CSCI 5340 - PRINCIPLES OF SYSTEMS PROGRAMMING

CREDIT HOURS: 3
PREREQUISITES: CSCI 3323 or 3342
GRADE REMINDER: Must have a grade of C or better in each prerequisite course.

CATALOG DESCRIPTION

Operating systems principles, systems utilities, language processors, and user interfaces.

PURPOSE OF COURSE

The purpose of this course is to enable the student to develop an understanding of the integral role played by systems software in a computing system. The components of a software system are studied along with the interactions between software, hardware, and the user. Emphasis is placed on the impact at the systems software level of user-level requests for system services.

EDUCATIONAL OBJECTIVES

Upon successful completion of the course, students should be able to:

1. Distinguish between machine-dependent and machine-independent features of system software.
2. Demonstrate a thorough understanding of the machine language instruction execution process.
3. Design and implement a translator and pre-processor for a macro-assembly language.
4. Demonstrate familiarity with high-level programming language syntax specification schemes, and to use these in the implementation of a parser.
5. Present an overview of other system software functions, such as process management, memory management, and file management.

COURSE CALENDAR

This course meets for a minimum of 37.5 lecture contact hours during the semester. Students have significant assignments based on readings from the primary literature, participate in classroom discussions regarding current research topics, complete periodic homework and laboratory/programming assignments, and periodic exams in addition to the final exam. Students are expected to prepare for any class assignments or quizzes over the material covered in class or in the reading material. Successful completion of these activities requires at a minimum six additional hours of outside of classroom work each week.

CONTENT **Hours**

Machine Structure and Functional Architecture 5
- Machine instruction formats
- Design of an architecture and development of software emulator

Assemblers 8
- Assembly language formats, addressing modes, two-pass assemblers, symbol tables, load-and-go assemblers, modular assemblers, assembler directives, error handling

Linkers and Loaders 5
- Binary loaders, linking loaders
- Linkage editors, relocation, overlay handling
- Dynamic linking and loading

Macro Processors 5
- Macro definition and expansion
- Nested calls and nested definitions
- Two-pass processors, multi-level recursive processors

Language Processors 8
- Language specification
- Compilers and Interpreters
- Parsing techniques
- Compiler development tools

Components of Operating Systems 11
- Process management, concurrency, synchronization
- Resource scheduling
- Memory management techniques, paging, virtual memory
- File Systems
- Distributed systems

Exams (plus final) 3

TOTAL 45

REFERENCES

Andrews, Gregory R., Foundations of Multithreaded, Parallel, and Distributed Programming, Addison-Wesley, 2000.

Beck, and Leland, System Software: An Introduction to Systems Programming, 3rd Ed., Addison-Wesley, 1997.

Silberschatz, A. and Galvin, P., Operating Systems Concepts, 5th Ed., Addison-Wesley, 1994.