COM 392, 451, System Programming or Parallel and Distributed Programming

American University of Central Asia
Software Engineering Department

1 Course Information

Course Codes
COM-392
COM-451

Course IDs
4953, 3708

Prerequisite
COM-341, Operating systems
or
COM-410, Computer Architecture

Credits
6

Professors, TAs, Time, Place
Lecture (Dmitrii Toksaitov): Wednesday 10:50–12:05, Online
Lab (Dmitrii Toksaitov): Wednesday 12:45–14:00, Online
Lab (Dmitrii Toksaitov): Wednesday 14:10–15:25, Online

Course Repository
`https://github.com/auca/com.392`

Class Discussions
`https://piazza.com/auca.kg/fall2020/com392/home`

Student Informal Discussions
`https://auca-sfw.slack.com/archives/G019KJJ3SDB`

2 Contact Information

Professors
Dmitrii Toksaitov
toksaitov_d@auca.kg

Office
AUCA, room 315

Office Hours
By appointment throughout the work week remotely through Zoom

3 Course Overview

The course introduces students to the topic of programming multi-core multi-processor systems with data-parallel facilities of various hardware on a single machine or in a distributed system connected to a high-performance network. The students will learn the most popular shared-memory parallel programming API such as Pthreads and CUDA, and a distributed memory programming API such as MPICH. Students will get a chance to work on two projects accelerating image processing tasks and astronomy simulations. Results of the projects will be tested together with students on high-performance parallel machines on the Amazon's cloud.

At the end of the course student should be able to research, analyze, design, develop, and maintain functioning parallel software systems in accord to the goals of the AUCA Software Engineering Department and the 510300 IT competency standard (OK 17, 17, 115).

4 Topics Covered

- Flynn's taxonomy
- Amdahls law
- CPU caches and locality
- CPU pipelines and branch prediction
- Data-parallelism with SIMD instructions
- Hardware multithreading
- Shared and distributed memory systems
- NUMA and UMA architectures
- Synchronization
  - Spinlocks, barriers, mutexes, semaphores
  - Conditional variables
  - Race conditions
  - Deadlocks
- General-purpose computing on graphics processing units

5 Examinations

Students will get midterm and final examinations in the form of interviews, where they have to defend their labs and projects to the instructor.

6 Course Projects

Throughout the course, students will work on three major projects to accelerate programs performing image processing tasks and astronomy simulations.

6.1 Project #1

In the first projects, students will be asked to accelerate a Sobel and Median filters in an image-processing program by utilizing all the cores of a test machine through the Pthread API. They may also accelerate their solutions even further by utilizing the parallel vector-processing facilities such as SIMD instructions of modern CPUs. For extra points, students may move the image processing algorithms to the GPU and compare the achieved speed up.

6.2 Project #2

The second project requires accelerating the N-body simulation in a distributed environment with a high-speed interconnection between machines through the Message Passing Interface (MPI). Students will compete on how many planetary bodies can their accelerated systems handle compared to others.

6.3 Project #3

The third project requires building a toy ray tracer to render computer graphics scenes. Ray tracing should be accelerated with the use of a graphics processing unit and the CUDA API.

7 Reading

An Introduction to Parallel Programming by Peter Pacheco (ISBN: 978-0123742605)

7.1 Supplemental Reading

1. Parallel Programming, 2nd Edition by Thomas Rauber and Gudula Rnger (ISBN: 978-3642048173)
2. Computer Architecture: A Quantitative Approach, 5th Edition by David Patterson and John L. Hennessy (ISBN: 978-0123838728)

8 Grading

- Class participation (through Piazza) (5%)
- Course projects (25% + 35% + 40%)

8.1 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

9 Rules

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

9.1 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 lose points.

9.2 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 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.

9.3 Late Policy

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

9.4 Exam Ceremonies

Students MUST follow exam ceremonies. It means they MUST prepare task list forms with all points appropriately calculated. They MUST submit them correctly. They must bring task list forms to the exam. 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.

9.5 Incomplete

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.

9.6 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 teamwork 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