

CSC 441 - PRINCIPLES OF SYSTEMS PROGRAMMING

CREDIT HOURS: 3
PREREQUISITES: CSC 323 or 342 or 343
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.

NOTE: Graduate students taking CSC 441 for graduate credit will be expected to complete additional requirements, including but not limited to special projects, class presentations, relevant research, and supplemental evaluation (i.e., additional questions, quizzes, tests). Graduate students are expected to perform at a higher level than undergraduates. Students should contact the course instructor early in the semester (i.e., before the end of the add/drop period) to determine the specific additional requirements.

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.

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.