments.

We will be using GitHub to manage project code and materials. You will be required to learn to use Git and Github, perform code commits and use branching and source control as required. The GitHub repository you will create and maintain for this course is to be 'private', but shared with your TA and the instructor (This will be done automatically through your assignments). Do NOT use personal or public GitHub accounts. We will review progress in projects by following your GitHub commits. You must make regular commits to GitHub as you progress in project development.

Lab Assignments are generally multi-week and you will be given deliverables required each week as the project proceeds. Project code and results are to be submitted to Canvas. Projects are graded based on the last CANVAS submission prior to the project deadline.

The CIS laboratory computer systems in SERC laboratories 204, 206, BioSci 137, and Tuttleman 9 are available for use this semester.

We will be using the C language for program development to complete the projects. You will need a C compiler, linker and editing environment for this work. Your lab instructor will give you help in choosing software for local (your own computer) use. CIS-linux2.temple.edu is a CIS server that you can use for editing, compiling and demos. There are a number of cloud-based development environments that you can also use and one we can recommend for programming assignments:

CLion

COLLABORATION POLICY

We encourage you to discuss the problem sets and programming assignments with your fellow classmates. We welcome discussions of possible interpretations of questions, solution

approaches, and points of confusion. You are also welcome to use existing public libraries in your programming assignments (such as public classes for queues, trees, etc.) You may also look at operating systems code for public domain software such as Linux. Such activities qualify under approved collaboration practices and you are welcome to take advantage of them.

Unless explicitly stated otherwise on the assignment, you must write all code that you submit for programming assignments and all work on quizzes, homework sets, and exams must be your own. You are NOT allowed to LOOK AT other solutions, including those of other students or partial solutions for similar projects that may be available online through github or other platforms. For example, you may not look at the work done by a student in past years' courses, and you may not look at similar course projects at other universities. If you are unsure about whether a particular source of external information is permitted, contact the instructor before looking at it.

Some in-lab parts of laboratory projects may require/allow you to work in teams. In those cases, team members should contribute equally and will be graded individually. The write-ups and out-of-class portions of labs must be completed independently.

Intellectual dishonesty can end your academic career, and it is your responsibility to stay on the right side of the line. It is generally OK to verbally discuss the concepts needed to do project assignments. These discussions should focus on overall approach and understanding, not the detailed answer to the specific problem. The following guidelines can help you to keep on the right side of the line:

- First, other than to the TA and instructor, it is never OK to look at the written work of another person or show another person your written work until after all grading on an assignment is completed. This includes looking at paper print-outs, sketching solutions on a white board or napkin, or looking at a screen to help debugging. It should go without saying that copying other people's code or solution sets is strictly prohibited.

- Second, everyone in the class is expected to take appropriate measures for protecting one's work. For example, you should protect your files and printouts from unauthorized access.

Students who violate university rules on academic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Because such dishonesty harms the individual, all students, and the integrity of the University, University policies on scholastic dishonesty will be strictly enforced.

If you are not sure about the use of a resource or the appropriateness of a collaboration, ask.

LATE PROJECT ASSIGNMENT SUBMISSION POLICY

This is a fast-paced course with frequent programming assignments. We have many projects to complete this semester. If you delay in completing a project, you will end up behind in each subsequent project. The course and assignments are designed so that you have sufficient time to complete the associated coursework. In the event that you cannot submit an assignment by the submission deadline, late assignments will be accepted, with penalty, as follows:

- 1 second up to 1 day late, maximum grade of 90% of project value
- 1 to 2 days late, maximum grade of 80% of project value

- 2 to 3 days late, maximum grade of 70% of project value
- 3 to 4 days late, maximum grade of 60% of project value
- 4 to 5 days late, maximum grade of 50% of project value
- after the 5th day, the submission will not be accepted and a grade of 0 will be assigned.*
  You are given 3 FREE LATE Days for the semester.

*Exceptions to this late policy can be made for illness and other life situations requiring flexibility. If you are experiencing health or personal circumstances that impact your ability to meet the deadlines, please let me know as soon as possible so that we can help you to stay on track and be successful in the course.

ATTENDANCE

Attendance for all class sessions is strongly encouraged. Attending classes is critical for you to be successful in this course and to participate in all of the group interactions. You are expected to be an active participant in the course and take responsibility for putting yourself in the best situation to advance your learning throughout the entire semester. If you miss class, you may miss important announcements, clarification, or discussion about assignments that may not be repeated on the course website.

Attendance for lectures will be taken through Canvas (Qwickly)

Important points:

- If you do miss class, you are responsible for catching up on material that you missed and meeting course deadlines. Read the assigned text, ask a classmate for notes, and review lecture slides, which will be posted within 48 hours of each lecture. We are happy to answer questions about the lecture after you have reviewed the material that you missed.
- Religious holy days sometimes conflict with class and examination schedules. If you notify the instructional staff 14 days in advance of a planned absence due to observance of a religious holiday, you will have an opportunity to make up missed work (including exams) within a reasonable timeframe, either before or after the absence.

COURSE COMMUNICATION POLICY

ASK HOMEWORK AND CONTENT UNDERSTANDING QUESTIONS ON DISCORD: When you have a question related to the course content and assignments, you should post your question on Discord, where the instructional staff that is assigned to monitor questions can answer. We do our best to respond to questions within 48 hours. It is your responsibility to plan accordingly to study/do your assignments in time to ask questions. Questions about course content and assignments that are sent only to the instructor by email are not guaranteed to be unanswered. A link to the Discord server is shared on the Canvas course home page.

ASK QUESTIONS ABOUT GRADES AND REQUEST APPOINTMENTS BY EMAIL: Your email should be sent to both the instructor and the TA. Our TA is very knowledgeable and is skilled in answering student questions; you should attend the TA's office hours or schedule an individual meeting with the TA before meeting with the instructor. Email Dr. Kwatny directly for an appointment.

CHECK YOUR EMAIL FOR COURSE ANNOUNCEMENTS: Canvas notifications are delivered via email. As such, students are responsible for checking e-mail on a frequent and regular basis for class announcements.

CODE OF CONDUCT

It is expected that all students will work together to make the CIS 3207 classroom and extended learning spaces a welcoming learning environment for every person. This can be summarized as being respectful to others (even if they don't use your favorite programming language or IDE!) and behaving professionally. Keep in mind: listen first, don't talk over others, and don't be dismissive of others' opinions and potential for their contributions.

Harassment based on gender, sexual orientation, disability, physical appearance, body size, race, nationality, religion, political affiliation, or choice of programming language or tool will not be tolerated. Please review the university policy and procedures for addressing discrimination and harassment.

STUDENT RESPONSIBILITIES

Students are responsible for reading all assigned text materials, handouts, and referenced sources. Students are responsible for participating in classroom discussions and discussions carried out though Canvas and Discord.

Much of the source code work in the course will require you to have familiarity with (or become familiar with) the C language and development environments for compiling and building C or C++ programs primarily in Linux. Reference material for the C Language is available via the course Canvas site.

This course requires the use of Canvas, including access to materials and for assignment submission. Some videos posted via Canvas will require the use of speakers on your computer. We may utilize web-conferencing tools to deliver synchronous material (in particular, Zoom). In order to participate in synchronous sessions, you should have a computer, a webcam, headphones, and microphone. **Polling will be used for some in class questions and you will need web access via phone or computer to participate**.

Students are responsible for taking all quizzes and exams in the course. All work turned in for grading or review by the instructors of the course must be the students own work. The objectives of the course can only be met by you doing all of the work and presenting only your work for grading. Presenting work that is not your own will result in disciplinary action and no credit.

POLICY ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE AND ASSOCIATED TOOLS

The use of generative AI tools (such as ChatGPT, DALL-E, etc.) is not permitted in this class unless specifically assigned. Therefore, any unassigned use of AI tools for work in this class, at all stages of the work process, including preliminary ones, may be considered a violation of Temple University's Academic Honesty policy and Student Conduct Code, since the work is not your own. The use of unauthorized AI tools will result in disciplinary action and no credit.

STUDENT AND FACULTY ACADEMIC RIGHTS AND RESPONSIBILITIES
Freedom to teach and freedom to learn are inseparable facets of academic freedom. The University has a policy on Student and Faculty and Academic Rights and Responsibilities (Policy #03.70.02) which can be accessed through the following http://policies.temple.edu/PDF/99.pdf.

COLLABORATION POLICY

We encourage you to discuss the problem sets and programming assignments with your fellow classmates. We welcome discussions of possible interpretations of questions, solution approaches, and points of confusion. You are also welcome to use existing public libraries in your programming assignments (such as public classes for queues, trees, etc.) You may also look at operating systems code for public domain software such as Linux. Such activities qualify under approved collaboration practices and you are welcome to take advantage of them.

Unless explicitly stated otherwise on the assignment, you must write all code that you submit for programming assignments and all work on quizzes, homework sets, and exams must be your own. You are NOT allowed to LOOK AT other solutions, including those of other students or partial solutions for similar projects that may be available online through github or other platforms. For example, you may not look at the work done by a student in past years' courses, and you may not look at similar course projects at other universities. If you are unsure about whether a particular source of external information is permitted, contact the instructor before looking at it.

Some in-lab parts of laboratory projects may require/allow you to work in teams. In those cases, team members should contribute equally and will be graded individually. The write-ups and out-of-class portions of labs must be completed independently.

Intellectual dishonesty can end your academic career, and it is your responsibility to stay on the right side of the line. It is generally OK to verbally discuss the concepts needed to do project assignments. These discussions should focus on overall approach and understanding, not the detailed answer to the specific problem. The following guidelines can help you to keep on the right side of the line:

- First, other than to the TA and instructor, it is never OK to look at the written work of another person or show another person your written work until after all grading on an assignment is completed. This includes looking at paper print-outs, sketching solutions on a white board or napkin, or looking at a screen to help debugging. It should go without saying that copying other people's code or solution sets is strictly prohibited.
- Second, everyone in the class is expected to take appropriate measures for protecting one's work. For example, you should protect your files and printouts from unauthorized access.

  Students who violate university rules on academic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the

University. Because such dishonesty harms the individual, all students, and the integrity of the University, University policies on scholastic dishonesty will be strictly enforced.

If you are not sure about the use of a resource or the appropriateness of a collaboration, ask.

Academic Honesty

According to the University Student Code of Conduct, students must not commit, attempt to commit, aid, encourage, facilitate, or solicit the commission of academic dishonesty and impropriety including plagiarism, academic cheating, and selling lecture notes or other information provided by an instructor without the instructor's authorization. Violations may result in failing the assignment and/or failing the course, and/or other sanctions as enumerated in the University Code of Conduct (Links to an external site.).

Students are responsible for taking all quizzes and exams in the course. All work turned in for grading or review by the instructors of the course **must be the students own work**. The objectives of the course can only be met by you doing all of the work and presenting only your work for grading. **Presenting work that is not your own will result in disciplinary action and no credit.**

Recording and Distribution of Recordings of Class Sessions

Some sessions in the course may be recorded and access to those recordings will provided in the Canvas course.

Any recordings created or permitted in this class can only be used for the student's personal educational use. Students are not permitted to copy, publish, or redistribute audio or video recordings of any portion of the class session to individuals who are not students in the course or academic program without the express permission of Professor Kwatny and of any students who are recorded. Distribution without permission may be a violation of educational privacy law, known as FERPA as well as certain copyright laws. Any recordings made by the instructor or university of this course are the property of Temple University

ACCOMMODATIONS FOR STUDENTS WITH DISABILITIES

Any student who has a need for accommodation based on the impact of a documented disability, including special accommodations for access to technology resources and electronic instructional materials required for the course, should contact me privately to discuss the specific situation by the end of the second week of classes or as soon as practical. If you have not done so already, please contact Disability Resources and Services (DRS) at 215-204-1280,4th Floor Gittis Student Center South to learn more about the resources available to you. I will work with you and DRS to coordinate reasonable accommodations for all students with documented disabilities.

INCOMPLETE GRADE POLICY

Incomplete grades will be granted only in rare circumstances and require the approval of the instructor. A student will be eligible for a grade of "Incomplete" only if he/she: 1) has completed at least 51% of the work at a passing level, 2) is unable to complete the work for a serious reason

beyond his or her control, and 3) files a signed agreement with the instructor outlining the work to be completed and the time frame in which that work will be completed. The student is responsible for initiating this process and all incomplete forms must be sent to the Associate Dean for Academic Affairs prior to the start of study days in that semester.

Please refer to the following for further details: Temple University's Incomplete Policy (Links to an external site.) (Policy #02.10.13).

Withdraw from the Course

If a student wishes to withdraw from a course, it is the student's responsibility to meet the deadline for the last day to withdraw from the current semester

See Temple University's Academic Calendar (Links to an external site.) for withdrawing deadlines and consult the University policy on withdrawals (Links to an external site.) (Policy # 02.10.14).

Statement on Academic Rights & Responsibilities

Freedom to teach and freedom to learn are inseparable facets of academic freedom. The University has a policy on Student and Faculty Academic Rights and Responsibilities (Links to an external site.) (Policy #03.70.02).

Last Modified: 8/1/2024