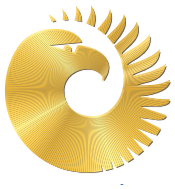

COM-341, Operating Systems

American University of Central Asia
Software Engineering Department

1 Course Description

This course offers a comprehensive introduction to operating systems, covering essential concepts such as the interaction between kernel and user space, process management, scheduling, memory management, and file systems. Students will gain hands-on experience with the Unix environment and both high- and low-level system programming in C and assembly. These skills will be applied in labs and projects involving the construction of Unix utilities, working with system calls, exploring the Linux kernel, and developing tools for FUSE-based file systems. The course also addresses performance optimization for server software using OS-specific APIs, equipping students with both the theoretical foundations and practical skills needed to understand and use modern operating systems as software engineers.

2 Course Information

Course Materials
`https://github.com/auca/com.341`

Course Codes
COM-341, COM-341.1

Course ID
3325

Prerequisites
COM-229, Data Structures
COM-123, Principles of Computing Systems

Prerequisites for
The list below depends on the student's year of admission.

COM-416.1, Computer Networks
COM-451, Parallel Programming

Credits
6

Time and Place
Lecture: Monday 12:45–14:00, Room 410
Lab: Wednesday 12:45–14:00, Room 410

3 Contact Information

Professor
Dmitrii Toksaitov
toksaitov_d@auca.kg

TA
Nikita Achkasov
achkasov_n@auca.kg

Office
AUCA, Room 315

Office Hours
Office hours are available by appointment, either on-site or remotely, during business hours (Monday–Friday). Please contact your professor or TA to schedule an appointment.

4 Topics

- Week 1–2: Introduction, History, OS Concepts Overview, Terminals (6 hours)
- Week 3–5: System Calls (9 hours)
- Week 6–8: Processes, Threads, Scheduling (9 hours)
- Week 9–11: Virtual Memory (9 hours)
- Week 12–14: File Systems (9 hours)
- Week 15–16: Networking (6 hours)

5 Learning Outcomes

By the end of this course, students will:

1. **Acquire programming skills to write and test system programs**:
   - Modify or develop system software such as standard Unix utilities, a command-line shell, and high-performance server software, to gain practical experience with system programming
   - Gain hands-on experience with kernel programming by making targeted modifications to the Linux kernel

2. **Evaluate and use appropriate tools and techniques for system software development**:
    - Use a standard Unix development environment, including editors, compilers, debuggers, and build systems
    - Work directly with low-level operating system (OS) APIs to build and optimize applications that interface with the kernel
    - Apply version control and collaborative development practices
3. **Design, plan, and critically evaluate system software solutions to problems**:
    - Model the fundamental interactions between user-space applications and the OS kernel via the system call interface
    - Design applications that leverage core OS abstractions, including processes, threads, scheduling, and virtual memory
    - Analyze system requirements to architect solutions for file management
4. **Evaluate low-level systems in terms of quality attributes and trade-offs**:
    - Analyze the performance implications of different CPU scheduling algorithms and virtual memory management techniques
    - Optimize server software by applying knowledge of OS-level APIs and their interaction with hardware resources
    - Evaluate the trade-offs between different file system designs and their impact on data persistence and access patterns
5. **Reflect and reason about information handling problems**:
    - Reason about the role of the operating system in managing shared resources, process isolation, and network communication
    - Critically examine techniques for managing concurrency using processes and threads
    - Analyze the principles of data persistence and abstraction as implemented in modern file systems
6. **Understand and apply ethical principles and professional standards**:
    - Recognize the ethical implications of system software design choices, particularly in terms of security, privacy, and resource management
    - Adhere to professional standards in collaborative software development, including code quality, documentation, and version control practices
    - Reflect on the societal impact of operating systems and system software in enabling modern computing environments

6 Assignments and Exams

6.1 Moodle/e-Course Checkpoints

Students must maintain instructor-provided private GitHub repositories for their assignments. They must periodically commit and push the required number of project solutions, as directed by the course staff. Instructors or teaching assistants will review the work either during lab sessions (on-site) or after the submission deadline (off-site) and award points based on completed assignments.

6.2 Labs and Projects

Throughout the course, students will be assigned several laboratory tasks. In these assignments, students will examine standard Unix utilities and then attempt to implement them in C on their own. The C versions should only utilize direct system calls, staying as close to the kernel as possible. Through these tasks, students will explore the interfaces between the OS kernel and userspace applications. The solutions written by students for their lab work will help them pass the practical assignments during the midterm and final examination sessions.

Students are required to complete four projects during the course. These projects encompass studying a toy shell, modifying the Linux kernel, writing utilities for a basic file system driver, and enhancing a web server using custom low-level OS APIs.

7 Course Materials, Recordings and Screencasts

All course materials are available on GitHub at `https://github.com/auca/com.341`. Using GitHub will help students become familiar with the Git version control system and the widely used GitHub platform.

We aim to record each class and upload it to YouTube for accessibility; however, recordings are not guaranteed. Recordings are produced on a best-effort basis as time permits. If you need immediate access, consider recording class sessions on your own computer. Links to YouTube recordings can be found in the course repository at `https://github.com/auca/com.341`.

While recordings provide flexibility, they are not a substitute for attending classes. Active participation is crucial for success in this course. Accumulating three or more unexcused absences may lead to an X grade. If overall class attendance is poor, the instructor reserves the right to discontinue class recordings.

Access lecture screencasts remotely via Zoom at either `http://com-341-zoom.auca.space`. When joining a Zoom session, students must identify themselves using their properly capitalized first and last names in the Latin alphabet. Your lab instructor may also use Zoom or other tools for remote work and set additional etiquette rules. Consult your instructor for more information. Install and configure the required remote tools on your computer during the first week of the semester so you can properly share your camera, microphone, and screen.

Remember that to attend lecture or lab classes remotely, you must obtain written permission from the instructor in advance. If such permission is not granted before the class begins, you will be marked absent (unexcused) in the attendance system (even if you join via the online platform). Please send a brief email to the instructor

explaining why you will attend remotely (e.g., illness, personal reasons) and, when possible, provide appropriate documentation to support your request to participate online, either immediately or later in the same email thread. If appropriate documentation is not provided within a reasonable timeframe in that thread, you will be marked absent without excuse and may be considered for a grade of X if you have three or more unexcused absences. AUAF students who are outside the country and can only participate online do not need to request this permission.

8 Software

Students are advised to install the following software on their machines at the start of the course. Additional installations may be required later.

- Git with SSH: `https://git-scm.com/downloads`

The compilers, assemblers, and debuggers will be available on the remote course server.

9 Hardware

There are no specific hardware requirements, as you can use lab machines where we guarantee you will be able to complete all the tasks of our course. However, we recommend a machine with more memory (rather than less), high-speed disk I/O, and a CPU that supports modern virtualization technologies.

10 Reading

1. Operating System Concepts, 10th Edition by Abraham Silberschatz (AUCA Library Call Number: QA76.76.O63 S5583 2013, ISBN-13: 978-1119456339, ISBN-10: 1119456339)

10.1 Supplemental Reading

1. Understanding the Linux kernel, Third Edition by Daniel P. Bovet and Marco Cesati (AUCA Library Call Number: QA76.76.O63 B683 2006, ISBN: 978-0596005658)
2. Linux Kernel Development, 3rd Edition by Robert Love (ISBN: 978-0672329463)
3. Windows Internals, Part 1 (6th Edition) by Mark E. Russinovich and David A. Solomon (AUCA Library Call Number: QA76.76.W56 R885 2012, ISBN: 978-0735648739)
4. Windows Internals, Part 2 (6th Edition) by Mark E. Russinovich and David A. Solomon (AUCA Library Call Number: QA76.76.W56 R885 2012, ISBN: 978-0735665873)
5. Mac OS X and iOS internals : to the apple's core by Jonathan Levin (AUCA Library Call Number: QA76.774.M33 L48 2013, ISBN: 978-1118057650)

6. Mac OS X Internals: A Systems Approach by Amit Singh (AUCA Library Call Number: QA76.76.O63 S564 2007, ISBN: 978-0321278548)

11 Grading

The preliminary distribution of points is outlined below. Please note that the distribution may change throughout the course if tasks are canceled, merged, or made optional (for bonus points). This usually happens for reasons such as software issues, online service outages, or classes canceled due to events outside our control.

Remember that some LMS platforms, such as Moodle, may not calculate final scores and grades correctly until all tasks have been published in the system and properly weighted. Do NOT assume your grade until all tasks are in the system and your instructor has notified you to review your scores and grades.

11.1 Moodle/e-Course Checkpoints

Your instructor will periodically announce reviews of your work. For such checks, you can be awarded up to the specified number of points.

- Project #1 (10%)
- Project #2 (10%)
- Project #3 (10%)
- Project #4 (10%)

11.2 Exams

In the middle and at the end of the course, you will need to pass the Midterm and Final examinations. These exams are significant and will greatly affect your grade.

- Midterm Exam (25%)
- Final Exam (35%)

11.3 Totals

100% is formed from the Moodle/e-Course Checkpoints (40%) and the two exams (60%).

11.4 Scale

– [94%–100]%: A
– [90%–94)%: A-
– [87%–90)%: B+
– [83%–87)%: B

- [80%–83)%: B-
- [77%–80)%: C+
- [73%–77)%: C
- [70%–73)%: C-
- [67%–70)%: D+
- [63%–67)%: D
- [60%–63)%: D-
- Less than 60%: F

Please note that requests for a higher grade due to points being marginally close will be ignored. For instance, 93.99 is an A-, NOT an A. Similarly, requests for extra assignments to boost points will also be disregarded.

12 Rules

First and foremost, in addition to all the rules listed in the syllabus document, students are required to follow the Code of Conduct of the American University of Central Asia.

12.1 Participation

Active participation during class may be rewarded with extra points at the instructor's discretion. Poor student performance during class can result in points being deducted from the final grade.

Instructors may conduct random pop-up checks during classes without prior notice. Students MUST be prepared for every class to avoid losing points. Students who are absent from classes with graded work without a valid reason will also lose points, unless their absence is due to force majeure circumstances. Instructors must be notified in advance of the reason for a student's absence in order to avoid a loss of points.

12.2 Questions

A question raised by one student is often relevant to others as well. Therefore, students are encouraged to use the online discussion board of the LMS (Learning Management System) they are enrolled in (e.g., AUCA e-Course System) to post questions publicly so that all students may benefit from the discussion.

Students must not post complete source code for any task on the LMS discussion board. Any such public post will result in a grade of zero for that assignment. In addition, students should refrain from submitting vague or overly general questions such as "Why doesn't my code work?"Instead, they are expected to carefully analyze and debug their code before seeking assistance.

12.3 Late Policy

Late submissions and late exams are not permitted. Exceptions may be granted at the professor's discretion only in cases of force majeure. If you become ill, experience

serious personal difficulties, or encounter technical problems with your computer or Internet, you MUST notify the instructors at least 24 hours in advance. Failure to do so will result in no extension being granted. Last-day requests will be considered procrastination. Additionally, no student will be granted more than one emergency extension during the course.

Beginning six hours before the deadline for any assignment, instructors will enter "silent mode."During this period, no questions related to the assignment will be answered, and no requests for office hours will be considered. At all other times prior to the deadline, instructors will make every effort to respond to questions and provide assistance, either via Zoom or in person during office hours.

12.4 Exam and Task Submission Ceremonies

Students MUST adhere to all exam and task submission procedures. This means they must strictly follow the rules specified by the instructors, whether provided in written or verbal form. Failure to comply will result in lost points. In your professional career, you will frequently work with formal documents (e.g., contracts, timesheets, reports), so it is important to develop accuracy and attention to detail early. Points will be deducted—or, in some cases, submissions may be rejected entirely—if the specified procedures are not followed. Any violation of deadlines or strict exam timing rules will result in a grade of zero.

12.5 Administrative Drop

Instructors reserve the right to drop a student from the course for non-attendance. If a student misses three or more classes without an acceptable excuse, the faculty may assign an X grade and remove the student from the course.

12.6 Incomplete Grade

Consistent with the late exam policy, a grade of I (Incomplete) will be granted only under highly exceptional circumstances. Students must initiate the request for an I grade well in advance of the final week of the semester. Requests made during the last week before final exams will not be considered.

12.7 Academic Honesty

Plagiarism is the act of copying or appropriating someone else's words or ideas and presenting them as one's own. In the context of this course, plagiarism may occur in many forms, including but not limited to program code, comments, software documentation, design specifications, requirement documents, project reports, and technical analyses.

More specifically, examples of plagiarism in a Software Engineering course include:

- Submitting code written by others or generated by AI as your own
- Modifying existing code or designs (e.g., changing variable names or restructuring code) and claiming originality

- Incorporating code snippets, sentences, design patterns, or intellectual content from any source without citation
- Using algorithms, patterns, or architectural designs without proper acknowledgment
- Using graphics, datasets, audio, video, or other materials from external works without attribution
- Purchasing or otherwise acquiring code, software, or project content and presenting it as original work

Plagiarism is unethical and undermines the educational process. The consequences are as follows:

- First instance: A grade of zero for the plagiarized work, and a report filed with the Registrar's Office
- Second instance: A failing grade (F) for the entire course

Both parties involved, the student who plagiarizes and the student whose work was used without authorization, will face equal consequences. It is the responsibility of every student to safeguard their assignments, code, and related materials to prevent unauthorized use.

The use of artificial intelligence, including generative AI tools, to complete certain assignments, projects, or exams (whether off-site or on-site) may be strictly prohibited. If AI involvement is suspected in submitted work, the student may be required to replicate the same or a similar task and respond to related questions in a supervised setting. Failure to do so satisfactorily will be regarded as evidence of AI-generated work, and the penalties described above will apply. Please note that for some tasks, the instructor may permit the use of AI technologies, but explicit written permission must be obtained in the assignment requirements before using them.

Finally, this course adheres to all global university policies on academic honesty. If university-wide regulations are stricter than those outlined here, the stricter rules will take precedence.

13 Access and Support Services

This course is committed to fostering an inclusive learning environment that supports the diverse needs of all students. If you have a disability or require specific accommodations to participate fully, please contact the instructors as early as possible. We will work with you to ensure that appropriate adjustments are made to support your learning experience.

For guidance on time management, presentation skills, writing, or study strategies beyond the scope of Software Engineering, we encourage you to connect with the Advising Office. The Advising Office offers practical workshops and peer advising to help you navigate academic challenges.

If you prefer working with peers, you may schedule an appointment with a student tutor through the Writing and Academic Resource Center (WARC). Tutoring is available both in person and online. Additional resources can be found on the WARC webpage.

If you are feeling stressed, overwhelmed, or struggling with emotions, relationships, or other mental health concerns, we encourage you to seek support from the AUCA Counseling Service. The Counseling Service provides confidential and free professional assistance to students.