Princeton University
COS 217: Introduction to Programming Systems
Spring 2004 Final Exam Preparation

Topics

*You are responsible for all material covered in lectures, precepts, and required readings. New topics are in **boldface**.*

1. C programming

    The program preparation process
    Memory layout
    Data types
    Operators
    Statements
    Function declarations and definitions
    Pointers
    Arrays
    Command-line arguments
    Constants
    Text files
    Structures
    Dynamic memory management
    Void pointers
    Function pointers
    Variable declarations and definitions
    Variable scope, linkage, and duration
    Macros and their dangers
    The assert macro

2. Programming style

    Modularity, interfaces, implementations
    Multi-file programs using header files
    Opaque pointers
    Abstract data types
    Abstract objects
    Testing strategies
    Profiling and instrumentation
    Performance tuning
    Robust programming, error handling strategies
    Exception handling
    Portable programming

3. IA-32 architecture and assembly language

- **Registers vs. memory**
- **Assembly language**
  - **Directives (.section, .asciz, .long, etc.)**
  - **Instructions/mnemonics (movl, addl, call, etc.)**
  - **Condition codes and conditional branch instructions**
  - **Instruction operands**
    - **Immediate operands**
    - **Register operands**
    - **Memory operands**
  - **The stack and local variables**
  - **The stack and function calls**
- **Number representation**
  - **The binary, octal, and hexadecimal number systems**
  - **Signed numbers**
    - **Signed magnitude, one's complement, two's complement**
  - **Floating point numbers**
- **Assemblers**
  - **The forward reference problem**
  - **Pass 1: Create symbol table, relocation records, other sections**
  - **Pass 2: Use relocation records and symbol table to partially patch code in other sections**
- **Linkers**
  - **Resolution**
    - **Fetch library code**
    - **Enhance symbol table**
  - **Relocation**
    - **Use relocation records and symbol table to completely patch code in other sections**
- **Inline assembly language within C programs**

4. Operating systems

- **History and overview**
- **UNIX shells**
  - **Shell built-in commands vs. executable binary commands**
- **Processes**
  - **Scheduling, context switching**
  - **UNIX system calls: getpid, execvp, fork, wait, kill, chdir, setenv, unsetenv**
  - **Standard C functions: exit, atexit, getenv**
- **I/O**
  - **UNIX file descriptors**
  - **UNIX file redirection**
  - **Inter-process communication via pipes**

Inter-process communication via sockets
Standard C functions: fopen, fclose, **fflush**, **perror**, fgetc, fputc, fgets, fputs, fscanf, fprintf, scanf, printf, getc, putc, putchar, getchar, gets, puts, etc.
UNIX system calls: creat, open, close, dup, read, write, pipe

Signals
UNIX kill command
Standard C function: signal
UNIX system calls: sigaction, alarm, setitimer

5. Applications

“De-commenting” and lexical analysis via finite state automata
String manipulation
Symbol tables and hash tables
Execution profilers
UNIX shells

6. Tools: The UNIX/GNU programming environment

UNIX, bash, xemacs, gcc, gdb, **make**, **gprof**

Readings

As specified on the course web pages...

Required:

C Programming (King): 1-15, 16.1-3, 17-19, **24.3**

The Practice of Programming (Kernighan & Pike): 1, 2, 4, 5, 6, **7**, **8**

***IA32 Intel Architecture Software Developer's Manual, (Volume 1: Basic Architecture)*: 4**

***Programming from the Ground Up* (Bartlett): 1, 2, 3, 4, 9, 10, B, E, F**

***The UNIX Programming Environment* (Kernighan & Pike): 7.4-5**

Recommended:

Programming with GNU Software (Loukides & Oram): 2, 3, 4, 6, **7**, **9**

The C Programming Language (Kernighan & Ritchie): 1, 4.11, 5

C Interfaces and Implementations (Hanson): 3.2

Using as, the GNU Assembler

IA32 Intel Architecture Software Developer's Manual, Volume 1: Basic Architecture**: 2.1, 3, 5**

Programming from the Ground Up **(Bartlett): 5, 6, 7, 8, 11, 12, 13, C**

Executable and Linkable Format

The UNIX Programming Environment **(Kernighan & Pike): 1, 2, 3, 4, 5, 7.1-3**