(Advanced) Linux Kernel Programming (Spring 2025) Course Syllabus

CS-5264/CS-4224/ECE-5414/ECE-4414 is primarily intended for senior students (motivated junior students) and graduate students who want to concentrate on systems. To benefit from the course, low-level programming skills (e.g., C or ASM programming) and preliminary knowledge on computer system and architecture are necessary. You will learn key subsystems of Linux Kernel and will be asked to develop kernel code to extend kernel subsystems. Upon completion of the course, you will be able to:

- Identify the various subsystems composing the Linux kernel and describe their functionality, architecture, as well as the main characteristics of their implementation
- Design, implement and modify Linux kernel code and modules for these subsystems
- Test, debug and evaluate the performance of systems software in kernel or user space, using debugging, monitoring and tracing tools

Prerequisite

- Both levels: Good knowledge of C programming and the Linux command line is assumed. Knowledge concerning computer architecture and operating system is recommended.
- 4224/4414: CS 3214 (Computer Systems), CS 3114 (Data Structures)

Class

- When: TR 3:30-4:45pm
- Where: WMS 120

Office Hours

Instructor office hours and TA office hours will be announced soon.

- Instructor: Huaicheng Li
- TA: TBD

Tentative Schedule & Topics

- Intro to Linux Kernel
- Syscall
- Kernel data structures and debugging
- Process management and scheduling (scheduler, ebpf, etc)
- Interrupt handling
- Kernel synchronization
- Timer management
- Memory management (address space, in-kernel memory allocation, page table and fault management, heterogeneous memory management)
- File system and block layer (VFS, page cache, NVMe storage)
- Networking stack (TCP, RDMA, TBD)

Text Book

- (highly recommended) Linux Kernel Development, 3rd Edition, by Robert Love
- (optional) Linux Kernel Programming: A Comprehensive and Practical Guide to Kernel Internals, Writing modules, and Kernel Synchronization, Second Edition, by Kaiwan N. Billimoria
- (optional) Linux Device Drivers, 3rd Edition, by Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman
- (optional) Professional Linux Kernel Architecture, by Wolfgang Mauerer
    - VT students can access this book online for free, check instructions here
- (optional) Understanding the Linux Kernel, by Daniel P. Bovet, Marco Cesati

Grading (Tentative)

- Participation (5%)
- Exercise (6%: 2% x 3): small programming exercises, such as implementating your own system call, writing a Linux kernel module to extend a subsystem, etc.
- Paper reading (15%: 3% x 5): research papers about recent OS design and implementations, to educate you about the cutting-edge systems research and practice in industry
- Project (64%: 4% + 10% + 20% + 30%): A bunch of assigned kernel programming tasks and a final project
- Final exam (10%): course project report and/or presentation (TBD)

Policies

> ## Note
>
> All assignments must be turned in by the time/date indicated on the assignment. **No late project work will be assigned a grade.**

> ## Important
>
> **Cheating vs. collaboration**
>
> Collaboration is a very good thing. On the other hand, cheating is considered a very serious offense and is vigorously prosecuted. Vigorous prosecution requires that you be advised of the cheating policy of the course before the offending act.
>
> **For this semester, the policy is simple: don't cheat:**
>
> - *Never* share code or text on the project.
> - *Never* use someone else's code or text in your solutions.
> - *Never* consult project code or text that might be on the Internet.
> - *Never* publicly post your code on the Internet (e.g., GitHub)
>
> **On the other hand, for this class, you are strongly encouraged to:**

- Share ideas.
- Explain your code to someone to see if they know why it doesn't work.
- Help someone else debug if they've run into a wall.

If you obtain help of any kind, always **write the name(s) of your sources**.

Important

Honor Code Policy

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.

Adherence to Virginia Tech's honor code is expected in all phases of this course. All graded work, other than the in-class exercises, is expected to be the original work of the individual student. In working on the assignments, discussion and cooperative learning is encouraged. However, solutions are to be the work of the individual student. In all assignments you may discuss general concepts, such as algorithms, language syntax, Internet resources, or class and text topics, with others. However, copying of specific assignment program-code in the current or from previous semesters is an honor code violation. Any violations of the honor code will automatically be forwarded to the Office of the Honor System with the recommendation of an F* sanction as your final grade in the course. *I do use a code comparison system.*

If you have questions or are unclear about what constitutes academic misconduct on an assignment, please speak with me. I take the honor code very seriously in the course. The normal sanction I will recommend for a violation of the Honor Code is an F* sanction as your final course grade. The F represents failure in the course. The "*" is intended to identify a student who has failed to uphold the values of academic integrity at Virginia Tech. A student who receives a sanction of F* as their final course grade shall have it documented on their transcript with the notation "FAILURE DUE TO ACADEMIC HONOR CODE VIOLATION." You would be required to complete an education program administered by the Honor System in order to have the "*" and notation "FAILURE DUE TO ACADEMIC HONOR CODE VIOLATION" removed from your transcript. The "F" however would be permanently on your transcript.

For additional information about the Honor Code, please visit: https://www.honorsystem.vt.edu/

Special Accommodations

Reasonable accommodations are available for students who have documentation of a disability from a qualified professional. Students should work through Services for Students with Disabilities (SSD) in Lavery Hall. Any student with accommodations through the SSD Office should contact me during the first two weeks of the semester.

If participation in some part of this class conflicts with your observation of specific religious holidays during the semester, please contact me during the first two weeks of class to make alternative arrangements.

If you miss class due to illness, especially in the case of an exam or some deadline, see a professional in Schiffert Health Center. If deemed appropriate, documentation of your illness will be sent to the Dean's Office

for distribution to me. If you experience a personal or family emergency that necessitates missing class, contact the Dean of Students at 231-3787 or see them in 152 Henderson Hall.

Acknowledgments

Dr. Changwoo Min created this course at VT back in 2017 and credit goes to him regarding the course structure and content. This course also borrows materials from GT's CS 3210. UW's CSE 451, OSPP, and MIT's 6.828.