

Organização de Computadores – 2005/2006

Paulo Ferreira
paf@dei.isep.ipp.pt

Fevereiro de 2006

| | |
|--|-----------|
| Informação | 2 |
| Docentes | 3 |
| Avaliação | 4 |
| Pontos importantes | 5 |
| Trabalho | 6 |
| Apresentação do trabalho | 7 |
| A Disciplina | 8 |
| Organização de Computadores | 9 |
| Hardware? | 10 |
| Fazer Hardware | 11 |
| Hardware feito em Software | 12 |
| <i>Hardware</i> feito em Sware | 13 |
| Máquinas Virtuais | 14 |
| Comprar Hardware | 15 |
| Programar Hardware | 16 |
| Problemas | 17 |
| Referências 1 | 18 |
| Referências 2 | 19 |
| Referências 3 | 20 |
| Tópicos | 21 |
| Prática | 22 |
| Matéria | 23 |
| Novas coisas | 24 |
| Programação | 25 |
| Conclusão | 26 |

Docentes

- Teóricas
 - ▲ Paulo Ferreira – paf@dei.issep.ipp.pt
- Práticas
 - ▲ Alexandre Bragança – alex@dei.issep.ipp.pt
 - ▲ Paulo Ferreira – paf@dei.issep.ipp.pt
 - ▲ Luís Nogueira – luis@dei.issep.ipp.pt
 - ▲ Orlando Sousa – orlando@dei.issep.ipp.pt

ORGC

Apresentação da disciplina – slide 3

Avaliação

- Trabalho prático – 40% da nota final (40% de nota mínima)
- Exame Final – 60% da nota final
 - ▲ Parte teórica – 50% da nota de exame (40% de nota mínima)
 - Sem consulta
 - ▲ Parte prática – 50% da nota de exame (40% de nota mínima)
 - Com consulta de material em papel trazido pelo próprio
 - Não é permitido o uso de computadores

ORGC

Apresentação da disciplina – slide 4

Pontos importantes

- Se o aluno não tiver nota mínima numa das partes do exame, tem de fazer novamente as duas partes na época de recurso
- O exame é individual
- É possível fazer «melhoria» do trabalho para a época especial

ORGC

Apresentação da disciplina – slide 5

Trabalho

- Lançado a 20 de Março
- Entrega até 28 de Maio
- Trabalhos atrasados ⇒ 30% de desconto na nota por cada dia de atraso
- Entrega:
 - ▲ Por correio electrónico ao docente das aulas práticas
 - ▲ Código fonte + curto relatório descritivo em formato PDF
- Grupos
 - ▲ Máximo dois alunos por grupo

ORGC

Apresentação da disciplina – slide 6

Apresentação do trabalho

- Todos os elementos do grupo devem estar presentes
- Será sorteado um aluno do grupo para fazer a apresentação
- É possível adiar a apresentação uma vez no máximo
- Sem apresentação não há nota de trabalho

ORGC

Apresentação da disciplina – slide 7

A Disciplina

slide 8

Organização de Computadores

- «Arquitectura de Computadores II»
- O que é que existe dentro?
- Como funcionam?
- Como se organizam (dentro)?
- Como se organizam (fora)?
- O que é que apareceu desde que os alunos fizeram ARQC?

ORGC

Apresentação da disciplina – slide 9

Hardware?

Vale a pena aprender como funciona o hardware?

1. Para fazer hardware
2. Para especificar/comprar
3. Para usar da forma mais adequada

ORGC

Apresentação da disciplina – slide 10

Fazer Hardware

- Em Portugal?
- Ferro de soldar e afins?
- Conhecimentos de electrónica?
- No entanto...

ORGC

Apresentação da disciplina – slide 11

Hardware feito em Software

Onde está a fronteira?

- Linguagens de descrição de hardware + FPGAs
- Exemplos: VHDL, Verilog, ...
- Pega-se na descrição de um circuito digital, simula-se, verifica-se e configura-se o hardware mexendo apenas num teclado
- O que é que metemos em hardware, e o que é que fazemos em software? ⇒ *Hardware-Software CoDesign*
- Normal: Placa com μ P + Memória + Periféricos para fazer SW
- Mais recente: Placa com FPGA + memória + ligações
 - ▲ Fazemos o μ P que quisermos + os periféricos e programas que quisermos
- Pac-Man, Asteroids, PowerPC, SPARC, Cray XD-1, Mitrion, Nallatech, ...
- Não vamos dar nada disto. ...

ORGC

Apresentação da disciplina – slide 12

Hardware feito em Sware

Máquinas virtuais

- Java
- CLR (.Net)
- Flash
- Visual Basic
- UCSD-Pascal
- Smalltalk
- Emulação nos *mainframes* (já desde os anos 60)
- VirtualPC
- Rosetta
- 68000 nos Macs PowerPc (ambiente Classic)

ORGC

Apresentação da disciplina – slide 13

Máquinas Virtuais

- Como funcionam?
- Como fazer?
- Como é que as coisas encaixam?

ORGC

Apresentação da disciplina – slide 14

Comprar Hardware

- O que se quer fazer?
- A performance da máquina chega?
- Porquê?
- O que se pode mudar/modificar no hardware?
- Como ler as especificações?

ORGC

Apresentação da disciplina – slide 15

Programar Hardware

Visão do programador – a linguagem «esconde» a máquina, mas...

- Como se consegue uma maior performance?
 - ▲ Ex: tamanho das caches, saltos condicionais, pipelines
- Que suporte há para certas operações?
 - ▲ Ex: memória virtual, *locks*, multiplicações, vírgula flutuante
- Como tirar o máximo partido do Hardware?
 - ▲ Ex: GPUs, MMX, SSE, SSE2...
- Tem de se saber «alguma coisa» de Arquitectura de Computadores

ORGC

Apresentação da disciplina – slide 16

Problemas

- O curso é de informática
- Não é de *Computer Science*
- Não é de *Computer Engineering*
- A maioria dos cursos existentes não são de informática
- Livros existentes têm uma grande ênfase ou na teoria da computação ou na implementação física
- Nós queremos a «visão do programador»

ORGC

Apresentação da disciplina – slide 17

Referências 1


Os tópicos são quase os mesmos, a aproximação será diferente

- Curso 6.004 do MIT: *Computation Structures*
- Curso 6.823 do MIT: *Computer System Architecture*
- «*Computer Organization and Design, the hardware/software interface*», David Patterson e John L. Hennessy; Elsevier/Morgan Kaufman, 620 páginas, 3ª Edição (!), ISBN 1-55860-604-1
 - ▲ O CD que o acompanha tem muito material adicional (texto, tutoriais e simuladores)
 - ▲ Arquitecturas: Processadores MIPS + x86

ORGC

Apresentação da disciplina – slide 18



Referências 2

-  Há outro livro mas chama-se «*Computer Architecture: a quantitative approach*» e é de Hennessy & Patterson ;-)
- O segundo livro é bem mais «pesado»
- O primeiro é mais leve, e pode ser complementado mais facilmente
- Alvo: desde Hardware Designers a programadores
- «*Microprocessors a programmers view*» por Dewar e Smosna, da McGraw-Hill
 - ▲ É de 1990, mas está muito bom (legível e orientado ao software)
 - ▲ 450 páginas, ISBN 0-07-016639-0

ORGC

Apresentação da disciplina – slide 19

Referências 3

- «*Computers systems: a programmer's perspective*», por Randall E. Bryant e David R. O'Hallaron, Prentice Hall, ISBN 0-13-034074-X
 - ▲ cerca de 908 páginas
 - ▲ O que é que pode afectar um programa escrito em C?
 - ▲ Vai desde representação de informação a programação concorrente e em rede
 - ▲ Alvo: programadores avançados
 - ▲  Diferença de preços entre edição internacional e «North American»
-  Os livros usados podem ser uma boa opção financeira, e comparar antes de comprar pode ajudar. ;-)

ORGC

Apresentação da disciplina – slide 20

Tópicos

- História dos Computadores
- Anos 60, CISCs
- Máquinas de pilha
- Micro programação
- *Pipelines*
- *Out of Order Execution e Register Renaming*
- *Branch Prediction e Speculative Execution*
- Hierarquia de memória
- Arquitecturas Super Escalares, *VLIW, Vector Computers*
- SMP, processadores *Multithreaded*
- Computação Embebida e Móvel
- Computação Paralela

ORGC

Apresentação da disciplina – slide 21

Matéria

Normalmente:

- Arquitectura
 - ▲ Programação em Linguagem Assembly
 - ▲ Um RISC com pipeline
 - ▲ Simulação e afins
- Computação Paralela
 - ▲ PVM, MPI, GRID Computing...
 - ▲ Programming for performance
- O que é que terá interesse?

ORGC

Apresentação da disciplina – slide 23

Novas coisas

- Concorrência/Paralelismo
- Fiabilidade/Robustez
- Sistemas Distribuídos
- *Service Oriented Architectures*

ORGC

Apresentação da disciplina – slide 24

Programação

- Orientada à concorrência?
 - ▲ Linguagens «normais» é difícil
 - ▲ Hoje queremos várias tarefas ao mesmo tempo
 - ▲ Queremos que elas corram (não importa o sítio)
 - ▲ Queremos que cooperem com outras tarefas
- Funcional?
 - ▲ Um paradigma que não foi explorado no curso
 - ▲ O que é isso?
 - ▲ Como funciona?
 - ▲ Será que funciona?

ORGC

Apresentação da disciplina – slide 25

Conclusão

Erlang

- Uma COPL – *Concurrency Oriented Programming Language*
- Uma linguagem funcional
- Como organizar um sistema computacional a partir de diferentes processos que comunicam entre si?
- Como obter fiabilidade?

Continua dentro de momentos. . .