

# Assembly do IA-32 em ambiente Linux

## TPC7 e Guião laboratorial

Baseado no guião de Alberto José Proença

---

### Objectivo

A lista de exercícios/tarefas propostos no TPC7 / Guião laboratorial continua a analisar o **suporte a estruturas de controlo e a funções em C**, no IA-32. Estes trabalhos experimentais deverão ser realizados no servidor Unix de SC, à semelhança dos trabalhos anteriores.

A resolução deverá ser entregue **impreterivelmente** no início da sessão PL, com a presença do estudante durante a sessão PL para que o TPC seja contabilizado na avaliação por participação. Não serão aceites trabalhos entregues fora da PL.

---

1. Na directoria `/home/shared/TPC7` no servidor remoto Unix de SC encontra-se disponível o ficheiro executável `m-contaN`; copie-o para a sua directoria (**no servidor remoto**) e realize o trabalho a partir de lá.

O ficheiro `m-contaN` contém um programa executável que calcula o somatório dos dígitos (algarismos) de uma cadeia de caracteres, a partir de uma dada posição (tirando partido do facto de que o valor em hexadecimal do código ASCII do símbolo "0" é `0x30`).

O executável foi obtido a partir compilação do programa C seguinte, mas a função `contaN` apresentada não está completa.

```
#include <stdio.h>
#include <stdlib.h>

int contaN(char *s, int c) {
    int result=???;
    for (int i=??? ; s[i]!= ??? ; i???)
        if (s[i] >= '0' && ??? )
            result += ???;
    return result;
}

int main() {
    char cadeia[50];
    int c;
    printf("Introduza a cadeia de caracteres -->\n");
    scanf("%s",cadeia );
    printf("Indique a posicao inicial na cadeia de caracteres -->\n");
    scanf("%d",&c );
    printf("O somatorio dos digitos na cadeia e' -->%d\n",
           contaN(cadeia,c));
    exit(0);
}
```

2. **Teste** o funcionamento do programa a partir da consola usando como entrada de dados uma cadeia de caracteres contendo alguns algarismos em decimal (ex.: "1239aaswe67899") e um inteiro para a posição inicial na cadeia de caracteres.

3. **Utilize** um dos comandos dos TPCs anteriores para visualizar o código simbólico ("desmontado" ou *disassembled*) correspondente à função `contaN` e responda às seguintes questões, indicando as instruções *assembly* relevantes para a resposta:

a. Atribuição dos registos pelo compilador:

- i. Que registos contém cópias dos argumentos `char *s` e `int c` da função?
- ii. Que registos são atribuídos às variáveis locais `result` e `i`? Qual o seu valor inicial?

b. Implementação do ciclo `for`

- i. Identifique a condição de teste/salto do ciclo `for` (nota: lembre-se que os ciclos são implementados como `do { ... } while(condição);`, assim os saltos para endereços menores permitem, em geral, delimitar o corpo do ciclo)
- ii. Como é atualizado o valor de `result` e de `i` dentro do ciclo?
- iii. Indique a condição de paragem do ciclo.

4. **Complete** o código C da função `contaN` com as respostas anteriores:

```
int contaN(char *s, int c) {  
    int result=??? (a.ii);  
    for (int i=??? (a.ii) ; s[i] != ??? (b.iii) ; i??? (b.ii))  
        if (s[i] >= '0' && ??? )  
            result += ??? (b.ii);  
    return result;  
}
```

Nº

Nome:

Turma:

**Resolução dos exercícios (deve ser redigido manualmente)****2) Teste do programa**

Escreva aqui o que apareceu no monitor desde que começou a execução do código, incluindo os caracteres que tiver introduzido e o resultado da execução do código.

**3) Código desmontado da função contaN**

Escreva aqui os comandos que usou para obter o código desmontado (*Assembly*) da função.

**4) Resposta às questões colocadas no enunciado e código *Assembly***

Questão	Registo atribuído pelo compilador	Instrução <i>Assembly</i>
registo com cópia do argumento <code>char *s</code>		
registo com cópia do argumento <code>int c</code>		
registo atribuído à variável local <code>result</code> e valor inicial		<code>result=</code>
registo atribuído à variável local <code>i</code> e valor inicial		<code>i</code>
condição de teste/salto do ciclo <code>for</code> e limites do corpo do ciclo (endereços)		<code>8048____ a 8048____</code>
Atualização do valor de <code>result</code> dentro do ciclo		
Atualização do valor de <code>i</code> dentro do ciclo		
condição de paragem do ciclo		