CS 4348/5348 Operating Systems

NEWS

- **If you still have problem with your UTD account, check here.**
- If you have problem accessing the videos of the first week lectures, I have retrieved it from MS Stream and posted it here.

Helpful URLs for Honorlock Exams

The exams for this course will be conducted online in elearning and monitored by Honorlock. If you are not experienced with Honorlock, I have prepared the "Survey with Honorlock" item in elearing for you to try out. If you have problems with the computing facilities for taking an Honorlock exam, please check below.

- Please reference the instructions and Q&A about Honorlock exams first.
- If you need a webcam and/or a laptop that are capable of handling Honorlock exams, please check one out from the university. Please do this ASAP because the resources are limited.
- If you have problem with your Internet connection or if you do not feel comfortable to use your own facility for the exams, then you can take the exam on campus at the Founders Brazos Lab. Please inform me through the survey (clearly state that you want to take the exam on campus). This has to be done now and there is no way to request it later.

Homeworks and Exams

- Review for Exam 3.

Policies related to homework assignments

- The main purpose of homework assignments is to help you understand the material better and to help you prepare for the exams.
- No homework will be accepted after 11:59pm of the due date. Answers to homework assignments can be submited through elearning or in class.
- Homework questions in one assignment will be given out progressively after the corresponding materials are covered. The due date for the homework will be very soon after the last questions are posted. You are supposed to work on the questions right after the materials are covered, not to wait till all questions are posted.

Projects

- Project descriptions are posted on elearning.
- Note: All projects are given 3 day late due period after the due date with a 5 points (out of 100 points) penalty. After 3 days, we will not accept any submission, unless there are critical circumstances, such as medical conditions (with proof).
- Note: We will not answer any question about the project when it is within 2 days of the project due date. Please start early for any unexpected problems.

- About your submissions.
  - You have to follow the file naming and command line input specifications exactly. TA will run your programs the same way as specified in the handout. If your code does not work by such testing, you will not get any credit.
  - If you do have some reason and have to deviate from the spec, you have to specify how to run your code and what to expect in the "readme" file (there is no need to submit the readme file if you have no special instructions for TA). Of course, if you do deviate from the spec, you will lose some points.
  - If your project does work, but has failed TA's testing, you may discuss with TA to see what to do. If it is because that you did not follow the spec and did not provide a readme file, then even if your code works perfectly, you will still incur significant point deduction.
  - These are all for easing TA's grading task. TA will have to spend a lot of time to grade the projects. Your cooperation is highly appreciated.

- Project 1:
  - Code and input to be used for your "makefile" and "exec.sh" are in this tar file.
  - Available Linux servers and some basic shell commands.
  - A simple sample makefile.

- Projects 2&3&4:
  - Code and input files to be used for this project are included in this tar file.
  - New cpu.o that handles adcmdInterrupt and submitInterrupt (with cpu.c, simos.h) is in this tar file.
    - For those who work directly on Department Linux servers, you can simply copy the tar files (path names below) instead of downloading.
    - /people/cs/i/ilyen/public_html/course/os/for21s/simos-proj.tar
    - /people/cs/i/ilyen/public_html/course/os/for21s/client.tar
  - Due dates
    - **Project 2**: Phases 1&2 due 3/5; Phase 3 due 3/11.
    - **Project 3**: Phases 1&2 due 3/17; Phases 3&4 due 3/29.
    - **Project 4**: Phase 1 due 4/9; Phase 2 due 4/16; Phase 3 due 4/23.

  - Some common Questions and Answers can be found here.

  - Sample programs for Unix (Linux) system calls.
    - Example fork and pipe program

- Example server socket
- Example client socket
- Some thread programming examples: with sync problem, with argument passing problem, all fine.
- Example signal program

  - simOS changes
    - Bug fix (2/6): For the submit command, if you give a non-existing file name, there will be an infinite loop by failed scanf(). The bug has been fixed. The tar file now has a new loader.c and a recompiled simos.exe.
    - C99 version of gcc warnings (2/13): Definitions for some external functions are missing in simos.h. Some header files are missing in .c code. All fixed now. If you use C99 or C11 of gcc, now you will not get many warnings, only warning will be usleep and you can ignore them.

  - The due date for the last phase of each project is the final due date of the project. TA will test your code only after the final due date of each project, and you will only get one grade for each project. However, you do need to submit your code for each phase on the due date given above.
  - There will only be a single entry in elearning for each project. So, you will need to submit all phases of the same proect via the same entry. We will be able to see all your submissions. Also, you will only see one due date listed in elearing (the final one).
  - When testing your code, if your code fully works (minor issues are fine), your earlier submissions will be ignored. If your code does not work well, TA may choose to test the earlier versions of your code to make sure that you did spend effort early on and the grading will take your earlier versions into account.
  - It is recommended that once you start a project, just try to complete the project fully. Sometimes, doing the work in one shot may be easier.

- Some web resources
  - Makefile Tutorial
  - Unix Shell Basics
  - General Introduction to Shell Programming
  - C Programming
    - This tutorial quickly gives you all the basics, a very good start. C Tutorial (Beginners)
    - This tutorial gives more information on some aspects of C, good introduction to some library functions. C Programming Tutorial
    - This tutorial discusses some advanced features, some are unlikely to be useful, but interesting read. It provides more advanced aspects of C, which are not even in the book I recommend. C Tutorial for Geeks
  - Basic Socket Tutorial
  - Good pthread Tutorial

Policies related to projects

- Working knowledge of C++ and Unix is required.
- Projects that do not work will not get partial credits.

- Project testing will be performed on CS Linux servers. You can develop your code on any computer, but have to test your code on University Linux systems.
- Debugging help: As a CS student, debugging is a skill that you need to develop. Go through the program logic carefully and print out the results after each major step (best is to use a debugger to do this). Please do not seek for help in debugging your program logic. Do it by yourself. I have instructed TA not to debug for students.
- You can get help if you do not understand some compilation errors or if you are sure a system call is giving you problem and do not know how to make it work correctly.
- No discussions among students are allowed for project assignments. Questions regarding assignments can be discussed with the TA or instructor. **At the end of the semester**, we will conduct plagiarism detection on all projects. Students who are involved in plagiarism will get an F grade for the course.