CS A253L: PRINCIPLES IN SYSTEM DESIGN LAB

Item	Value
Curriculum Committee Approval Date	12/02/2020
Top Code	070600 - Computer Science (Transfer)
Units	1 Total Units
Hours	54 Total Hours (Lab Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Grading Policy	Standard Letter (S)

Course Description

This lab is required by four-year institutions, coupled with CS A253, to satisfy lower-division work that prepares students for upper-division work in Computer Science. PREREQUISITE: CS A150 and CS A170. Transfer Credit: CSU; UC.

Course Level Student Learning Outcome(s)

1. Students will be able to write and compile code in the C programming language from the command-line.
2. Students will be able to use various POSIX API functions in their programming.

Course Objectives

- 1. Write and compile code in the C programming language from the command-line.
- 2. Navigate from the command-line in POSIX
- 3. Use debugging tool gdb from the command-line.
- 4. Use various POSIX API functions in their programming.

Lecture Content

This is a lab-only course and must be taken concurrently with CS A253.

Lab Content

Hello World in C Text editing in a Linux environment using vim Basic C compilation The importance of man pages Pointers Memory Introduce/ Re-introduce pointers and pointer arithmetic Apply gdb skills for debugging to understand code Visualize the memory of a process Utilize the dynamic memory allocator Work with pointers and space on the heap Structs Investigate data alignment in memory through the use of structs and their associated sizes in memory Use structs to visuallize and access memory. Processes Fork Investigate the operation of the fork( ) function call Introduce foreground and background processes. Signals Signal Handlers Investigate the operation of the signal( ) function call Investigate the operation of the sigprocmask( ) function call Read more about blocking, ignoring and handling signals. Implicit Memory Allocator Investigate the operation of an implicit first-fit memory allocator Work with basic tests cases for illustrating the memory allocator behavior Visualize how the allocator manages its space Network Socket Programming Learn the basics of socket programming in C Use and modify a simple client/server application with a customized application-level protocol Concurrent Programming Investigate three ways to handle concurrent connections on the server side: Multiprocessing, Multithreading and Multiplexing Examine common pitfalls and complexities of each approach See the use of semaphores for synchronization

Method(s) of Instruction

- Lab (04)
- DE Live Online Lab (04S)
- DE Online Lab (04X)

Instructional Techniques

Students will write code that will gives them experience working in working in the Linux environment. Students will be shown examples that demonstrate various concepts and practices in programming in the C language.

Reading Assignments

Students will read topics assigned that are necessary to complete labs.

Writing Assignments

Students will spend a minimum of 6 hours per week writting code.

Out-of-class Assignments

Students will spend a minimun of 6 hours per week completing weekly programming assignments.

Demonstration of Critical Thinking

Based on participation in the lab work, and the lab reports.

Required Writing, Problem Solving, Skills Demonstration

Successful performance of the assignments.

Eligible Disciplines

Computer science: Masters degree in computer science or computer engineering OR bachelors degree in either of the above AND masters degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelors degree in engineering AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in mathematics AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in any of the above AND a masters degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum qualification purposes, under the discipline of the application. Masters degree required.

Textbooks Resources

1. Required Bryant, R.E., OHallaron, D.R.. Computer Systems: A Programmers Perspective, 3rd ed ed. Santa Monica, CA: Pearson, 2016