



1. Introdução

Pretende-se com esta aula prática que os alunos analisem alguns mecanismos de optimização do código gerado por um compilador de C. Para atingir este objectivo o aluno deve desenvolver um pequeno programa em C, compilá-lo e interpretar o código gerado.

Nota: a questão assinalada com **TPC** deve ser resolvida **antes da sessão TP** e entregue ao docente no início da mesma. A folha anexa a este módulo, assinalada como TPC, destina-se a esse fim.

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 { int idade; char nome[10]; } sociot; sociot socios[1000]; void Maius (char *); main () { int i; for (i=0; i<1000 ; i++) Maius (socios[i].nome); }</pre>	<pre>void Maius (char *s) { int j; for (j =0; j< comp (s) ; j++) if (s[j]>='a' && s[j]<='z') s[j]= s[j] + 'A' - 'a'; } int comp (char *s) { int z; for (z=0 ; s[z]!='\0' ; z++); return (z); }</pre>

3. Compilação sem optimização

Compile o programa `prog.c` usando o comando

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

Analisando o código *assembly* das funções `main()` e `Maius()` responda às seguintes questões:

Questão 1 – Identifique os blocos de instruções *assembly* correspondentes a cada instrução C.

TPC : Questão 2 – Onde se encontra armazenada a variável *i* da função `main()`? E a variável *j* da função `Maius()`? E o parâmetro *s*?

4. Compilação com optimização

Compile o programa `prog.c` usando o comando

```
gcc -O2 -S prog.c
```

Analisando o código *assembly* das funções `main()` e `Maius()` responda às seguintes questões:

Questão 3 – Como é testado em `main()` o fim do ciclo? Que aconteceu à variável *i*?

Questão 4 – Onde é mantido o valor do parâmetro *s* durante a execução de `Maius()`? E onde é mantida a variável *j*? Qual é a optimização que o compilador pretende com esta técnica?

Questão 5 – O teste da instrução condicional `if` na função `Maius()` é optimizado. Qual o objectivo do compilador com esta optimização?

5. Loop unrolling

Compile o programa `progopt.c` usando o comando

```
gcc -O2 -S -funroll-all-loops progopt.c
```

Analisando o código *assembly* da função `main()` e comparando com o código da secção 4 responda às seguintes questões:

Questão 6 – Qual a grande diferença relativamente ao código da secção 4?

Questão 7 – Qual o objectivo do compilador? Qual a desvantagem óbvia?

TPC	
Número:	Turno:
Nome:	

Questão 2 – Onde se encontra armazenada a variável *i* da função `main()`? E a variável *j* da função `Maius()`? E o parâmetro *s*?