EECS678 Introduction to Operating Systems
General Information – Spring 2020

Class Information: As below:
Lectures: Room: LEEP2 G415, Time: MWF 2:00PM – 2:50PM
Labs: Room: Eaton 1005D, Time: Mon 12:00PM – 1:50PM,
Room: Eaton 1005D, Time: Mon 4:00PM – 5:50PM,
Room: Eaton 1005D, Time: Wed 9:00AM – 10:50AM
Room: Eaton 1005D, Time: Wed 4:00PM – 5:50PM, and
Room: Eaton 1005D, Time: Fri 12:00PM – 1:50PM

Prerequisites: EECS388 (Computer Systems and Assembly language) and EECS448 (Software Engineering I)

Instructor: Prasad Kulkarni
Office: 2001F Eaton (Ph: 785-864-8819)
Office Hours: 3:00PM – 4:00PM Tuesday and Friday, or by appointment
Email: prasadk@ku.edu
Other Location: 136 Nichols

Teaching Assistants: Ahmet Soyyigit and Ishrak Hayet
Office: Eaton 2041
Office Hours:
Ahmet: Tuesday 2:30 – 3:45PM and Thursday 3:00 - 3:45PM
Ishrak: Monday: 2:30 – 03:30PM and Tuesday: 1:00 – 2:00PM
Email: Ahmet Soyyigit (ahmet.soyyigit@ku.edu); Ishrak Hayet (ishrakhayet@ku.edu)

Texts: Operating System Concepts by Silberschatz, Galvin, and Gagne, 9th (or recent) Edition, Wiley Publications, ISBN: 978-1118063330

Class Homepage: The class home page is at http://www.ittc.ku.edu/∼kulkarni/teaching/eecs678/. The page will contain a variety of information, which will include the syllabus, schedule, slides, and assignments.

Course Objectives: Objectives are:

1. Define and explain key operating system concepts – operating system, process and process management, threads, scheduling, synchronization and deadlocks, main memory and virtual memory, file-system and I/O interface, security.
2. Apply key operating system concepts – several OS algorithms and concepts are useful in other areas of computer science.
3. Develop system-level programs – understanding practical OS issues, such as OS level system calls and APIs, other issues in OS (kernel) implementations.

Course Structure: Structure is:

- Theoretical OS concepts will be discussed during the lectures.
- Some practical or implementation OS aspects may also be discussed during the lectures.
- Practical concepts, and OS programming using OS API and system calls will be re-enforced during lab sessions and projects.

- Grades will be based on two exams, in-class quizzes, labs, and up to three programming assignments.

Slides: There is a lot of material to cover in this class. Lecturing from slides will allow me to cover the material at a more rapid pace. I will be presenting slides that I have developed along with figures and tables from the text. Slides and additional material that I have developed for the class will be made available from the class homepage prior to their presentation.

Grading: Grades will be based on two exams (15% + 20%), in-class quizzes (20%), continuous programming assignments (20% total), and lab evaluations (25% total). Keep all graded material to provide evidence of grades. A final comprehensive exam may be given in place of the second exam.

Attendance and Punctuality: Roll is not taken, but you are responsible for all material presented in class. Exams and due dates will be scheduled in advance. A grade of zero will be recorded for missed exams and late assignments unless prior arrangements are made. Assignments turned in after the due date, but by the beginning of the next scheduled class will be penalized 10%. Assignments will not be accepted that are more than one class period late.

Cheating: Students are encouraged to discuss programs in general and to help one another find bugs in existing programs. Copying another's code or writing code for someone else is cheating. All submitted programming assignments will be checked using automatic plagiarism detection programs. Keep listings to provide evidence of creative development.

Programming Projects: There will be two programming projects. These are to update/write certain portions of a mature research OS. Assignments will be discussed in class lectures, and expanded during the lab sessions. Some programming assignments may be performed in groups.

Discussions / Lab Sessions: Attending the lab sessions will provide hands-on experience on several OS concepts introduced in class lectures. Most practical OS components will be discussed during the lab sessions. Assignments will also be covered in more depth in the labs. Tools to assist in assignment coding and debugging will be discussed in the lab discussions.

Your Responsibilities: Responsibilities are:

1. Understand lecture and reading material.
2. Attend office hours for extra help, as often as required.
3. Uphold academic honesty.
4. Turn in assignments on time.
5. Check class web-page and your email account regularly.

OS involves many concepts, and some may be inter-dependent. To really understand the class you will need to read the lecture notes and the book *repeatedly.*

Please provide information on the class if you feel that the class is too easy or too hard, you don't have the necessary background, or any general improvements that can be made to the class.

Please advise the instructor of this class at your earliest convenience (minimum of five working days) if you have a disability that will require a reasonable accommodation for any of the activities in the course schedule.