

Final Exam Review

T. Howell

5-13-08

class28.ppt

Topics

Outline of topics for final exam

1. Representing Information
2. Integer Arithmetic
3. Floating Point Formats
4. Assembly Language Instructions
5. Procedures
6. Arrays
7. Structures/Unions
8. Y86 instructions
9. Logic Design

- 2 -

CS 47 Spring '08

Topics (cont.)

10. SEQ Processor
11. Linking/Loading
12. Relocation
13. Exceptional Control Flow
14. Storage Technology
15. Cache Memories
16. Virtual Memory
17. Measuring Execution Time
18. Performance Optimization

- 3 -

CS 47 Spring '08

1. Representing Information

Everything is bits: 1's and 0's

Char

Int

Unsigned int

Float

Double

Pointer

Strings, Arrays, Structures, Unions

Code

- 4 -

CS 47 Spring '08

2. Integer Arithmetic

Signed and unsigned

Char, word, int, long long int

Umin, Umax

Tmin, Tmax

Commutative, Associative, Distributive

Overflow destroys ordering: $T_{max} + 1 = T_{min}$

Results are correct modulo 2^w

- 5 -

CS 47 Spring '08

3. Floating Point Formats

Float, Double, Extended precision

Sign, Biased Exponent, Fraction

Exponent = biased exponent - bias

Significand = 1 + fraction (normalized)

Significand = 0 + fraction (denormalized)

Special codes for infinity, NaN

Commutative

Ordering works

Rounding ruins associativity, distributivity

- 6 -

CS 47 Spring '08

4. Assembly Language Instructions

mov, push, pop

call, ret, jmp (jnz, je, jg, etc.)

lea, add, sub, inc, dec, and, or, xor, imul, sal, sar, ...

cmp, test, set

Translation between C and assembly

Use of registers %ebp, %esp to manage the stack

- 7 -

CS 47 Spring '08

5. Procedures

Stack frames

push arguments

call pushes return address

save and restore registers (callee save vs caller save)

allocate space for local variables

ret pops return address

Study Problem 1 from Midterm 2.

- 8 -

CS 47 Spring '08

6. Arrays

One dimensional or Nested

Addressing elements by index: $t = A[i]$;

```
movl (%edx, %ecx, 4), %eax
```

Addressing elements with pointers: $t = *(A + i)$;

```
leal (%edx, %ecx, 4), %ebx
```

```
movl (%ebx), %eax
```

7. Structures/Unions

Similar to arrays, but with diverse element types

Union elements share storage

Know how to map the storage for structures and unions

Alignment rules depend on element types

8. Y86 instructions

Summarized on Midterm 2 handout

Simplified IA32 instruction set

Consistent format: [code:fn] [rA:rB] [disp or data]

- 11 -

CS 47 Spring '08

9. Logic Design

HCL = Hardware Control Language

describe logic gates with C-like code

bit-level with Boolean

word-level with integer

Combinational circuits do not store information

Sequential circuits use clocked registers to store state information

- 12 -

CS 47 Spring '08

10. SEQ Processor

Divide each instruction into six stages:

fetch, decode, execute,
memory, write back, PC update

Same stages for all instructions

Table for each instruction gives its actions at each stage

Resources: memory, PC incrementer,
register file, ALU, CC

- 13 -

CS 47 Spring '08

11. Linking/Loading

Linker performs

symbol resolution
relocation

Inputs are relocatable object files (.o)

Symbols can be *strong* or *weak*

1. multiple strong symbols not allowed
2. strong symbol definition overrides weak
3. linker may choose arbitrarily among weak symbols

- 14 -

CS 47 Spring '08

12. Relocation

Compiler produces relocation entries of various types

**R_386_PC32 (DISP32) is for PC-relative addressing
(used for procedures, labels)**

**R_386_32 (dir32) is for absolute addresses encoded
in the instruction.**

(used for global constants, strings)

**Linker assigns addresses to relocation entries when it
has determined locations of code and data
segments.**

- 15 -

CS 47 Spring '08

13. Exceptional Control Flow

Exceptions include

interrupt from I/O device

trap from program execution

fault from possibly recoverable error condition

abort from nonrecoverable error condition

Control passes to interrupt handler

Control may or may not return to interrupted program

**Exceptions are similar to procedure calls but are
asynchronous**

- 16 -

CS 47 Spring '08

14. Storage Technology

SRAM (cache memory)

DRAM (main memory)

ROM, EEPROM (flash) are nonvolatile

Disk storage

platters, surfaces, tracks, sectors, bytes

seek, rotational latency, transfer time

- 17 -

CS 47 Spring '08

15. Cache Memories

Small, fast, expensive storage serves as cache for large, slow, cheap storage in a memory hierarchy

Depends on locality of reference

Cache is addressed by {set index: tag: block offset}

set index comes from middle of physical address

tag is high order bits, offset is low order bits

Study example from Class 23, Practice Problems 6.9, 6.10

- 18 -

CS 47 Spring '08

16. Virtual Memory

One large virtual address space per process

All share a smaller physical address space

Pages (4k bytes) are read from disk to memory as needed. Memory serves as a cache for disk.

Page tables indicate physical locations of virtual pages.

Translation look-aside buffer (TLB) is a cache for page table entries inside the CPU.

VM simplifies memory management and memory protection. All memory accesses go through page tables.

- 19 -

CS 47 Spring '08

17. Measuring Execution Time

Tools for measuring execution time:

interval counting

cycle counter

time-of-day clock

Cycle counting has variability from overhead and cache effects.

Better results by warming the cache and using K-Best measurement scheme. Report the best time that can be repeated K times out of N within a factor $1+\epsilon$.

- 20 -

CS 47 Spring '08

18. Performance Optimization

Optimizing compilers have limitations

Programmers can

- eliminate loop inefficiencies**

- reduce procedure calls**

- eliminate unneeded memory references**

- reduce loop overhead**

- use pointer code (when appropriate)**

- identify bottlenecks by profiling**

Examples in Class 26 notes.