



1. Introdução

Pretende-se com este trabalho que o aluno seja capaz de interpretar o *assembly* gerado pelo `gcc` para a arquitectura IA32, nomeadamente no que respeita à invocação de funções e criação da *stack frame*.

2. Linguagem de alto nível

Escreva em C, usando o editor de texto que considerar mais adequado, o seguinte programa:

prog.c
<pre>typedef struct { char S; int idade; } Telem; int copia (Telem *o, Telem *d, int n) { int ind=0; for (; n>0 ; n--) { if (o[n-1].S==1) { d[ind].S=1; d[ind].idade=o[n-1].idade; ind++; } } return (ind); }</pre>

Note que a função `copia()` copia os elementos do array `o` que tiverem o campo `S=1` para o array `d`. O parâmetro `n` indica o número de elementos de `o`.

3. Compilação

Compile o programa `prog.c` usando o comando

```
gcc -S prog.c
```

e copie o ficheiro `prog.s` para `prog.00.s`

Questão 1 – Qual o factor de escala para os *arrays* do tipo de dados `Telem`?

Questão 2 – Quais os deslocamentos relativamente a `%ebp` de cada um dos parâmetros e variáveis locais?

Questão 3 – Descreva os mecanismos usados para calcular os endereços de `o[n-1].S`, `o[n-1].idade`, `d[ind].S` e `d[ind].idade`.

Compile de novo o programa `prog.c` usando o comando

```
gcc -O1 -S prog.c
```

e copie o ficheiro `prog.s` para `prog.01.s`

Questão 4 – Quais as optimizações introduzidas pelo compilador relativamente à versão anterior?

Questão 5 – Descreva os mecanismos usados para calcular os endereços de `o[n-1].S`, `o[n-1].idade`, `d[ind].S` e `d[ind].idade`.

Questão 6 – Porque é que os registos `%edi`, `%esi` e `%ebx` são guardados na *stack* no início da função e restaurados no fim, não se fazendo o mesmo para os registos `%eax`, `%ecx` e `%edx`?