ECS 341L – Fall 2007

Jedidiah Crandall

crandall@cs.unm.edu

Administrativia

- Go over the syllabus

The Goal of the Class

- Make you aware of the nitty-gritty details of the hardware that can affect your software.

Last value of i printed?

```
float f; int i = 0;

for (f = 0.0; f != 1.0; f += 0.1)
{
        printf("%d\n", i++);
}
```

Equivalent?

```
int i, j;
int sum;

for (i = 0; i < 5000000; i++)
      for (j = 0; j < 5000000; j++)
      {
            sum += A[i][j];
      }
```

```
int i, j;
int sum;

for (i = 0; i < 5000000; i++)
      for (j = 0; j < 5000000; j++)
      {
            sum += A[j][i];
      }
```

Themes

- Bits are only bits until they are interpreted
- Most of computer architecture is keeping up with process technology and the market – i.e. there is never a “right way” to do something (e.g. crack an egg)
- Sometimes the lines that have been drawn (RISC vs. CISC, scalar vs. superscalar, Von Neumann vs. Harvard etc.) are kind of fuzzy.

Why study assembly language and computer architecture?

- Sometimes-for reasons of performance, security, or otherwise-you just have to get at what's going on at the machine level, *e.g.*
  - Malware hiding using DMA virtualization technology
  - "Inside, the PlayStation 3 uses the seven core IBM designed Cell microprocessor as its CPU. While graphics processing is handled by the NVIDIA RSX." --Wikipedia

Times in my research when a background in architecture has helped

- Understanding buffer overflows and other memory corruption vulnerabilities.
- Worm network polymorphism.
- Reverse-engineering malware.
- Understanding the Cisco 12000 series of routers?

Code Red II

```
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3
%u7801%u9090%u9090%u8190%u00c3%u0003%u8b00
%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0\x0d\n.
```

MIPS vs. VAX

MIPS versus VAX

MIPS	VAX
Saving register	

swap: addi $29,$29, -12
sw $2, 0($29)
sw $15, 4($29)
sw $16, 8($29)
	swap: .word ^m<r0,r1,r2,r3>

Procedure body	

muli $2, $5,4
add $2, $4,$2
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
	movl r2, 4(a)
	movl r1, 8(a)
	movl r3, (r2)[r1]
	addl3 r0, #1,8(ap)
	movl (r2)[r1],(r2)[r0]
	movl (r2)[r0],r3

Restoring registers	

lw $2, 0($29)
lw $15, 4($29)
lw $16, 8($29)
addi $29,$29, 12	

Procedure return	
jr $31	ret

Other kinds of assembly

- Blackfin

```
A1 += R0.H * R1.H, A0 += R0.L * R1.L || R0 =
[I0++] || R1 = [I1++];
```

- REDCODE

```
0000: ADD.AB # 4, $ 3
0001: MOV.F $ 2, @ 2
0002: JMP.B $ -2, $ 0
0003: DAT.F # 0, # 0
```

Architecture is Getting Weird

- Corn with binary counters?
- “Cache” of encoded qubits in a quantum computer?
- ISA for nanotubes? Probably won’t be MIPS.

…but, architecture was already weird

- Dataflow architectures (80’s)
- Ternary machine Setun at Moscow State University in the 60’s
- Konrad Zuse’s Z1 with mechanical memory, 22-bit floating point with nine instructions and CPI ranging from 1 to 20 (1936)
- “While I'm digging in the tunnel, the elves will often come to me with solutions to my problem.” –Seymour Cray