COM-341, Operating Systems

American University of Central Asia
Software Engineering Department

1 Course Information

Course Codes
COM-341, COM-341.1

Course ID
3325

Prerequisites
COM-223.1, Algorithms and Data Structures
COM-410.1, Computer Architecture & Organization

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

COM-392, System Programming
COM-416.1, Computer Networks
COM-431, Senior Project Preparation I/II
COM-451, Parallel Programming

Credits
6

Time and Place
Lecture: Monday 14:10–15:25, room 410
Lab: Friday 14:10–15:25, lab 410

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

2 Contact Information

Professor
Dmitrii Toksaitov
toksaitov_d@auca.kg

Office
AUCA, room 315

Office Hours

By appointment throughout the work week (write to your professor to make an appointment)

3 Course Overview

This course introduces students to the fundamentals of operating systems design and implementation. Topics include an overview of the components of an operating system, kernel and userspace communication, implementation of processes, scheduling algorithms, memory management, and file systems. In addition, students will learn the basics of the Unix environment, practice with the C programming languages, and the x86/ARM/RISC-V... assembly. These technologies will help them finish lab and project tasks to build standard Unix utilities, study the concept of systems calls, peak into the inner workings of the Linux kernel, write a system utility to maintain a Fuse file system, accelerate a web server by using OS-specific APIs available on the platform at hand.

As a result, students should be able to research and analyze the functioning of the information technology systems, improve their skills using programming languages for software design, development, and maintenance in accord with the goals of the AUCA Software Engineering Department and the 510300 IT competency standard.

4 Topics Covered

- 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 Assignments and Exams

5.1 GitHub Checkpoints

Students will have to maintain a personal private GitHub repository with all their works shared with their instructor. Students have to periodically commit and push a specific number of lab and project solutions as told by the instructor. The instructor will regularly check the work and give points for the accomplished work.

5.2 Labs and Projects

Students will have a number of laboratory tasks and have to finish several course projects. Students will have to defend their work to the instructor during midterm and final examination sessions.

Students will have to finish several lab tasks. In every assignment, students will study a standard Unix utility and try to implement it in C on their own. The C version will only have to use direct system calls and be as close to the kernel as possible. Through such tasks, students will study the interfaces between an OS kernel and userspace applications.

Students are required to do four projects. The course projects include studying a toy shell, modifying the Linux kernel, writing utilities for a simple file system driver, and accelerating a web server using custom low-level OS APIs.

6 Course Materials, Recordings and Screencasts

Students will find all the course materials on GitHub. We hope that by working with GitHub, students will become familiar with the Git version control system and the popular (among developers) GitHub service. Though version control is not the focus of the course, some course tasks may have to be submitted through it on the GitHub Classroom service.

Every class is screencasted online and recorded to YouTube for students' convenience. The course Zoom link is `http://com-341-zoom.auca.space`. YouTube links can be found on the course repository at `https://github.com/auca/com.341`. An ability to watch a class remotely at any time MUST NOT be a reason not to attend classes. Active participation is necessary to succeed in this course. Class recording may stop if the instructor feels the attendance and participation are unsatisfactory.

7 Software

Students are recommended to install the following software on their machines.

- VirtualBox: `https://www.virtualbox.org/wiki/Downloads`
- Git with SSH: `https://git-scm.com/downloads`

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

8 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)

8.1 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)

9 Grading

9.1 GitHub Checkpoints

Your instructor will announce a periodic review of your work. You will be awarded up to the following number of points for such checks.

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

9.2 Interview

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

9.3 Totals

- 100% is formed from the GitHub submissions (40%) and the two exams (60%).

10 Scale

- [92%–100]%: A
- [85%–92)%: A-
- [80%–85)%: B+
- [75%–80)%: B
- [70%–75)%: B-
- [65%–70)%: C+
- [60%–65)%: C
- [55%–60)%: C-
- [50%–55)%: D+
- [45%–50)%: D
- [40%–45)%: D-
- Less than 40%: F

Please, note that requests to award a better grade if the number of points is close to such a grade will be ignored. For example, 91.99 is A-, NOT A. Likewise, requests to get extra assignments to increase the number of points will also be overlooked entirely.

11 Rules

Students are required to follow the rules of conduct of the Software Engineering Department and the American University of Central Asia.

11.1 Participation

Active work during the class may be awarded with extra points at the instructor's discretion. Poor student performance during a class can lead to points being deducted from the final grade.

Instructors may conduct pop-checks during classes at random without prior notice. Students MUST be ready for every class in order not to lose points. Students absent without a good reason from such classes with graded work will also lose points unless it is force-majeure circumstances. Instructors must be notified in advance about why a student is absent not to lose points.

11.2 Questions

We believe that a question from one student is most likely a question that other students are also interested in. That is why we encourage students to use the online discussion board of the LMS (Learning Management System) that you use (e.g., AUCA e-Course System or Canvas) to ask questions in public that other students can see and answer. We discourage students from asking questions through E-mail.

If it is a private matter, write direct messages to your instructor through the LMS system (such as AUCA e-Course System or Canvas) too. We will not be answering most E-mail messages this semester (unless it is a severe emergency) to consolidate all the course correspondence in one place.

Do not post the complete source code for any task on the LMS discussion board. You will get zero for that work for any such public post. Do not ask generic questions about your code to know why it does not work. Please spend some time thinking about your code, debugging it.

11.3 Late Policy

Late submissions and late exams are not allowed. Exceptions may be made at the professor's discretion only in force-majeure circumstances. If you got ill, got severe personal issues, got problems with your computer or the Internet, you MUST notify instructors at least 24 hours in advance. Otherwise, we will not give you an extension. We will consider that you were procrastinating until the very last day. We will also not be giving more than one emergency extension throughout the course.

Six hours before the deadline for any work on the course, instructors will go into a silent mode. No questions will be answered about the work that has to be submitted, no requests to have office hours will be considered. However, at any other work time before the deadline, we will try our best to answer your questions and help you through Zoom or in our office.

11.4 Exam and Task Submission Ceremonies

Students MUST follow exam and task submission ceremonies. It means they MUST strictly follow all the rules specified by the instructors in written or verbal form. Failure to do so will result in lost points. Throughout your career, you will have to work with various supporting documents (contracts, timesheets, etc.). It is a good idea to start learning to work with such documents accurately early. We will remove points for not following these rules or even refuse to accept your exam defense or tasks submitted to us. We will also give zero for not following deadlines or the strict exam timing rules.

11.5 Incomplete Grade

As with late exams, the grade *I* may be awarded only in exceptional circumstances. The student must start a discussion on getting the grade *I* with the instructors in advance and not during the last week before the final exams.

11.6 Academic Honesty

Plagiarism can be defined as "an act or an example of copying or stealing someone else's words or ideas and appropriating them as one's own". The concept of plagiarism applies to all tasks and their components, including program code, comments, documentation, abstracts, reports, graphs, statistical tables, etc.

The following are examples of some common acts of plagiarism:

1. Representing the work of others as their own
2. Using other people's ideas or phrases without specifying the author
3. Copying code snippets, sentences, phrases, paragraphs or ideas from other people's works, published or unpublished, without referring to the author
4. Replacing selected words from a passage and using them as your own
5. Copying from any type of multimedia (graphics, audio, video, Internet streams), computer programs, graphs or diagrams from other people's works without representation of authorship
6. Buying work from a website or from another source and presenting it as your own work

In addition to being unethical, this indicates that the student has not studied the given material. Tasks written from somewhere for 5% or less will be assessed accordingly or will receive a 0 at the discretion of the teacher. If plagiarism is more than 5%, the case will be transferred to the AUCA Disciplinary Committee.

In this course, teamwork is NOT encouraged. The same blocks of code or similar structural pieces in separate submissions will be considered academic dishonesty, and all parties will get zero for the task.

Students are not recommended to memorize lab and project code before exams, as this is a difficult and inefficient way to learn; and since practice exams may consist of open questions designed to test a student's analytical skills, memorization invariably leads to the fact that the answers are inappropriate and of poor quality.