

## CSCE 510: SYSTEM PROGRAMMING

### Catalog Description:

**510—System Programming.** (3) (Prereq: CSCE 245) System software such as command language interpreters, client-server applications, debuggers, mail systems, browsers, macro-processors, and revision control systems; file systems, processes, threads, and interprocess communication.

### Prerequisite(s) By Topic:

Introductory programming and data structures  
Logical operations

### Textbook(s) and Other Required Material:

*Linux Application Development*, Johnson and Troan, Addison-Wesley, 1998.

A C or C++ reference book

#### Alternative Textbooks:

*Advanced Programming in the Unix Environment*, W. Richard Stevens, Addison-Wesley, 1992

*Unix Systems Programming*, 2nd Edition, K. F. Haviland and B. Salama, Addison-Wesley, Reading, Massachusetts, 1997.

### Computing Platform: SUN/Solaris

### Course Objectives: {Assessment Methods Shown in Braces}

1. Demonstrate mastery of the internal operation of Unix system software including assemblers, loaders, macro-processors, command language interpreters, inter-process communication. {tests}
2. Develop medium to large C/C++ programs in a Unix Environment utilizing the C preprocessor, the debugger (gdb), make, source code revision systems (sccs), etc. {programming assignments, tests}

### Topics Covered:

1. Introduction to systems software. (1)
2. File Systems: directory structures, files, the stat system call, system data files (7)
3. Processes: arguments, environments, memory allocation, creation, threads (7)
4. Interprocess communication: signals, pipes, semaphores, shared memory, sockets (12)
5. System Software Implementation including: macro-processors, editors, terminal handling, software development tools: make, sccs, debuggers, library management: loaders, archivers, tar, compression software, command language interpreters: shells, mail systems: pine, client-server applications: browsers. (12)

### Laboratory Projects:

Four to five fairly extensive programming projects.

**Syllabus Flexibility:** Medium. The Undergraduate Committee approves the choice of textbook and syllabus.

**Relationship of Course to Program Outcomes:**

The contribution of each course objective to meeting the program outcomes is indicated with the scale:

3 = major contributor, 2 = moderate contributor, 1 = minor contributor. Blank if not related.

Course Objectives	Program Outcomes										
	1. Logic & Math	2. Computing Fundamentals	3. Apply Computing Principles	4. Work on teams	5. Communicate Effectively	6. Liberal arts & Soc. Sciences	7. Basic Science and Lab Procedures	8. Learn New Tools & Processes	9. Employed upon Graduation	10. Application Area	11. Electronics and Digital Sys Design
1. Demonstrate mastery Unix System Software		1	2					2	2		
2. Develop medium to large C/C++ programs in a Unix Environment			3	1				3	2		

**Estimated Computing Category Content (Semester hours):**

Area	Core	Advanced	Area	Core	Advanced
Algorithms			Data Structures		
Software Design		2	Programming Languages		
Computer Architecture		1			

**Estimated Information Systems Category Content (Semester hours):**

Area	Core	Advanced	Area	Core	Advanced
Hardware and Software		1	Networking and Telecommunications		
Modern Programming Language			Analysis and Design		2
Data Management			Role of IS in an Organization		
Quantitative Analysis			Information Systems Environment		

**Oral and Written Communication: None**

**Social and Ethical Issues: None**

**Theoretical Content:**

Some concurrency, semaphores

**Analysis and Design:**

Development of several large programs

**Class/Laboratory Schedule:**

Lecture: 3 periods of 50 minutes or 2 periods of 75 minutes per week

**Modification and Approval History**

Initial description November 2000

Modified June 2005 by Caroline Eastman to modify format