

## AULA PRÁTICA N.º 12

### Objectivos:

- Codificação de estruturas em linguagem *assembly* – parte 2.

### Guião:

1. As estruturas podem, à semelhança dos tipos nativos da linguagem, ser organizados num *array*. O programa seguinte declara um *array* de 4 estruturas do tipo **student** e chama uma função para a inicialização dos dados dos 4 elementos desse *array*. O *array* é depois percorrido pela função "**max()**" para determinar o valor máximo e a média das notas.

```
typedef struct
{
    unsigned int id_number;
    char first_name[18];
    char last_name[15];
    float grade;
} student;

#define MAX_STUDENTS 4

void read_data(student *, int);
student *max(student *, int, float *);
void print_student(student *);

int main(void)
{
    static student st_array[MAX_STUDENTS];
    static float media;
    student *pmax;

    read_data( st_array, MAX_STUDENTS );
    pmax = max( st_array, MAX_STUDENTS, &media );
    print_string("\nMedia: ");
    print_float( media );
    print_student( pmax );
    return 0;
}
```

- a) Traduza a função "**main()**" para *Assembly* do MIPS.
- b) A função "**read\_data()**", apresentada na página seguinte, lê da consola os dados de cada aluno e preenche a respectiva estrutura. As estruturas estão organizadas num *array* e, no código apresentado de seguida, utiliza-se acesso indexado. Traduza a função para *Assembly* do MIPS (relembre que **sizeof(student)=44**).

```

void read_data(student *st, int ns)
{
    int i;

    for(i=0; i < ns; i++)
    {
        print_string("N. Mec: ");
        st[i].id_number = read_int();
        print_string("Primeiro Nome: ");
        read_string(st[i].first_name, 17);
        print_string("Ultimo Nome: ");
        read_string(st[i].last_name, 14);
        print_string("Nota: ");
        st[i].grade = read_float();
    }
}

```

```

.eqv stidof, 0
.eqv stfnof, 4
.eqv stlnof, 22
.eqv stgrof, 40
.eqv stmxfn, 18
.eqv stmxln, 15

```

```

.data
st array: .space 176 # 44
read data(st array, 4);

&st[i] = st array + i * 44;
$t1 - i

```

```

move $t0, $a0
li $t2, 44
mul $t2, $t1, $t2
addu $t0, $t0, $t2
li $v0, read_float
syscall
s.s $f0, stgrof($t0)

```

- c) A função "**max()**" determina a média das notas e devolve um ponteiro para a estrutura que contém os dados do aluno com a nota mais elevada. Traduza-a para *Assembly* do MIPS.

```

student *max(student *st, int ns, float *media)
{
    student *p;
    student *pmax;
    float max_grade = -20.0;
    float sum = 0.0;

    for(p = st; p < (st + ns); p++)
    {
        sum += p->grade;
        if(p->grade > max_grade)
        {
            max_grade = p->grade;
            pmax = p;
        }
    }
    *media = sum / (float)ns;
    return pmax;
}

```

```

st array:
i=0 44
st array+44:
i=1 44
st array+88:
i=2 44
st array+132:
i=3 44

```

- d) Finalmente, a função "**print\_student()**" imprime (sem grandes cuidados de formatação) os dados de uma instância da estrutura **student** referenciada pelo ponteiro "**p**". Traduza-a para *Assembly* do MIPS.

```

void print_student(student *p)
{
    print_intu10(p->id_number);
    print_string(p->first_name);
    print_string(p->last_name);
    print_float(p->grade);
}

```

- e) Tendo traduzido para *Assembly* todas as funções apresentadas anteriormente, teste o funcionamento do programa no MARS.