Assembly do IA-32 em ambiente Linux

TPC7

Alberto José Proença

Objectivo

A lista de tarefas propostas neste guião continua a analisar o **suporte a estruturas de controlo e a funções em C**, no IA-32 e em Linux, com recurso ao depurador (*debugger*) **gdb**.

Ciclo *For*

1. Na seção **Conteúdo** da página da UC na Blackboard encontra-se disponível o ficheiro executável `m-contaN` para Linux. O ficheiro contém um programa executável que calcula o somatório dos dígitos (algarismos) numa cadeia de carateres, a partir de uma dada posição (tirando partido do facto de que o valor em hexadecimal do código ASCII do símbolo "0" é `0x30`).

Este ficheiro foi obtido a partir da consola de um sistema Linux com a execução do comando:

```
gcc –Wall -m32 -O2 -I. contaN.c m-contaN.c -o m-contaN
```

Contudo, após a execução desse comando, o ficheiro `contaN.c` ficou danificado...

O ficheiro `m-contaN.c` contém o seguinte:

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

int contaN(char *s, int c);

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

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

b) Execute de novo o mesmo programa através do `gdb`. Use os comandos disponíveis para examinar código, de forma a visualizar o código simbólico ("desmontado" ou *disassembled*) correspondente à função (e apenas este). **Registe** o código *assembly* obtido.

c) Desconfia-se que a estrutura da função que estava em `contaN.c` seja do tipo:

```
int i;
int result;
???
for ( ??? ; s[i]!= ??? ; ???)
   if (s[i] >= '0' && ??? )
      result += ??? ;
return result;
```

Anote cuidadosamente o código visualizado na alínea anterior tendo em consideração que o resultado da função é devolvido no registo `%eax`.

Identifique no código:

- os registos que são atribuídos às variáveis locais `result` (_____) e `i` (_____)
- os registos que são usados com os argumentos da função _____
- a condição de teste do ciclo `for` _____
- o modo como a variável `i` é atualizada _____
- o código decimal correspondentes aos dígitos representados em *ASCII* _____
- a expressão em C que atualiza o valor de `result` no ciclo_____

d) Com base no resultado das alíneas anteriores, **recupere** o ficheiro `contaN.c`.

Nº	Nome:	Turma:

Resolução dos exercícios (deve ser redigido manualmente)

1.

a) 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.

b) Código desmontado da função `contaN` (com o `gdb`, em *assembly*)

Escreva aqui os comandos que usou para obter o código desmontado da função.

c) Anotação do código *assembly* e resposta às questões colocadas no enunciado.

Mostre aqui o código desmontado pedido na alínea anterior, mas devidamente anotado. Não esquecer que anotação de código deve corresponder a uma explicação do que faz cada instrução numa perspetiva **(i)** de gestão da *stack frame* ou **(ii)** de código C da função.

Assinale no código em cima a resposta a cada uma das 6 questões colocadas no enunciado. Se precisar de acrescentar alguns esclarecimentos adicionais, faça-o a seguir a este texto.

d) Recuperação do ficheiro `contaN.c`

Se conseguiu recuperar o código fonte da função `contaN.c`, apresente-o aqui, com algumas explicações do porquê de certas decisões que tomou.