ISA do IA-32 *(parte 2)*

Teste 4

Nº	**Nome**	**Turma/Grupo/Nº:**
		Total de horas dedicadas a PI+AC na semana anterior **:**

Nota: Apresente sempre o raciocínio ou os cálculos que efectuar; o não cumprimento desta regra equivale à não resolução do exercício.

1. Considere o seguinte programa em C (armazenado num ficheiro) que foi posteriormente compilado para executar num PC (com CPU IA32), com um *link* com um outro ficheiro contendo a função **`fac`**:

```
#include <stdio.h>

int fac(int a);

int main()
{
 int d;

 printf("Introduza um numero inteiro positivo\n");

 scanf("%d",&d );

 printf("O resultado da operacao e':%d\n", fac(d));
 return 0;
}
```

Considere ainda a listagem anexa, obtida durante o desenvolvimento desse programa (com `objdump -d`).

a) **(A)** **Identifique** (e **comente**) as instruções no código simbólico (*assembly*) que salvaguardam o *frame pointer* do **`main`** e criam um novo *frame pointer* para a função **`fac`** .

b) **(R)** **Identifique** e **anote** o código que executa o corpo da função **`fac`**.

c) **(A/R)** Considere o ciclo no corpo da função (provavelmente localizado entre 2 instruções de salto condicional). **Mostre a evolução** do comportamento dos registos que são modificados no interior do ciclo, considerando que o valor introduzido durante a execução do programa foi **4**. Comente os valores lidos.

d) **(R)** A partir do código anotado do corpo da função **`fac`**, o qual contém pelo menos uma estrutura de controlo, **identifique a(s) expressão(ões) de teste** (no código em *assembly*) que deverá(ão) estar presente(s) no código fonte em C, estabelecendo a sua correspondência com o código fonte.

e) **(R/B)** **Idem, para a(s) estrutura(s) de controlo** que provavelmente deverá(ão) estar presente(s) no código fonte em C, **e recupere o ficheiro original `fac`**.

Nº	**Nome**	**Turma/Grupo/Nº:**

```
main-s1:     file format elf32-i386

Disassembly of section .init:

080482c0 <_init>:
...
Disassembly of section .plt:

...
Disassembly of section .text:

08048328 <_start>:
...
0804834c <call_gmon_start>:
...
08048370 <__do_global_dtors_aux>:
...
<__do_global_dtors_aux+0x38>
...
<__do_global_dtors_aux+0x31>
...
<__do_global_dtors_aux+0x1c>
...
080483ac <frame_dummy>:
...

080483d8 <fac>:
 80483d8:    55                          push   %ebp
 80483d9:    89 e5                       mov    %esp,%ebp
 80483db:    8b 55 08                    mov    0x8(%ebp),%edx
 80483de:    83 fa 01                    cmp    $0x1,%edx
 80483e1:    b8 01 00 00 00              mov    $0x1,%eax
 80483e6:    7e 09                       jle    80483f1 <fac+0x19>
 80483e8:    0f af c2                    imul   %edx,%eax
 80483eb:    4a                          dec    %edx
 80483ec:    83 fa 01                    cmp    $0x1,%edx
 80483ef:    7f f7                       jg     80483e8 <fac+0x10>
 80483f1:    c9                          leave
 80483f2:    c3                          ret
 80483f3:    90                          nop

080483f4 <main>:
 80483f4:    55                          push   %ebp
 80483f5:    89 e5                       mov    %esp,%ebp
 80483f7:    83 ec 08                    sub    $0x8,%esp
 80483fa:    83 e4 f0                    and    $0xfffffff0,%esp
 80483fd:    83 ec 0c                    sub    $0xc,%esp
 8048400:    68 08 85 04 08              push   $0x8048508
 8048405:    e8 de fe ff ff              call   80482e8 <puts@plt>
 804840a:    58                          pop    %eax
 804840b:    5a                          pop    %edx
 804840c:    8d 45 fc                    lea    0xfffffffc(%ebp),%eax
 804840f:    50                          push   %eax
 8048410:    68 4f 85 04 08              push   $0x804854f
 8048415:    e8 de fe ff ff              call   80482f8 <scanf@plt>
 804841a:    58                          pop    %eax
 804841b:    ff 75 fc                    pushl  0xfffffffc(%ebp)
 804841e:    e8 b5 ff ff ff              call   80483d8 <fac>
 8048423:    5a                          pop    %edx
 8048424:    59                          pop    %ecx
 8048425:    50                          push   %eax
 8048426:    68 30 85 04 08              push   $0x8048530
 804842b:    e8 e8 fe ff ff              call   8048318 <printf@plt>
 8048430:    31 c0                       xor    %eax,%eax
 8048432:    c9                          leave
 8048433:    c3                          ret

08048434 <__libc_csu_init>:
...
0804847c <__libc_csu_fini>:
...
080484c0 <__do_global_ctors_aux>:
...

Disassembly of section .fini:

080484e4 <_fini>:
...
```