Computer Systems Architecture
http://cs.nott.ac.uk/~txa/g51csa/

Thorsten Altenkirch

School of Computer Science and IT
University of Nottingham

Lecture 00: Introduction
What is this course about?

- Layers of abstraction
- MIPS programming

The University of Nottingham
What is this course about?

Layers of abstraction

MIPS programming
Why bother?

Joe Programmer, 2007

“Most software is produced using high level languages, methods and powerful tools. There is little need for low-level programming.”

High-level programming

- Haskell, Java, O’Caml, C#, Perl, Python, PHP, Ruby, …
- Extreme programming
- Eclipse, MS Visual Studio
Computer and processor types

![Bar graph showing the number of computers and servers from 1998 to 2002.](chart1)

![Stacked bar graph showing the number of processors from 1998 to 2002.](chart2)

The University of Nottingham
Layers of abstraction

**Abstractions are idealisations**
- Limited by low-level implementation
- Difficult to gauge or improve performance
  - with only an understanding of the abstract interface

**Embedded systems**
- Mobile phones, TV set-top boxes, games consoles
- Manufacturers provide firmware and software development kits...
- But who creates the infrastructure?
Overview

- Bits, bytes and numbers
- Basic MIPS assembly programming
- Integer arithmetic
- Floating point arithmetic
- More MIPS assembly programming
- Basic Processor architecture
Why study CSA?

Course Textbook

Computer Organisation and Design
The Hardware/Software Interface; 3rd Edition
By David A Patterson and John L Hennessey

- Price: £39.99
- Copies in the library

The University of Nottingham
Lab Software: PCSpim

PC = 00000000  EPC = 00000000  Cause = 00000000  BadVAddr = 00000000
Status = 3000ff10  HI = 00000000  LO = 00000000

General Registers
R0 (r0) = 00000000  R8 (t0) = 00000000  R16 (s0) = 00000000  R24 (t8) = 00000000
R1 (at) = 00000000  R9 (t1) = 00000000  R17 (s1) = 00000000  R25 (t9) = 00000000
R2 (v0) = 00000000  R10 (t2) = 00000000  R18 (s2) = 00000000  R26 (k0) = 00000000

[0x00400000] 0x8fa40000  lw $4, 0($29) ; 175: lw $a0 0(Ssp)
[0x00400004] 0x27a50004  addiu $5, $29, 4 ; 176: addiu $a1 $s$0 4
[0x00400008] 0x24a60004  addiu $6, $5, 4 ; 177: addiu $a2 $a1 4
[0x0040000c] 0x00041080  sll $2, $4, 2 ; 178: sll $v0 $a0 2
[0x00400010] 0x00c23021  addu $6, $6, $2 ; 179: addu $a2 $a2 $v0
[0x00400014] 0x00000000  jal 0x00000000 [main] ; 180: jal main

DATA
[0x10000000]...[0x10040000] 0x00000000

STACK
[0x7ffe0000] 0x00000000

All Rights Reserved.
DOS and Windows ports by David A. Carley (dac@cs.wisc.edu).
Copyright 1997 by Morgan Kaufmann Publishers, Inc.
See the file README for a full copyright notice.
Loaded: C:\Program Files\PCSpim\exceptions.s
What is PCSpim?

- Assembler and simulator for the MIPS32 architecture
- Free to install on your own machine
- Available on the lab computers
- `xspim` for X11/GNU/Linux or X11/MacOS
Lectures twice a week
  - Mondays at 12:00 and Thursdays at 16:00
Tutorials (starting next week)
  list will be on the web, any changes, please email lyh.
Labs: Tuesdays 10-12 or Wednesdays 11-13
Compulsory weekly practical coursework (25 %)
Labs and tutorials are **starting next week**.
Lists will be on the web, any changes, please email lyh.
Set 8 courseworks, count only 5 best.
Can work in small groups (1 to 3 students)
  - but assessed individually
  - may have to explain your solution to the tutor.

Slides, exercises and additional material: http://www.cs.nott.ac.uk/~txa/g51csa/