CS 3214: Computer Systems

Spring 2026 Syllabus

Virginia Tech — Department of Computer Science

1   Course Overview

CS 3214 provides an introduction to computer systems as they are relevant to application program-
mers today, with an emphasis on operating systems principles. Topics covered include:

- Operating systems design and architectures
- Processes, threads, synchronization techniques, deadlock
- CPU scheduling
- System call interfaces, system level I/O and file management
- Shell programming
- Separate compilation, loading and linking
- Inter-process communication (IPC)
- Virtual and physical memory management and garbage collection
- Network protocols and programming
- Virtualization and cloud computing

The topics will be accompanied by a series of exercises and programming projects that will give
you hands-on experience in interacting with systems at different levels. The projects will be done
in groups of 2, which gives you the added benefit of learning how to work in a small team.

2   AI Considerations

Recent advances in generative AI require us to adapt the way in which we teach and assess students
in all disciplines of computer science. Current coding agents such as Jules, Codex, or Claude
Code can produce fully working artifacts for structured course projects with little or no human
intervention, solely on the basis of detailed specifications like the ones we typically prepare for such
projects. As a result, we will no longer use code artifacts created in unproctored environments to
assess the skills or level of mastery of students.

That said, we continue to believe that projects that require you to apply the systems-level concepts
you're exposed to in the context of specific requirements, environments, and system designs, and
that result in deliverables are very likely to be beneficial, if not necessary, to achieve the level of
skills and understanding of systems we (and future employers) expect CS graduates to possess.

For this reason, our approach will be fourfold:

1. We will continue to include projects that are designed to be implemented by humans with
   minimal or no AI support.
2. We will allow the use of AI assistants for these projects.

3. We will grade these projects based on functionality and correctness.

4. We will increase the number of unassisted, proctored assessments (exams and quizzes). Part of these assessments will assume, and be specifically designed to test understanding that we believe can only be gained by having worked on the projects in the manner we recommend: namely, by using AI as an assistant, not by becoming an assistant to AI.

3 In-Person Format

During the Spring 2026 semester, CS 3214 will be taught as an **in-person class** augmented with various modes of online communication. This consists of in-class meetings/discussions/lectures and demos, book reading assignments, slide sets, use of Discourse, Discord, and Zoom for in-person and online office hours.

4 Staff Information

4.1 Instructors

Instructor	Section Details
Dr. Huaicheng Li	CRN 13470, MW 5:30–6:45pm, TORG 2150
Mr. Milo Craun	CRN 13471, MW 2:30–3:45pm, SURGE 104C

Attendance at class meetings is strongly encouraged. Class meetings are not recorded.

4.2 Contact Information

Resource	Details
Office hours	Available on class website
Class website	https://courses.cs.vt.edu/cs3214/spring2026
Email	cs3214-staff@cs.vt.edu (or use Discourse forum)
Forum	https://cs3214.cs.vt.edu (Discourse, authenticates via @vt.edu)
Discord	https://discord.gg/S9V3DzGHQK (for TA office hours queuing)

4.3 Graduate Teaching Assistants (GTAs)

Name	Email
Hao Li	lh25@vt.edu
Inho Song	inhoinno@vt.edu
Noah Provenzano	noahpro@vt.edu
Bernard Moussad	bernardm@vt.edu
Yunqi Shen	yshen424@vt.edu
Sina Heidari	sinaheidari@vt.edu
Yoon Lee	yoonl18@vt.edu
Egor Lukiyanov	egor@vt.edu
Donia Mohammed Ghazy Amer	doniaghazy@vt.edu
Zhengjie Ji	zhengjie@vt.edu

4.4 Undergraduate Teaching Assistants (UTAs)

Name	Email
Christian Calvo	kuzoto@vt.edu
Thuan Van Tu	ttv27042004@vt.edu
Prashant Joshi	joship5255@vt.edu
Christian Franklin	christianf23@vt.edu

5 Prerequisites

The formal prerequisites for this class consist of a **C or better** in:

- CS 2506 Introduction to Computer Organization II
- CS 2114 Software Design and Data Structures

Prerequisite checking is done per the departmental procedures.

Note: CS 2505 is indirectly a prerequisite for CS 3214. If you received transfer credit for CS 2506, but have not taken CS 2505, you are strongly advised to complete CS 2505 before attempting CS 3214.

6 Learning Objectives

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

1. Explain the major components of an operating system and their respective functionality and interaction, including runtime libraries and systems programs
2. Use an operating system's command line interface to run shell scripts and perform productivity tasks
3. Describe the theories underlying CPU scheduling and their impact on applications
4. Describe proper synchronization techniques, deadlock detection and avoidance
5. Describe the principles underlying virtual memory and their impact on applications
6. Design, implement, test, and debug:
    a. Programs that interact directly with an operating system using its system call application programming interface (API)
    b. Multi-process applications that use inter-process communication
    c. Multi-threaded applications
    d. Custom user-level memory allocators
    e. Simple network applications
7. Describe the principles underlying virtualization

7 Textbook

Required:

Computer Systems: A Programmer's Perspective, Third Edition (CS:APP3e)
Randal E. Bryant and David R. O'Hallaron
Prentice Hall, 2016, ISBN 978-0134092669

8   Tentative Schedule

This schedule is tentative. The course website and syllabus are under active development, and we aim to finalize them by end of the first week.

Week	Date	Due Date	Test Date	Topic Outline
1	1/21			Intro, Getting Started
2	1/26	1/31 ex0		Process Management, Shell, Job Control, Unix Signals, Pipes, File Descriptors, Std I/O
3	2/2	2/6 syllabus quiz		
4	2/9	2/13 ex1		
5	2/16			Compilation, Loading and Linking
6	2/23	2/27 p1, part 1/3	Test 1 (2/25)	
7	3/2	3/6 ex2		Multithreading, Concurrency, Synchronization
Spring Break: March 7–15				
8	3/16	3/20 ex3		
9	3/23			
10	3/30	4/3 p2, part 2/3		Memory Management: automatic & explicit; Virtual Memory
11	4/6		Test 2 (4/8)	
12	4/13	4/18 p3		
13	4/20	4/24 ex4		Networking and Socket Programming
14	4/27			Virtualization, Containers, Cloud Computing
15	5/4	5/6 p4, ex5, part 3/3		
	5/7		Reading Day	
	5/8–13		**Final Exam**	(per HokieSpa)

8.1   Key Dates (VT Academic Calendar Spring 2026)

- **Classes Begin:** January 20, 2026 (Tuesday)
- **MLK Holiday:** January 19, 2026 (No classes)
- **Spring Break:** March 7–15, 2026
- **Last Day of Classes:** May 6, 2026
- **Reading Day:** May 7, 2026
- **Final Exams:** May 8–13, 2026

9   Course Work

9.1   Tests and Final Exam

There will be **2 in-class tests** and **1 final exam**. Each test and exam will cover material from the lectures and textbook. Tests and exams will include questions related to the programming projects and exercises. The final exam will be comprehensive and include material from the lectures, textbook, and programming projects.

- All tests will be **closed book and closed Internet/no AI**
- A sheet of prepared notes will be allowed for each test

- These sheets may be reused for the final exam, and you may bring one additional sheet
- The tests will be during a class period
- The final exam will take place at the scheduled time as per HokieSpa

9.2   Exercises

Exercises are done **individually**. They consist of smaller tasks that reinforce necessary skills. Exercises may run in parallel with projects, but generally have a shorter deadline.

9.3   Projects

There will be **4 medium-size projects**. Projects must be done in **groups of 2 students**. Teaming up with a partner is not optional. Teams may be formed across sections.

Each project is worth 5% of your grade.

Group Formation:

1. Self-select a partner and register via the "grouper app" on the course webpage
2. Choose to be assigned a partner

For each project, there will be a time period where the grouper app is unlocked (open for self-registration). Once this time period is over, the grouper app will be locked.

You may switch partners only between projects. You may work with at most one partner on a given project. If you do not unregister your partner when the grouper app is unlocked, we will assume that you continue to work with your assigned partner.

Equal Contribution: 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 member contribute equally. Inform staff right away if your partner is not responsive.

If there are questions within a group regarding contributions, we may ask each group member to report:

- How often they met with their partner
- Their specific contributions to the project
- Any difficulties they had with their partner
- What attempts they made to overcome those

We will use this information to decide whether both group members should get credit for the project, or whether to award extra late days where warranted.

10   Participation Requirement

To encourage attendance and engagement, we set aside a small portion of your grade for participation. This credit is earned through **class attendance**.

We will track attendance using a QR code system during lectures. At certain points during class, a QR code will be displayed that you can scan to record your attendance.

We will set aside **6 points** for participation, via roughly 6-12 attendance checks.

11   Late Policy

Deadlines are necessary for everyone to stay on schedule and to facilitate grading. We use built-in late days every student can individually budget.

Each student has:

- **4 late days** for projects
- **4 late days** for exercises (separate budget)

Late days are granted in whole integer multiples of days. For projects, late submissions count against both partners' budgets. Submissions received after all late days are exhausted will not be graded and receive zero credit.

11.1   Accommodations Beyond Automatic Late Days

- **DoS/University accommodations:** Contact Dean of Students Office (`https://www.dos.vt.edu`) for family or other emergencies. We will provide as many additional late days as the DoS advises.
- **Sickness policy:** Tell the teaching staff how many days you need to catch up on work due to medical reasons. The deadline will be moved by this many days without counting against late days. No doctor's note is required, but the honor code applies.

12   Incomplete Policy/Academic Relief

As per College of Engineering guidance, an Incomplete grade may be granted if:

- Student has been attending class on a regular basis
- Student has completed 75% or more of the required coursework
- Student is passing the course (with exception for missing work)
- Student has a viable plan and timeline for completing remaining work

13   Grading

Component	Weight
In-Class Tests (2 tests, 16% each)	32%
Final Exam	20%
Projects (5% each)	20%
Exercises	20%
Participation (Attendance)	6%
Syllabus Quiz	2%

13.1   Letter Grade Conversion

We guarantee your letter grade will not be worse than the standard scale:

- 90% = A-
- 80% = B-
- etc.

In practice, we may, and typically will, adjust the final results by setting cut-off points lower than these. The goal is to ensure that students with similar performance receive the same grade whereas students whose performance differs more significantly do not.

From past experience, the median final grade for this class typically lies between B and C. Students who consistently perform above the median can reasonably expect a B- or better.

14   Modes of Communication

Channel	Purpose
Website	Primary communication, announcements
Discourse Forum	Required subscription, all Announcements category posts
Email	cs3214-staff@cs.vt.edu for administrative issues
Discord/Zoom	Office hours, ensure audio/screen sharing works

Email Etiquette:

- Always contact us via cs3214-staff@cs.vt.edu
- Use VT accounts only (FERPA)
- No email attachments unless requested
- Never send project/exercise submissions by email

15   Collaboration Policy and Honor Code

The tenets of the honor code will be strictly enforced. The Undergraduate Honor Code pledge:

"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."

15.1   Academic Misconduct Includes

- **Cheating:** Unauthorized materials, information, notes, study aids
- **Plagiarism:** Copying language, structure, programming, code, ideas
- **Falsification:** Statement of untruth in academic work
- **Fabrication:** Making up data and results
- **Multiple Submission:** Submitting previous work without authorization
- **Complicity:** Helping another engage in academic misconduct

15.2   Sanctions

Normal sanction for any violation: F* (Failure due to Academic Honor Code Violation)

- F* documented on transcript with notation "FAILURE DUE TO ACADEMIC HONOR CODE VIOLATION"
- Education program required to remove "*" notation
- "F" remains permanently
- Second F* typically leads to expulsion

15.3   Specific Policies

- All submitted work must be original
- Projects: work of individual student or team only
- **No code posting** on Discourse/Discord (except single lines for errors)
- **Must use git.cs.vt.edu** (not github.com, code.vt.edu)
- Read-protect your work on shared file space
- After class ends, do not publicly share solutions
- Borrowing code or hiring someone to perform work for you is an egregious violation

15.4   Use of Generative AI (CoPilot, ChatGPT, Claude, Gemini, etc.)

There are no restrictions on the use of generative AI tools for generating code.

That said, we advise that you be careful in their use. No matter how sophisticated AI tools become, it is the value *you* provide, not them, that ultimately determines whether you will be attractive to future employers. Employers will assess job candidates in unassisted or minimally assisted scenarios.

The creation of discussion posts using generative AI is **forbidden without proper attribution/quotation** that clearly delineates your (human) contribution and that of AI. If you contribute AI-generated material, you must clearly attribute it and augment it with your own contribution to receive participation credit.

15.5   Policy for Students Retaking the Class

Students retaking the class can use previous code as a **reference only** but:

- Cannot share with new group members
- Cannot copy old code
- All submissions must be created from scratch

We will enforce this policy: you can expect an honor code case to be filed if we detect similarities between your submission and a submission made in previous semesters.

16   Students with Disabilities

Students seeking accommodations should:

1. Obtain a Faculty Letter from Services for Students with Disabilities (SSD)
   - Lavery Hall, Suite 310
   - 540-231-0858
   - http://www.ssd.vt.edu/
2. Share the letter with your instructor in a timely manner

We reserve a room for students whose accommodations include a reduced distraction environment for taking exams. We do not support the use of the SSD testing center.

17   Mandatory Reporting Statement

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).

Confidential Resources:

- University Ombuds Office
- The Graduate School Office of the Ombudsperson
- Schiffert Health Center
- Cook Counseling Center
- Virginia Tech Women's Center

The University offers supportive measures to survivors regardless of whether they file a Formal Complaint. More information is available from the OEA.