

CSC 4304: Systems Programming

Credit Hours: 3 hours

Prerequisites:

CSC 4103

Prerequisites By Topic:

Basic programming knowledge in C/C++. Basic constructs of programming: e.g. data types, control structures, modular design and implementation, and algorithm design techniques. Computer Science concepts: e.g. operating systems, file and I/O structures, networking, memory structures, processes and threads.

Catalog Course Description:

Batch process system programs, their components, operating characteristics, user services and limitations; implementation techniques for parallel, distributed and concurrent processing; interrupt handling; addressing techniques, file system design and management, system accounting, and other user-related services; traffic control, interprocess communication, remote procedure calls, design of system modules, and interfaces; system updating, documentation, and operation.

Course Outcomes

1. Master in using the C/C++ programming language, its constructs and grammar, to create system software.
2. Master in the usage of makefiles, linking, object files, loading, symbol resolution, shared and static libraries, debugging, and execution of system programs.
3. Be familiar with basic UNIX OS concepts such as: process, program, process groups, signals, running programs, process control, address space, user and kernel modes, system calls, and context switching.
4. Master in file I/O (i.e. open, close, read, write, seek)
5. Be familiar with using sockets to implement client-server environment.
6. Be familiar with using thread execution models (e.g. Posix threads).
7. Be familiar to handle signals and exceptions within a process and to control processes.
8. Be familiar with different approaches of concurrent programming.
9. Be familiar with different batch processing systems.
10. Be familiar with remote execution techniques.

Texts and Other Course Materials

1. Computer Systems: A Programmer's Perspective, by Bryant and O'Hallaron. Prentice Hall. 2002.
2. The C Programming Language. By Kernighan and Ritchie. Prentice Hall. Latest Edition.

References:

1. Advanced Programming in the UNIX Environment, by Richard Stevens. Addison Wesley Press, 1992, ISBN 0-201-56317-7.
2. UNIX SYSTEMS Programming: Communication, Concurrency and Threads by Kay A. Robbins and Steven Robbins, Prentice Hall ISBN 0-13-042411-0

Major Topics

- Describe the process of writing system software from code writing to compilation and linking,
- Brief introduction to the language C,
- Exceptions, interrupts, and systems calls,
- Processes and process control,
- Inter-process communication
- Signal and interrupt handling
- Remote procedure calls
- Impact of pipelining on program performance,
- File I/O and system calls,
- Performance measurement and improvement including discussion of parallel architectures,
- Structure and usage of storage technologies including the memory hierarchy,
- Reading and writing floating-point numbers,
- Multi-precision arithmetic,
- Caches and ways to improve cache performance,
- Virtual memory,
- Memory allocation, protection, and garbage collection,
- The client-server programming model and its application to networks and the global IP Internet,
- OS timing facilities,
- Optimizing of C programs using analysis of the generated machine code,
- Threads and concurrent programming
- Batch processing systems
- Distributed computing

Assignments/Projects/Laboratory Projects/Homeworks

Implementation of client/server model
Implementation of multi-threaded processes
Implementation of remote procedure calls

Implementation of inter-process communication, signal and interrupt handling

Implementation of file I/O optimization

Usage of batch processing systems and distributed computing

Debugging complex applications

Curriculum Category Content (estimated in semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	8	5	Data Structures	10	
Software Design	5		Prog. Languages	5	10
Computer Architecture	2		Mathematical fundamentals		

Relationship to Criterion 3 Outcomes

A	B	C	D	E	F	G	H	I	J	K
*	*	*		*		*	*	*	*	*

- *Math and Fundamentals*
- *Data Structures*
Stacks, queues, linked list, searching, sorting, tree structures, graphs (10 hours)
- *Algorithms and Software*
Algorithm analysis- Parallel algorithm design, concurrency, search and sort algorithms, threads, multiprocessing, distributed computing, memory allocation, garbage collection, file I/O system calls, signal interrupt handling, remote procedure calls, interprocess communication, pipelining, debugging, performance evaluation, caching, batch processing (18 hours)
Software Design – design and implement system software, guidelines for larger software systems development (5 hours)
- *Computer Organization and Architecture:*
Overview of basic computer organization and basic computer architecture structures (2 hours)
- *Concepts of Programming Languages*
Master C or C++ programming using the concepts of address pointers, list structures, tree structures, memory management, thread programming, network programming, introduction to programming with sockets, client server programming, file systems and I/O interfaces, UNIX shell (ex. Bash) programming, scripting, basics of UNIX system administration, (15 hours)
- *Social and Ethical Issues:*

- *Oral Communication (presentations)*
- *Written Communication:*

Course Coordinator: Dr. Tefvik Kosar
Last Modified: May 16, 2007