CSE 3903: Project: Design, Development, and Documentation of System Software

Course Description

Intensive group project involving design, development, and documentation of system software including an assembler and a linking loader; communication skills emphasized; builds programming maturity.

Prior Course Number: 560
Transcript Abbreviation: Proj: Sys Software
Grading Plan: Letter Grade
Course Deliveries: Classroom
Course Levels: Undergrad
Student Ranks: Junior
Course Offerings: Autumn, Spring
Flex Scheduled Course: Never
Course Frequency: Every Year
Course Length: 14 Week
Credits: 4.0
Repeatable: No
Time Distribution: 3.0 hr Lec
Expected out-of-class hours per week: 9.0
Graded Component: Lecture
Credit by Examination: No
Admission Condition: No
Off Campus: Never
Campus Locations: Columbus
Prerequisites and Co-requisites: Pre-req: 2231; and 2321; and 2421 or 3430, or 2451 and ECE2560
Exclusions: Not open to students with credit for 560.
Cross-Listings:

Course Rationale: Existing course.

The course is required for this unit's degrees, majors, and/or minors: Yes
The course is a GEC: No
The course is an elective (for this or other units) or is a service course for other units: Yes

Subject/CIP Code: 14.0901
Subsidy Level: Baccalaureate Course

Programs

Abbreviation	Description
BS CSE	BS Computer Science and Engineering

Course Goals

Be competent with using and implementing each component of the the assemble-link-load-relocate-execute process
Be competent with using bit manipulation of integers and ascii characters to be able to emulate a simple computer that handles both integer and character I/O
Be competent with writing, organizational, and presentation skills
Be competent with analyzing the intended audience for a written document and writing an audience profile

Be familiar with making engineering decisions involving tradeoffs
Be familiar with writing a relocating linking loader
Be familiar with subroutine linkage at the assembly level and with using different addressing modes
Be familiar with emulating in software, the fetch-decode-execute cycle of a CPU
Be familiar with using macros, including recursive and nested macros
Be familiar with software testing strategies including black-box versus white-box, unit testing, integration testing, top-down versus bottom-up testing, and construction and implementation of a test plan
Be familiar with defining the purpose (persuade, inform, etc.) of a written document and select the appropriate rhetorical devices.
Be familiar with writing several pieces of documentation that have different purposes and to use appropriate organization to tie them together.
Be familiar with group project organization techniques including conducting group meetings, recording minutes, and tracking project progress.
Be familiar with using one structured approach to large software design to carry out a large group project.
Be exposed to issues in systems programming as opposed to applications programming
Be exposed to memory management issues including caching and virtual memory
Be exposed to one-pass macro processing techniques

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Architecture	4.0							
Software engineering	4.0							
Technical writing	3.0							
System software	2.0							
Assemblers: algorithm, pseudo operations, expressions	8.0							
Searching and sorting	3.0							
Tools: makefiles, CVS, lex and yacc	4.0							
Linking and loading	4.0							
Macro processors	4.0							
Compilers: tokenizing, parsing, code generation	4.0							

Representative Assignments

Basic absolute loader and simulator
Assembler
Relocating, direct linking loader (and modified assembler)

Grades

Aspect	Percent
Lab assignments	55%
Midterm	20%
Exam	25%

Representative Textbooks and Other Course Materials

Title	Author
System Software - An Introduction to Systems Programming	Leland Beck
The Practice of Programming	Brian Kernighan and Rob Pike
A Writer's Reference	Diana Hacker

ABET-EAC Criterion 3 Outcomes

Course Contribution		College Outcome
**	a	An ability to apply knowledge of mathematics, science, and engineering.
	b	An ability to design and conduct experiments, as well as to analyze and interpret data.
***	c	An ability to design a system, component, or process to meet desired needs.
***	d	An ability to function on multi-disciplinary teams.
**	e	An ability to identify, formulate, and solve engineering problems.
*	f	An understanding of professional and ethical responsibility.
**	g	An ability to communicate effectively.
	h	The broad education necessary to understand the impact of engineering solutions in a global and societal context.
*	i	A recognition of the need for, and an ability to engage in life-long learning.
	j	A knowledge of contemporary issues.
***	k	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

BS CSE Program Outcomes

Course Contribution		Program Outcome
**	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
***	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
***	d	an ability to function on multi-disciplinary teams;
**	e	an ability to identify, formulate, and solve engineering problems;
*	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
**	g	an ability to communicate effectively with a range of audiences;
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
*	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
	j	a knowledge of contemporary issues;
***	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
*	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
*	m	an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
**	n	an ability to apply design and development principles in the construction of software systems of varying complexity.

Prepared by: Kathryn Reeves