CS61C: Review, ISAs & Alternative Ideas

CS61C Fall2007 - Discussion #4

Greg Gibeling

Buffer Overflows

- A Fixed Example

```
void foo(char* string) {
  int length = strlen(string);
  char* buffer =
  (char*)malloc((length+1)*sizeof(char));
  strncpy(buffer, string, length);
  buffer[length] = '\0';
  // etc…
}
```


Quiz6

- Max & Min `int` on nova.cs.berkeley.edu
  - 2147483647, -2147483648
    - (2^31)-1 and -(2^31)
  - Nova is a 32bit machine
    - C int and unsigned int datatypes will be 32 bits
    - `printf("%u\n", sizeof(int));`
- Convert the unsigned binary value 10110010 to decimal
  - 10110010 = 178
- Convert the signed (twos complement) binary value 10110010 to decimal
  - 10110010 = -78
  - Signed will always means "twos complement" unless otherwise specified
  - Other answers
    - 50: sign magnitude
    - -178: who knows?
    - 178: thought for some reason it was a 32bit value (Why? I don't know…)

Quiz7

- Lines of code
  - Order the three languages from most to least
    - Java: very dense, thanks to extensive libs & language support
    - C: has libraries, but little language support
    - MIPS Assembly: generally no libraries, certainly no language support, simple commands
  - Question hints at the power of abstraction
- Assemble `k++`, where `k` is in `$s1`
  - `addiu?$s1.?$s1.?[0-9]*[1-9][0-9]*`
  - What's the constant added to k?

Quiz8

- Two instructions for "branch on less than"
  - `slti?,b(ne|eq)`
    - These are the four ways I know of to do this
    - Anyone know any more?
  - Why isn't it in MIPS?
- Load `0xDEADBEEF` into `$s0`
  - `lui $s0, 0xDEAD; ori $s0, $s0, 0xBEEF`
    - Any other ways to do this?
  - Why in two halves? Why not in one instruction?

Notes From Greg

- Goto Statements
  - In MIPS/x86/Assembly: necessary
  - In C: I will not help with your code
- Register Coloring
  - Remember the 4-Color Theorem?
  - Deciding which variables go in which registers is like that…
  - See CS164 (seriously)
- C to Java: A rough transition
  - "." in Java becomes "`->`" in C
  - Keep track of whether you have a pointer or a block of memory
    - E.g. strings are all in your imagination in C
    - Can someone tell me how to free a string?
- All assignment are interrelated!
  - New parts of the class build on old ones (unlike most classes)

General Questions

- Current
  - Lab3
  - HW3
  - Proj1
- Future
  - Proj2: sprintf in MIPS
    - Should be out later today, I hope
  - Lab4: intro to MIPS
    - MARS: MIPS simulator & debugger
      - Requires X11/Exceed for GUI
      - `ssh -X cory`
    - Read it early, consider getting started early

Lab3 Vector Alternatives

- What we did
  - Explicitly store the size
  - NewSize = max(2*Size, loc+1)
  - Memset to zero out the new storage
  - Memcpy to copy other elements
- Other Options
  - Use EOF to mark end of the array
  - Ropes
  - Gapped circular array buffer

Minimal ISA (2)

- Design a minimal ISA
  - Break into groups of ~6
  - At least one person should take some notes
- ISA: Instruction Set Architecture
  - MIPS has 32 registers, 32b instructions, 3 operands per instruction, explicit load/store
  - What does your ISA consist of?
  - What instructions does it have?
  - How do you compute? How do you store data?

Minimal ISA (3)

- Single Instruction
  - sbn: subtract branch if negative
  - Build all operations up from this instruction
  - No need for registers, just use memory
  - "Universal Operator"
    - NAND, Mux, etc…
- Performance
  - CISC: Bad, RISC: Good, SBN: Very Bad
  - Why is the sweet spot in the middle at RISC?
  - So what?

Stump the TA

- Goal
  - A problem Greg can't solve
  - A question Greg can't answer
- Rules
  - No deliberate obfuscation
    - The problem/question may be complex
    - Your explanation of it must be as clear as possible
  - No detailed reference information
    - I'm not going to spend 20 minutes looking up Ann Margaret's pant size