Master Course Syllabus
School of Engineering and Computer Science
Washington State University Vancouver
CS 360
Systems Programming
4 Semester Hours
(3 lecture hours, 3 laboratory hours)

Catalog Description
Implementation of systems programs, concepts of computer operating systems; laboratory experience in using operating system facilities.

Prerequisite Courses
- CS 122 with a C or better
- CS 251 (or CS 261) with a C or better

Prerequisite Topics
- Previous experience with the C programming language
- Linked list data structures using C pointers
- Sorting and searching algorithms
- Use of Unix environment for coding, compilation, debugging and testing
- Assembly language concepts, included procedure calling conventions
- Hardware architectural concepts, including memory and I/O structures

Measured Course Outcomes
Students taking this course will:
1. Break down a programming problem into subtasks to implement a solution using the principles of encapsulation and data hiding. (Contributes to performance criterion 1-a.)
2. Design algorithms or data structures to meet the performance requirements of a systems programming problem. (Contributes to performance criterion 6-b.)
3. Solve systems programming problems by selecting or implementing appropriate algorithms and data structures.(Contributes to performance criterion 6-c.)
4. Demonstrate proficiency in the use of the C programming language, and related software development tools, as found in a Unix-based operating system environment. (Contributes to performance criterion 6-d.)

Required Textbooks
The Linux Programming Interface, Michael Kerrisk, No Starch Press, 2010.

Reference Material
The C Programming Language; Brian Kernighan and Dennis Ritchie, Prentice-Hall, 1988. *Algorithms in C: Fundamentals, Data Structures, Sorting, Searching*; Robert Sedgewick, Addison-Wesley, 1997. *Advanced Programming in the UNIX Environment*; W. Richard Stevens, Addison-Wesley, 1993

Major Topics Covered in the Course

1. Program development and object code structure
2. UNIX API introduction
3. UNIX file systems and I/O
4. Concurrency concepts and concurrent programming
5. Process creation, management and inter-process communication
6. Software interrupts
7. Multithreading and thread synchronization
8. Introduction to networking and Unix socket programming

Projects

Programming Project Area	Weeks
Low-level file I/O programming	2
Low-level file system directory manipulation	1
Process creation, termination and communication using pipes	2
Process synchronization using software interrupts	1
Inter-process communication mechanisms & deadlock avoidance	2
Multithread programming and synchronization	3
Client/server programming using Berkeley sockets	3

Design, Implementation and Analysis

The instructor performs analysis of representative problems in class. Student programming assignments require the student to analyze problem requirements and incorporate them into their solutions.

This course requires the student to craft 7-9 correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend and apply knowledge and concepts from particular areas of the Unix API and from lectures and apply them to the design of their program. These programs range from 100-200 lines of code at the beginning of the semester, to a final project whose design usually entails >1000 lines of program code, incorporating dozens of subprograms in multiple executables. The instructor will devotes laboratory time to guiding students in their program design efforts.

CS2013

This course provides coverage of CS2013 knowledge areas. Values listed are minimum course hours dedicated to the topic, percentages indicate the fraction of CS2013 knowledge area topics covered (acceptable values are: <25%, 25-75%, >75%, or 100%).

Area	Tier 1	Tier 2	Elective
NC/Introduction	2 (>75%)		
NC/Networked Applications	3 (25-75%)		
OS/Overview of Operating Systems	2 (25-75%)		

Area	Tier 1	Tier 2	Elective
OS/Operating System Principles	9 (25-75%)		
OS/Concurrency		3 (25-75%)	
OS/File Systems			3 (<25%)
PD/Parallelism Fundamentals	6 (<25%)		
PD/Parallel Decomposition	2 (>75%)	2 (<25%)	
SF/Parallelism	6 (<25%)		

Course Coordinator:	Ben McCamish
Last Updated:	January 24, 2022
Syllabus Version Number:	3.2