

**Arquitetura de Computadores I**  
**Ano Letivo 2011/12 - 1º Semestre**  
**Teste Prático I**

Nº Mec.: \_\_\_\_\_ Nome: \_\_\_\_\_ Data: 19-Nov-2011

**NOTE BEM:** Leia atentamente todas as questões, comente o código usando a linguagem C e respeite a convenção de passagem de parâmetros e salvaguarda de registos que estudou. Na tradução para o *Assembly* do MIPS respeite rigorosamente os aspetos estruturais e a sequência de instruções indicadas no código original fornecido. O código em C apresentado pode não estar funcionalmente correto, pelo que **não deve ser interpretado**.

~~1) Observe o seguinte screen shot do MARS e responda às perguntas que se seguem.~~

The screenshot shows the MARS MIPS simulator. The 'Text Segment' window displays assembly code with addresses from 0x00400000 to 0x00400058. The 'Registers' window on the right shows the state of MIPS registers, with \$t3, \$t0, and \$t1 highlighted.

Register	Value
\$zero	0x00000000
\$at	0x80000000
\$v0	0x00000004
\$v1	0x00000000
\$a0	0x7fffffef
\$a1	0x7ffffefc
\$a2	0x7ffff000
\$a3	0x00000000
\$t0	0x12345678
\$t1	0x8000ffff
\$t2	0x00000000
\$t3	0x00000000
\$t4	0x00000000
\$t5	0x00000000
\$t6	0x00000000
\$t7	0x00000000
\$s0	0x00000000
\$s1	0x00000000
\$s2	0x00000000
\$s3	0x00000000
\$s4	0x00000000
\$s5	0x00000000
\$s6	0x00000000
\$s7	0x00000000
\$s8	0x00000000
\$s9	0x00000000
\$k0	0x00000000
\$k1	0x00000000
\$gp	0x10008000
\$sp	0x7ffffef8
\$fp	0x00000000
\$ra	0x00400018
pc	0x00400040
hi	0x00000000
lo	0x00000000

Pergunta:	Resposta:
<del>a) Admitindo que o programa começou a ser executado no endereço 0x00400000 qual o resultado do <i>syscall</i> na instrução presente no endereço 0x0040002c?</del>	<b>Pratico</b>
<del>b) Qual o valor actual do <i>Stack Pointer</i>?</del>	<b>0x7FFFFEF8</b>
<del>c) Qual a próxima instrução a ser executada?</del>	<b>or \$t3, \$t0, \$t1 (or \$11, \$8, \$9)</b>
<del>d) Indique qual o código <i>ascii</i> do caracter armazenado no endereço 0x1001000e</del>	<b>0x20</b>
<del>e) Indique o código máquina da instrução <i>and</i> \$t5, \$t0, \$t1</del>	<b>0x01096824</b>
<del>f) Determine e apresente na tabela, <b>em hexadecimal</b>, os valores armazenados nos registos \$t3-\$t8, após a execução de todas as instruções a partir do endereço atual do PC até ao endereço 0x00400054.</del>	
<b>\$t3</b>	<b>0x9234FFFF</b>
<b>\$t5</b>	<b>0x00005678</b>
<b>\$t7</b>	<b>0xE0003FFF</b>
<b>\$t4</b>	<b>0x9234A987</b>
<b>\$t6</b>	<b>0x0003FFFF</b>
<b>\$t8</b>	<b>0x20003FFF</b>

Zona de rascunho

~~2)~~ Codifique em *assembly* do MIPS a seguinte função **main**:

```
int proc_string(char *);

int main(int argc, char *argv[])
{
    static int array[10];
    int *p;
    int res = 0;

    if( (argc >= 10) || (argc < 2) )
        return -1;
    for(p = array; p < (array + argc); p++)
    {
        *p = proc_string( *argv );
        res += *p;
        argv++;
    }
    print_int( res );
    return 0;
}
```

~~3)~~ Codifique em *assembly* do MIPS a seguinte função **proc1**:

```
int proc2(char *, char);

int proc1(char *p, char c, int n)
{
    int res=0;
    do
    {
        if(*p == c)
            res += proc2(p, c + *p);
        else
            *p = *p + 'A' - 'a';
        n--;
    } while (n > 0);

    return res;
}
```

~~4)~~ Codifique em *assembly* do MIPS a seguinte função **array\_proc**:

```
int *array_proc(int *list, int num, int *count)
{
    int i;
    int j = num-1, k = 0;

    for(i = 0; i < num; i++)
    {
        if( (list[i] < 200) || (list[i] >= list[j]) )
        {
            list[j] = list[i] ^ list[j];          // exclusive or
            k++;
        }
        j--;
    }
    *count = k;
    return list;
}
```