

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 223 – Advanced Data Structures

CS 224 – Programming Tools

CS 261 – C and Assembly Language Programming

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. Design and implement programs making direct use of operating system facilities to perform low-level file I/O and directory manipulations. (*Contributes to performance criterion K-3*)
2. Use a modern OS API and the concepts of process creation, synchronization & communication to solve concurrent programming problems. (*Contributes to performance criteria I-5*)
3. Exhibit Proficiency in the use of the C programming language to design and code systems-related programs (*Contributes to performance criterion I-1*)

Note: Students will be directly assessed in their knowledge of the C programming language and their ability to apply it to the solution of programming problems (*Contributes to performance criterion I-1*).

Required Textbooks

Unix Systems Programming, Second Edition; Haviland, Gray & Salama; Addison-Wesley, 1999.

The Unix Programming Environment; Kernighan & Pike; Prentice-Hall, 1984.

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. Process management and inter-process communication
5. Concurrency concepts and concurrent programming
6. Introduction to networking and Unix socket programming

Laboratory 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 with software interrupts	1
Inter-process communication mechanisms & deadlock avoidance	2
Client/server programming using Berkeley sockets	3

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	0	1	Computer Organization and Architecture	0	1
Algorithm & Software Design	0	2	Concepts of Programming Languages	0	0

Oral and Written Communications

There are no significant oral or written communications required in this course. Virtually all assignments consist of writing computer programs.

Social and Ethical Issues

This course contains no significant coverage of social and ethical issues beyond the usual proscriptions against plagiarism and cheating.

Theoretical Content

Topic	Hours
Deadlock, liveness and starvation	1
Synchronization models and mechanisms	3

Problem Analysis

The instructor performs analysis of representative problems in class. All student programming assignments require the student to analyze problem requirements. The instructor analyzes problem solutions (both his own and student's) in class.

Solution Design

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. The instructor will devote laboratory time to guiding students in their program design efforts.

CC2001

This course provides coverage of topics in the following areas (hours listed are minimums):

OS1. Overview of operating systems [core]	2
OS2. Operating systems principles [core]	2
OS3. Concurrency [core]	6
OS8. File systems [elective]	6
NC1. Introduction to net-centric computing [core]	2
NC5. Building web* applications [elective]	6
SE2. Using APIs [core]	5

* → web is taken to be synonymous with network in this course

Course Coordinator: Dick Lang
Last Updated: April 15, 2009 (Approved)
Syllabus Version Number: 2.0