Rutgers University, Department of Electrical and Computer Engineering
ABET COURSE SYLLABUS
COURSE: 14:332:456

Course Catalog Description: 14:332:456 Network-Centric Programming (3)
Advanced programming with a focus on developing software for networked systems using Linux as a reference platform. Topics: Programming Tools, Software Design, Programming Techniques, Environment of a UNIX Process, Memory Allocation, Garbage Collection, Process Control, Process Relationships, Signals, Reliable Signals, Threads, I/O Multiplexing, Datagram and Stream Sockets, Multicasting, Device Driver and Kernel Programming, Secure Programming.

Pre-Requisite Courses: 14:332:351

Co-Requisite Courses: 01:198:416

Pre-Requisite by Topic:

1. C/C++ Programming Methodology
2. Data Structures and Algorithms
3. Familiarity with Operating Systems

Textbook & Materials: W. R. Stevens, B. Fenner, A. M. Rudoff, *Unix Network Programming, Vol. 1: The Sockets Networking API*, 3rd Ed., Addison-Wesley, 2003.

References: W. R. Stevens and S. Rago, *Advanced Programming in the UNIX(R) Environment*, 2nd Ed., Addison-Wesley, 2005.

Overall Educational Objective: To introduce students to the development of network software using Linux as a reference platform, including a basic understanding of development at the kernel level. To create a foundation for further study and professional practice in software development.

Course Learning Outcomes: A student who successfully fulfills the course requirements will have demonstrated:

1. An ability to develop both connection-oriented and connectionless network programs, define the difference between them, and to choose the appropriate primitive for different applications requirements.

2. An ability to understand the performance characteristics and implement both incremental and concurrent network servers using threads or processes.

3. An ability to understand the essence of security exploits. Preventing, discovering, and correcting security weaknesses in network software, in particular access control, buffer overflow, and SQL insertion.

4. An understanding of the Linux IO Multiplexing, process address space organization, and development tools.

5. The ability to understand and implement the key elements of the World Wide Web, the HTTP protocols and web servers.

How Course Outcomes are Assessed:
Homeworks (40 %)
Class Discussions (10%)
Two Mid-Term Exams (20 %)
Final Exam (30 %)

N = none S = Supportive H = highly related

Outcome	Level	Proficiency assessed by
(a) an ability to apply knowledge of Mathematics, science, and engineering	H	HW Problems, Exams
(b) an ability to design and conduct experiments and interpret data	S	Design problems in HW and Exams
(c) an ability to design a system, component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability	N	
(d) an ability to function as part of a multi-disciplinary team	N	
(e) an ability to identify, formulate, and solve ECE problems	H	HW Problems, Exams
(f) an understanding of professional and ethical responsibility	N	
(g) an ability to communicate in written and oral form	S	HW Problems, Class Discussions
(h) the broad education necessary to understand the impact of electrical and computer engineering solutions in a global, economic, environmental, and societal context	N	
(i) a recognition of the need for, and an ability to engage in life-long learning	S	Lectures, subsequent courses
(j) a knowledge of contemporary issues	S	HW Problems, Exams
(k) an ability to use the techniques, skills, and modern engineering tools necessary for electrical and computer engineering practice	H	HW (including problem solution simulation and design)
Basic disciplines in Electrical Engineering	N	
Depth in Electrical Engineering	S	HW, Quizzes, Exams
Basic disciplines in Computer Engineering	H	HW, Exams
Depth in Computer Engineering	H	HW, Exams
Laboratory equipment and software tools	H	Linux, GNU C Development Env.
Variety of instruction formats	S	Lectures, Laboratory problem sessions, Office hour discussions

Topics Covered week by week:

Week 1: Programming Tools: Static and Dynamic Libraries, Build Systems, Configuration Management, and Documentation Tools, Debuggers

Week 2: Advanced Programming Techniques: Programming Style, Modularity, Design Patterns, Debugging Techniques

Week 3: Process Address Space and Environment, Dynamic Memory Allocation, Garbage Collection

Week 4: Concurrency: Process Control and Relationships, Signals, Threads

Week 5: Concurrency: Synchronization primitives, hourly exam 1

Weeks 6 and 7: Sockets Network Programming: Stream and Datagram sockets, Server Design, I/O Multiplexing, Multicasting

Week 8: Web services: Remote procedure calls, Web servers, HTTP, XML, XSLT, SOAP, WSDL

Weeks 9 and 10: Kernel Development: Kernel Modules, Allocating Memory, Timers, Debugging Techniques, Kernel Network Stack, Socket Buffers, Netfilter

Weeks 11 and 12: Device Drivers: Interrupt Handling, Char, Block, and Network Drivers

Weeks 13 and 14: Secure Programming: Access Control, Buffer Overflow, Covert Channels, Secure Sockets

Weeks 15 and 16: Review and Final Exam

Computer Usage: Use of Linux in Homework assignments and exams.

Laboratory Experiences: All homeworks and exams require use of the computer laboratory.

Design Experiences: ~80% Homework problems are design-oriented problems, which require students to design and implement computer programs that meet specified requirements. ~80% problems in the Exams are design related.

Independent Learning Experiences: 1. Computer Programming Home-Work, 2.Testing (Quizzes, Exams)

Contribution to the Professional Component:
(a) College-level mathematics and basic sciences: 0 credit hours
(b) Engineering Topics (Science and/or Design): 3 credit hours
(c) General Education: 0 credit hours
Total credits: 3
Prepared by: M. GRUTESER
Date: July 2007