T1:____
T2:____
T3:____
T4:____
T5:____
T6:____
T7:____
T: ____
P:_____

Arquitectura de Computadores I - 2003/04 Exame 1ª chamada – 16.Jan.04

NOME: _____ **Nº:**_____

TEÓRICA

As questões devem ser respondidas na própria folha do enunciado. As questões 1 a 4 são de escolha múltipla, e apenas uma das respostas está correcta, valendo 1 valor. Uma resposta errada desconta 1/3 de valor. As questões 5 e 6 valem 2 valores cada. A questão 7 (4 valores) só deve ser respondida pelos alunos que, justificadamente, não fizeram a componente teórico-prática.

1. De acordo com as definições de arquitectura e organização de um computador apresentadas nas aulas, qual dos seguintes items pode ser classificado como fazendo parte da arquitectura?

☐ O conjunto de barramentos

☐ A frequência do relógio do processador

☐ O número de registos visíveis ao programador

☐ A hierarquia de memória

2. Considere o seguinte código em C:

```
char vector[1000];
int i;
for (i=0; i<1000; i++) vector[i]='A';  /***/
```

Qual das seguintes sequências de instruções *assembly* do IA32 optimiza de forma mais agressiva a instrução C assinalada com asteriscos, sabendo que `%eax` está associado a `i` e `%ebx` contém o endereço base de `vector`?

☐
```
   movl %eax, 0
ciclo:
  cmpl %eax, 1000
  jge fim
  leal %ebx, vector
  add %ebx, %eax
  movb (%ebx), 'A'
  incl %eax
  jmp ciclo
```

☐
```
 movl %eax, 0
ciclo:
   leal %ebx, vector
   add %ebx, %eax
   movb (%ebx), 'A'
   incl %eax
   cmpl %eax, 1000
   jl ciclo
```

☐
```
  movl %eax, 0
   leal %ebx, vector
ciclo:
   movb (%ebx,%eax),'A'
   incl %eax
   cmpl %eax, 1000
   jle ciclo
```

☐
```
 movl %eax, 0
   leal %ebx, vector
ciclo:
  movb (%ebx, %eax), 'A'
  movb 1(%ebx, %eax), 'A'
  movb 2(%ebx, %eax), 'A'
  movb 3(%ebx, %eax), 'A'
  addl %eax, %eax, 4
  cmpl %eax, 1000
  jle ciclo
```

3. O número de instruções máquina que constituem um determinado programa (#I) depende:

☐ Da organização, ou micro-arquitectura, do processador e da tecnologia utilizada para o construir.

☐ Da capacidade da memória-cache disponível no sistema.

☐ Da frequência do relógio do processador.

☐ Do conjunto de instruções do processador, do algoritmo e do compilador usados.

4. Relativamente ao mapeamento associativo, a técnica de mapeamento directo de endereços da memória central na cache:

☐ reduz a probabilidade de contenção de dois endereços na mesma linha da cache.

☐ requer o mecanismo menos complexo para determinar em que linha da cache mapeia determinado endereço da memória.

☐ exige maior número de bits de tag.

☐ apresenta o maior hit-time.

NOME: _____ Nº:_____

5. Explique claramente o que entende por um compilador e um interpretador, comparando-os.

NOME: _____ **Nº:**_____

6. Considere o data path de ciclo único do MIPS apresentado em anexo. Caracterize as várias fases de execução da instrução "`lw`", o valor dos sinais de controlo (com excepção de `ALUop`), bem como o momento em que estes são inicializados.
   Porque razão são necessárias 3 componentes com capacidade para adicionar se, com outra organização, a ALU poderia fazer todos estes cálculos?

NOME: _____ Nº:_____

Apenas para os alunos que, justificadamente, não fizeram a componente teórico-prática.

7. O conjunto de instruções de uma família de processadores como o IA32 é desenhado com o objectivo de permitir a execução eficiente de programas, normalmente escritos numa linguagem de alto nível. Ao longo do semestre foi estudada a forma como o modo de endereçamento de memória do IA32 é utilizado para endereçar estruturas de dados especificadas em C. Explique o papel de cada um dos campos do modo de endereçamento para o caso particular de um array de estruturas.

NOME: _____ **Nº:**_____

PRÁTICA

As questões práticas devem ser respondidas em folha separada. A questão 1 vale 4 valores, as questões 2 e 3 valem 2 valores cada.

P1a:___
P1b:___
P1c:___
P1d:___
Σ: ____

1. Considere o programa escrito em linguagem C

```
#include <stdio.h>

typedef struct {
  int val;
  char str[40];
} mtp;

mtp array[200];
int boom, i;

main () {
  for (i=0; i<200; i++) {
    array[i].str[12] = '\0';
    array[i].val = i+5;
  }
  boom = 600;
  for (i=100; i>10; i--)
    boom -= array[i].val;
  printf ("%d\n", boom);
}
```

(a) Complete o programa em baixo, nas linhas marcadas de 1 a 6, de modo a corresponder a uma compilação válida do programa em C.

```
main:
  pushl %ebp
  movl %esp, %ebp
  pushl %ebx
  xorl %edx, %edx
  movl $0, i
  xorl %ecx, %ecx
  leal array, %ebx
.L6:
/*1*/  leal _____(%edx), %eax
  incl %edx
/*2*/  movb $0, __________
  movl %eax, (%ebx, %ecx)
  addl $44, %ecx
/*3*/  cmpl __________
  movl %edx, i
  jle .L6
  movl $600, boom
  movl $100, %edx
  movl $100, i
/*4*/  movl $array+____, %ecx
.L11:
  movl boom, %eax
/*5*/  decl _______
  subl (%ecx), %eax
  subl $44, %ecx
  cmpl $10, %edx
  movl %eax, boom
  movl %edx, i
/*6*/  jg _________
  subl $8, %esp
  pushl %eax
  pushl $.LC0
  call printf
  addl $16, %esp
  popl %ebx
  leave
  ret
```

(b) Identifique e explique as instruções responsáveis pelo teste do 1º ciclo `for`.

NOME: _____ **Nº:**_____

(c) Identifique e explique as instruções responsáveis pelo cálculo do endereço de `array[i].str[12]`.

(d) Que alterações seriam necessárias fazer, no programa assembly, caso o array str passasse a ter 48 caracteres.

NOME: _____ **Nº:**_____

2. Considere o fragmento de código, escrito em assembly do MIPS:

P2a:___
P2b:___
P2c:___
P3a:___
P3b:___
Σ: ____

```
LB1:   li   $s0, 0x5FF000
       lw   $t0, -3($s0)
       lw   $t2, -5($s0)    # QB
       add  $s1, $t2, $t0
       subu $s2, $s1, 3
       move $a1, $s2
```

a) Reescreva o código fornecido, usando apenas instruções nativas do MIPS.

b) Converta a instrução assinalada com "**# QB**" para código máquina do MIPS, apresentando o resultado final em hexadecimal e incluindo na resposta todos os passos intermédios usados.

c) Converta para assembly do MIPS a instrução máquina **0x008B9025** (em hexadecimal), apresentando os cálculos intermédios.

NOME: _____ **Nº:**_____

3. Uma máquina M1, com uma frequência de 2,8 GHz, apresenta um CPI global de 17,81 ciclos/instrução ao executar o programa P, cuja distribuição de tipos de instrução se apresenta na tabela. Para este programa a máquina apresenta uma *miss rate* de instruções de 6% e uma *miss rate* de dados de 8%.

Tipo de Instrução	**Número de Instruções**	**CPI_{CPU}**
Inteiros	$3{,}0*10^8$	2
Vírg. Flutuante	$2{,}0*10^8$	5
Acesso à memória	$3{,}5*10^8$	4
Saltos	$1{,}0*10^8$	3

a) Calcule a *miss penalty* exibida pela máquina M1. Apresente o resultado em nanosegundos.

b) Considere uma máquina M2, em tudo idêntica à máquina M1, excepto que M2 está dotada de uma cache com blocos de 8 palavras e um tempo de acesso à memória de 20 ns mais 10 ns por palavra. Justifique qual delas apresenta um melhor desempenho a executar o programa P, e qual é o tempo de execução de P nessa máquina.