COM 341, Operating Systems

American University of Central Asia
Software Engineering Department

1 Course Information

Course ID
COM 341, 3325

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

Place
AUCA, laboratory 233

Time
Group A: Tuesday 12:45
Group A: Thursday 12:45
Group B: Tuesday 14:10
Group B: Thursday 14:10

2 Prerequisites

- COM-117, Programming II. Object-oriented Design
- COM-223, Algorithms and Data Structures (eng)

3 Contact Information

Instructor
Toksaitov Dmitrii Alexandrovich
toksaitov_d@auca.kg

Office
AUCA, room 315

Office Hours
Wednesday 9:00–15:25

4 Course Overview

The course introduces students to the fundamentals of operating systems design and implementation. Topics include an overview of the components of an operating system, implementation of processes and threads, scheduling algorithms, memory management and file systems. This course is designed for Software Engineering majors and minors.

5 Topics Covered

Processes
- Scheduling
- Inter-process Communication

Memory Management
- Segmentation
- Virtual Memory Management
- Page Replacement Algorithms
- Swapping

File Systems
- File System Implementation
- Protection Mechanisms

Input & Output
- Principles of I/O Hardware & Software
- Deadlocks
- RAM Disks
- Disks
- Terminals

6 Exams

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.

7 Practice Tasks

Students are required to finish four practice tasks during the course.

In the first task students will take a look on how communication with the kernel works from the user space of an operating system. The task is to make a simple interactive shell not to rely on any C run-time or standard libraries. Students will do that by making system calls directly to the kernel on Linux, FreeBSD, or macOS

in x86, or ARM assembly.

In the second task students will take a look on how system calls are processed on the kernel side. They will have to add a new subsystem to a Linux kernel and a number of system calls to query information about running processes without using the */proc* virtual file system.

The third task is to study the scheduling subsystem of the Linux kernel and to modify it to allow requests from processes owned by privileged users for the maximum possible amount of CPU time for the calling process and its parents.

The final task is to get a look on an implementation of a custom user space file system for FUSE on *nix or Dokany on Windows. Students will have to write several management utilities to create and defragment the file system at hand.

8 Course Projects

Students are required to finish two course projects.

The first project is to develop a limited simulation of an OS kernel on top of an emulated computer system. In a simplified environment students will get a chance to build all major parts of a working preemptive kernel such as a scheduler, a virtual memory manager, and a file system driver in a high-level language.

The second course project is to port a working subsystem from the simulated computer environment from the first project to an x86 kernel developed at AUCA. Students will have to port their code from a high-level language to C and assembly and adapt it to work on a real hardware.

9 Reading

1. Modern Operating Systems, Fourth Edition by Andrew S. Tanenbaum and Herbert Bos (ISBN: 978-0133591620)

9.1 Other Reading Recommendations

1. Operating System Concepts, 9th Edition by Abraham Silberschatz, Peter B. Galvin, Greg Gagne (ISBN: 978-1118063330)

2. Operating System Concepts Essentials by Abraham Silberschatz and Peter B. Galvin, Greg Gagne (ISBN: 978-1118804926)

9.2 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)

10 Grading

- Class participation (through Piazza) (5%)
- Exams (20%)
- Practice tasks (35%)
- Course projects (40%)

– 90%–100%: A
– 80%–89%: A-
– 70%–79%: B+
– 65%–69%: B
– 60%–64%: B-
– 56%–59%: C+
– 53%–55%: C
– 50%–52%: C-
– 46%–49%: D+
– 43%–45%: D
– 40%–42%: D-
– Less than 39%: F

11 Rules

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

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