

CS 4094 Syllabus “Systems”

Overview

CS 4094 “Systems & Networking Capstone” provides an in-depth introduction to the principles and practices of operating systems. Particular emphasis is given to the topics of multiprogramming, process and thread management, memory management, including virtual memory, concurrency, including synchronization and deadlock, resource allocation and management, including scheduling, and storage management and file systems. Additional topics include inter-process communication, networking, and device management.

Rather than learning what an OS looks like from the outside and how to use its facilities, this course will show you what an operating system looks like from the inside. Whereas our core CS 3214 course looks at systems from the perspective of an application programmer using an OS, this course will look at operating systems from a system’s designer point of view.

The topics will be accompanied by a series of programming projects that will give you hands-on experience in building significant parts of a real operating system. All projects will be done in groups.

In addition to a series of structured projects, each group will engage in one open-ended project. Each group will create a poster for this project. These posters will be displayed at the VTURCS capstone symposium in April.

Staff Information and Meeting Times

Instructor: Dr Godmar Back
Gilbert Place 4208

Office hours:
by appointment

Class website: <http://courses.cs.vt.edu/cs4284/spring2026>

GTA: There is no GTA for this course.

Email: Send email to gback@cs.vt.edu or godmar@gmail.com

Class Meeting Times:
DER 3081 5:00pm – 6:15pm T R

Because subjects taught in class closely correspond to the concurrently run projects, we ask that you attend class regularly.

In addition, for each project, there will be a mandatory group meeting about 1.5 weeks before the project deadline. These group meetings will substitute for class time on the day they are held, but may extend longer in the evening.

Prerequisites

With the reform of the capstone system, this course no longer has CS3214 as a prerequisite. *I do recommend, however, that students either take CS3214 concurrently, or have already taken and appreciated it, or otherwise have a strong interest in operating systems. If this doesn't apply to you, consider switching to another capstone section (if possible).*

Objectives

Upon completion of the course, students should be able to

1. Understand the basic structure and organization of a multiprogrammed computer system, including the distinction between user and kernel mode, the use of interrupts and context switches, runtime organization, application-binary interfaces and system calls, program linking and loading.
2. Understand the principles underlying concurrency and know how to use proper synchronization and deadlock avoidance techniques.
3. Understand the principles behind memory management, including user-level memory management, virtual memory management and paging.
4. Understand the principles behind CPU scheduling, including round-robin, priority-based, multilevel feedback queues, and weighted fair queuing based scheduling algorithms.
5. Understand how an OS provides protection to its applications and how it manages and virtualizes resources.
6. Understand how file and storage systems are constructed and what factors influence their performance.

At a higher level, I would like for you to take away an appreciation for the complexity of operating systems, and view this class as an example of how to learn managing complexity.

Capstone Objectives

In addition, since this course is a capstone course, we include the capstone requirements and objectives. Students should be able to

- Identify and formulate a computing problem

- Design, implement, and evaluate a solution to an open-ended computing problem
- Apply design and development principles to produce a software solution
- Critically evaluate alternative design choices
- Demonstrate effective teamwork skills
- Use written communication skills effectively
- Use oral presentation skills effectively

Format

The course work consists of a mix of lectures, structured programming projects and one open-ended project. The open-ended project's results will be presented in a final presentation. In addition, each group will create a poster on their project.

There will be **no other exams or other homeworks** in this class.

Projects: There will be 4 structured and one open-ended project, where you will be working in groups of 4 or 5 (depending on enrollment). Projects will be submitted electronically via git tagging on git.cs.vt.edu. Instructions will be posted on the class website.

Structured Projects

We will be using Pintos as our project infrastructure, an educational operating system originally developed by Ben Pfaff at Stanford University. Pintos is used at a number of universities in comparable undergraduate OS courses. We provide a baseline version of Pintos, and you will add various features to Pintos through the course of the semester. Pintos runs on the x86, and could boot on a PC, but we will use a virtual machine simulator to run it on the Linux remote login cluster. We will grade your submissions on those machines.

Only 50% of your grade in those projects is determined by passing the test cases we provide (all of which are public). 25% comes from a review of your code by me to evaluate its design. The remaining 25% of your score comes from an design document you are required to submit with your solution. The design documents this semester will have to be written by **each student individually**. They consist of a free-form description of the overall submission's design (not just the part you worked on). Groups can – and should – discuss the designs, but each group member needs to be able to formulate it in their own words.

The original Pintos series was designed for a 10-week quarter and for students who had not taken a systems course before. We are using it for a 15-week semester course for students near graduation. We will start with

projects right away, and use the remaining 6-7 weeks to complete the implementation of an open-ended project.

Open-ended Project

Each group will work on an open-ended project that extends the Pintos infrastructure in an interesting way. Examples of possible projects include:

- Implementing multi-threading for user processes
- Implementing inter-process communication facilities such as pipe
- Implementing security such as principals and permissions
- Implementing shared memory
- Implementing more advanced virtual memory, including `sbrk()` and anonymous memory
- Implementing a more advanced file system facility
- Add a device driver for a new device
- Adding 64-bit support
- Adding basic support for TCP/IP over Ethernet networking
- Development of software checking tools
- Experimenting with support for other languages (e.g. Rust)
- Implementing exceptionless system calls (e.g. `io_uring`)
- Ports to different architectures (e.g. RISC-V)

Each group will create a poster presenting their project at the Capstone Symposium in April.

Late Policy

Accommodations beyond automatic late days (aka “extra late days”):

The automatic late day system is designed so that we do not have to respond to requests for additional late days. However, we will provide additional late days in 2 situations:

DoS/University accommodations: If you have family or other emergencies that prevent you from submitting assignments on time, please contact the Dean of Students Office (<https://www.dos.vt.edu>). They will make a determination as to what accommodations should be given, and inform the instructors of the classes in which you are enrolled of their decision. Our policy is to provide you with as many additional late days as the note from DoS advises. For reasons of consistency and fairness, we will not make any determinations about emergency accommodations; we will defer all such decisions to the DoS. If you have learning or other disabilities, please also see the section Students With Disabilities below.

Sickness policy: If you cannot complete an assignment due to illness, **you must tell the teaching staff how many days you need to catch up**

on any work you were not able to do because of sickness. The deadline for the assignment in question will then be moved by this many days without counting against the late days described above. For group projects, sick days will be granted to the group as a whole for the project in question. **No Doctor's note is required or expected, but the honor code and the university policy on class attendance apply.** If the number of requested sick days is on an assignment is more than 5, or the overall number in the semester is more than 10, we ask that you coordinate with the Dean of Students as well to ensure your ability to complete the semester.

Incomplete Policy/Academic Relief

In most cases, students who missed too much work will be asked to withdraw or drop and then retake the course. In exceptional circumstances, the College of Engineering may grant academic relief. As per updated guidance from the College of Engineering, such academic relief accommodation may include an Incomplete grade provided that all of the following are true:

- The student has been attending class on a regular basis.
- The student has completed 75% or more of the required coursework.
- With exception for the missing work, the student is passing the course.
- The student has a viable plan and timeline for completing the remaining work.

Grading

I estimate that the contributions of the different portions to your final grade will be as listed below, but I reserve the right to adjust these weights as necessary:

40%	Structured Projects
35%	Open-ended Project
25%	Final Poster & Presentation

Since this class is a capstone/elective, I will not predict where the median grade for the class will be (it is generally high since this course is chosen mostly by motivated students). Nevertheless, to achieve an A in this class, you should expect to produce working and well-documented solutions to both the structured projects and the open-ended project.

Collaboration Policy and Honor Code

The University’s officially required honor code statement applies to this course, which is reproduced below.

The Undergraduate Honor Code pledge that each member of the university community agrees to abide by states:

“As a Hokie, I will conduct myself with honor and integrity at all times. I will not lie, cheat, or steal, nor will I accept the actions of those who do.”

Students enrolled in this course are responsible for abiding by the Honor Code. A student who has doubts about how the Honor Code applies to any assignment is responsible for obtaining specific guidance from the course instructor before submitting the assignment for evaluation. Ignorance of the rules does not exclude any member of the University community from the requirements and expectations of the Honor Code. For additional information about the Honor Code, please visit: www.honorsystem.vt.edu.

The tenets of the honor code will be strictly enforced in this course, and all assignments shall be subject to the stipulations of the Undergraduate Honor Code. For more information on the Honor Code, please refer to the Undergraduate Honor System Constitution, located online at <http://www.honorsystem.vt.edu/>.

The following policies regarding collaboration apply in this class.

1. All submitted work is expected to be the original work of the individual student or group unless otherwise directed by the instructor. See also use of generative AI below.
2. Projects are to be the work of the individual student or team as specified. All source material used in project code and reports must be properly cited.
3. For the projects you will team up in groups of typically 4 students. You may switch teams or form new teams, but only between projects. You may work with at most one group on a given project. Students must contribute equally to the project within a team. It is not acceptable for students to either not contribute to the project or not to let the other group members contribute equally to the project. Please bring any problems in this regard to the instructor’s attention early on.
4. You are required to read-protect your work on shared file space so students outside of your team will not have access. This includes your git repository on git.cs.vt.edu.
5. Borrowing code or hiring someone to perform the work for you is an egregious violation of the honor code.

Use of generative AI tools such as Codex, GPT-5, Claude, etc.:

There **are no restrictions** on the use of generative AI as far as the production of computer code is concerned. However, you will need to retain mastery of your code at all times.

The use of generative AI for impersonation is forbidden, however. Do not post AI-generated comments on the class forum. Do not “improve” your writing with AI to make it “look more professional.” You achieve the opposite.

Your design documents must not have been AI-generated. (If they are, it is often easy to spot because most current AIs actually do not have good knowledge of Virginia Tech’s pintos.)

If you use AI to research and learn, and if you want to include what you have learned this way, cite and acknowledge it like you would cite a source.

Students with Disabilities

Students are encouraged to address any special needs or special accommodations with me during the first two weeks of the semester, or as soon as you become aware of your needs.

Those seeking accommodations based on disabilities should obtain a Faculty Letter from the Services for Students with Disabilities office (540-231-0858) located in Lavery Hall, Suite 310) <http://www.ssd.vt.edu/>.

If you need accommodations because of a disability (learning disability, attention deficit disorder, psychological, or physical), please contact the instructor as soon as possible.

Mandatory Reporting Statement: University Policies 1025/1026

University Policies 1025 and 1026 mandate employees with instructional responsibilities to report all suspected instances of discrimination, harassment, sexual harassment and/or sex/gender-based violence to the [Office for Equity and Accessibility](#) (OEA). The University also has identified certain offices as confidential resources. The University’s confidential resources include: [the University Ombuds Office](#), [The Graduate School Office of the Ombudsperson](#), [Schiffert Health Center](#), [Cook Counseling Center](#), and the [Virginia Tech Women’s Center](#). Additionally, the University offers supportive measures to survivors of sex and gender based harassment and violence, regardless of whether the survivors file a Formal Complaint. More information is available from the [OEA](#).