COM 341, Operating Systems

American University of Central Asia
Software Engineering Department

1 Course Information

Course Code
COM 341

Course ID
3325

Prerequisite
None

Credits
6

Professors, TAs, Time, Place
Lecture (Dmitrii Toksaitov): Monday 10:50–12:05, 220
Lab (Dmitrii Toksaitov): Monday 12:45–14:00, G31
Lab (Dmitrii Toksaitov): Monday 14:10–15:25, G31
Lab (Dmitrii Toksaitov): Wednesday 14:10–15:25, G31
TA Consultations (Bektur Umarbaev): By appointment

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

Class Discussions
`https://piazza.com/auca.kg/fall2019/com341`

2 Contact Information

Instructor
Toksaitov Dmitrii Alexandrovich
toksaitov_d@auca.kg

Teacher Assistant
Bektur Umarbaev
umarbaev_b@auca.kg

Office
AUCA, room 315

Office Hours

By appointment throughout the work week
Remotely through Skype on Saturday and Sunday from 12:00 to 18:00

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, synchronization, implementation of processes, scheduling algorithms, memory management and file systems. Students will learn basics of the Unix environment, the C programming language, and the ARM/x86 assembly. These technologies will help them to finish lab and project tasks to build common Unix utilities, to study the concept of systems calls, to peak into the inner workings of the Linux kernel, and to implement a simple Fuse file system.

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 to the goals of the AUCA Software Engineering Department and the 510300 IT competency standard (including competency elements OK 17, 17, 115).

4 Topics Covered

- Week 1: Introduction, History, OS Concepts Overview (3 hours)
- Week 2: System Calls (3 hours)
- Week 3: Scheduling (3 hours)
- Week 4: Interprocess Communication (3 hours)
- Week 5: Segmentation (3 hours)
- Week 5–7: Virtual Memory Management (9 hours)
- Week 8: Page Replacement Algorithms (3 hours)
- Week 9: Swapping (3 hours)
- Week 10–11: File System Implementation (6 hours)
- Week 11: Protection Mechanisms (3 hours)
- Week 12: Principles of I/O (3 hours)
- Week 13: Deadlocks (3 hours)
- Week 14–16: RAM Disks, Disks, Terminals (9 hours)

4.1 Lectures

Students will have to take midterm and final examinations on topics discussed during lectures. Each examination is in the form of a quiz with a set of open and multiple choice questions.

5 Practice Tasks and Labs

Students are required to finish 3 practice tasks during the course. These tasks are based on topics discussed during lectures.

Students will have to finish 10 lab tasks. In every task students will study a common Unix utility and try to implement it on their own.

6 Course Project

The course project is to develop an working multitasking operating system for Raspberry Pi. The OS should include a scheduler, a virtual memory manager, a file system, and a working set of Unix-like programs and utilities developed during practice tasks and labs.

7 Course Materials, Recordings and Screencasts

Students will find all the course materials on GitHub. We hope 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 screen casted online and recorded to YouTube for students convenience. An ability to watch a class remotely MUST NOT be a reason to not attend the class. Active class participation is necessary to succeed on this course.

8 Reading

1. Operating System Concepts, 10th Edition by Abraham Silberschatz (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

- Lecture Midterm (15%)
- Lecture Final (15%)
- Practice tasks (25%)
- Labs (15%)
- Course project (25%)
- Piazza Participation (5%)

10 Scale

- 92%–100%: A
- 85%–91%: A-
- 80%–84%: B+
- 75%–79%: B
- 70%–74%: B-
- 65%–69%: C+
- 60%–64%: C
- 55%–59%: C-
- 50%–54%: D+
- 45%–49%: D
- 40%–44%: D-
- Less than 40%: F

11 Rules

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

11.1 Participation

Active work during the class may be awarded with up to 5 extra points at the instructors discretion.

Poor student performance during a class can lead to up to 5 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 loose points.

11.2 Attendance

Lecture classes has 5% attendance/participation that is

Practice classes has More than three misses without a reason will result in 10 points being deducted from the student for every day. If a student has health/family/personal emergency, he MUST notify the instructor in advance (e.g., through e-mail). The student MUST also provide a valid proof afterwards. Without a prior notice and a valid proof the miss will still be counted.

11.3 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 Piazza to ask questions in public that other students can see and answer and NOT ask them through E-mail in private UNLESS the question itself is about private matters to discuss with the professor.

11.4 Late Policy

Late submissions and late exams are not allowed. Exceptions may be made at a discretion of the professor only in force-majeure circumstances.

11.5 Incomplete

As with late exams, the grade *I* may be awarded only in special circumstances. The student must start 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 elses words or ideas and appropriating them as ones own. The concept of plagiarism

applies to all tasks and their components, including program code, abstracts, reports, graphs, statistical tables, etc.

In addition to being unethical, this indicates that the student has not studied the given material. Tasks written from somewhere for 10% or less will be assessed accordingly or will receive a 0 at the discretion of the teacher. If plagiarism is more than 10%, the case will be transferred to the AUCA Disciplinary Committee.

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

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

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