



Conteúdo Teste Revisar envio do teste: Teste2.2-26Maio

Revisar envio do teste: Teste2.2-26Maio

Utilizador	Miguel Caçador Peixoto .
Curso	[20-21] Sistemas de Computação [MIEFIS]
Teste	Teste2.2-26Maio
Iniciado	26-05-2021 16:19
Enviado	26-05-2021 17:23
Data do vencimento	26-05-2021 17:30
Status	Completada
Resultado da tentativa	137,41176 em 200 pontos
Tempo decorrido	1 hora, 4 minutos de 1 hora e 5 minutos
Resultados exibidos	Todas as respostas, Respostas enviadas, Respostas corretas, Perguntas respondidas incorretamente

Pergunta 1

54 em 60 pontos



Considere as figuras com a listagem de uma função em C e dois fragmentos da compilação desse código para *assembly* do IA-32 sem e com otimizações:

Código C da função	Código <i>assembly</i> compilado com -O0 seguido de objdump -d	Código <i>assembly</i> compilado com -O2 seguido de objdump -d (parte 1)
<pre>int soma_conta (int *a, int n, int *s) { int conta=0; int i; for(i=0; i < n; i++) { if (a[i] > 20 && a[i] < 30) { *s += a[i]; conta++; } } return conta; }</pre>	<pre>080483b4 <soma_conta>: 80483b4: 55 push %ebp 80483b5: 89 e5 mov %esp,%ebp 80483b7: 57 push %edi 80483b8: 83 ec 08 sub \$0x8,%esp 80483bb: c7 45 f8 00 00 00 00 movl \$0x0,0xffffffff(%ebp) 80483c2: c7 45 f4 00 00 00 00 movl \$0x0,0xffffffff4(%ebp) 80483c9: 8b 45 f4 mov 0xffffffff4(%ebp),%eax 80483cc: 3b 45 0c cmp 0xc(%ebp),%eax 80483cf: 7c 02 jl 80483d3 <soma_conta+0x1f> 80483d1: eb 4c jmp 804841f <soma_conta+0x6b> 80483d3: 8b 45 f4 mov 0xffffffff4(%ebp),%eax 80483d6: 8d 14 85 00 00 00 00 lea 0x0(,%eax,4),%edx 80483dd: 8b 45 08 mov 0x8(%ebp),%eax 80483e0: 83 3c 10 14 cmpl \$0x14,(%eax,%edx,1) ...</pre>	<pre>0804839c <soma_conta>: 804839c: 55 push %ebp 804839d: 89 e5 mov %esp,%ebp 804839f: 53 push %ebx 80483a0: 56 push %esi 80483a1: 57 push %edi 80483a2: 8b 75 0c mov 0xc(%ebp),%esi 80483a5: 31 db xor %ebx,%ebx 80483a7: 31 c9 xor %ecx,%ecx 80483a9: 39 f3 cmp %esi,%ebx 80483ab: 8b 7d 10 mov 0x10(%ebp),%edi 80483ae: 7d 16 jge 80483c6 <soma_conta+0x2a> ...</pre>

Nas duas figuras seguintes estão representadas algumas posições de memória relativas ao quadro de ativação da função (*stack frame*), em que cada retângulo representa 4 bytes.

Preencha cada caixa descrevendo os conteúdos do quadro de ativação (para a versão -O0 e para a versão -O2).

Em ambos os casos considere que o estado da *stack* corresponde à execução do fragmento apresentado em *assembly*.

Para facilitar a correção automática use as seguintes designações para descrever os conteúdos do quadro de ativação: nomes das variáveis do programa, antigo_ebp, ender_regresso, nome de registos (e.g.: %eax). Nos campos não preenchidos coloque o símbolo menos ("-").

a) Quadro de ativação para a versão -O0				b) Quadro de ativação para a versão -O2			
%ebp-12 =>	XX XX XX XX	[s1a]		%ebp-12 =>	XX XX XX XX	[s2a]	
%ebp-8 =>	XX XX XX XX	[s1b]		%ebp-8 =>	XX XX XX XX	[s2b]	
%ebp-4 =>	XX XX XX XX	[s1c]		%ebp-4 =>	XX XX XX XX	[s2c]	
%ebp =>	XX XX XX XX	[s1d]		%ebp =>	XX XX XX XX	[s2d]	
%ebp+4 =>	XX XX XX XX	[s1e]		%ebp+4 =>	XX XX XX XX	[s2e]	
%ebp+8 =>	XX XX XX XX	[s1f]		%ebp+8 =>	XX XX XX XX	[s2f]	
%ebp+12 =>	XX XX XX XX	[s1g]		%ebp+12 =>	XX XX XX XX	[s2g]	
%ebp+16 =>	XX XX XX XX	[s1h]		%ebp+16 =>	XX XX XX XX	[s2h]	

c) Com base na análise dos códigos em *assembly* e nos quadros de ativação (das 2 versões), indique os endereços das instruções que inicializam as variáveis locais:

- variável `conta` : em -O0 [dc0], em -O2 [dc1];

- variável `i` : em -O0 [dc2], em -O2 [dc3];

Resposta Especificada para s1a ✖ -

Resposta Especificada para s1b ✖ -

Resposta Especificada para s1c ✔ %edi

Resposta Especificada para s1d ✔ antigo_ebp

Resposta Especificada para s1e ✔ ender_regresso

Resposta Especificada para s1f	✔ *a	
Resposta Especificada para s1g	✔ n	
Resposta Especificada para s1h	✔ *s	
Resposta Especificada para s2a	✔ %edi	
Resposta Especificada para s2b	✔ %esi	
Resposta Especificada para s2c	✔ %ebx	
Resposta Especificada para s2d	✔ antigo_ebp	
Resposta Especificada para s2e	✔ ender_regresso	
Resposta Especificada para s2f	✔ *a	
Resposta Especificada para s2g	✔ n	
Resposta Especificada para s2h	✔ *s	
Resposta Especificada para dc0	✔ 80483bb	
Resposta Especificada para dc1	✔ 80483a7	
Resposta Especificada para dc2	✔ 80483c2	
Resposta Especificada para dc3	✔ 80483a5	
Respostas Corretas para s1a		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[il]\s*	
Respostas Corretas para s1b		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[cC][oO][nN][tT][aA]\s*	
Respostas Corretas para s1c		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*\%*\s*[eE][dD][iI]\s*	
Respostas Corretas para s1d		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[aA][nN][tT][iI]?[gG]?[oO]?[_]*[eE][bB][pP]\s*	
Respostas Corretas para s1e		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[eE][nN][dD][eE]?[rR]?[sScCçÇzZ]*[oO]?[_]*[rR][eE][gG][rR][eE][sScCçÇzZ]*[oO]\s*	
Respostas Corretas para s1f		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[?]\s*[aA]\s*	
Respostas Corretas para s1g		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[nN]\s*	
Respostas Corretas para s1h		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[?]\s*[sS]\s*	
Respostas Corretas para s2a		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*\%*\s*[eE][dD][iI]\s*	
Respostas Corretas para s2b		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*\%*\s*[eE][sS][iI]\s*	
Respostas Corretas para s2c		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*\%*\s*[eE][bB][xX]\s*	
Respostas Corretas para s2d		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[aA][nN][tT][iI]?[gG]?[oO]?[_]*[eE][bB][pP]\s*	
Respostas Corretas para s2e		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[eE][nN][dD][eE]?[rR]?[eE]?[sScCçÇzZ]*[oO]?[_]*[rR][eE][gG][rR][eE][sScCçÇzZ]*[oO]\s*	
Respostas Corretas para s2f		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[?]\s*[aA]\s*	
Respostas Corretas para s2g		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	\s*[nN]\s*	
Respostas Corretas para s2h		

Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	<code>\s*[*]?[s*][sS]\s*</code>	
Respostas Corretas para dc0		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	<code>\s*(0[xX])?\s*80483b[bB]\s*</code>	
Respostas Corretas para dc1		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	<code>\s*(0[xX])?\s*80483[aA]7\s*</code>	
Respostas Corretas para dc2		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	<code>\s*(0[xX])?\s*80483[cC]2\s*</code>	
Respostas Corretas para dc3		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
✔ Correspondência de padrão	<code>\s*(0[xX])?\s*80483[aA]5\s*</code>	

Pergunta 2

49,41176 em 60 pontos



Considere as figuras com a listagem da mesma função em C e um fragmento da compilação desse código para *assembly* do IA-32 (com -O2), imediatamente a seguir ao da questão anterior:

Código C da função

```
int soma_conta (int *a, int n, int *s)
{
    int conta=0;
    int i;

    for(i=0; i < n; i++)
    {
        if (a[i]>20 && a[i]<30)
        {
            *s += a[i];
            conta++;
        }
    }
    return conta;
}
```

Assembly compilado com -O2 seguido de objdump -d (parte 2)

```
...
8048393: mov    0x8(%ebp),%edx
8048396: mov    (%eax,%ebx,4),%eax
8048399: lea    0xffffffff(%eax),%edx
804839c: cmp    $0x8,%edx
804839f: jan    80483a4 <soma_conta+0x25>
80483a1: add    %eax,(%edi)
80483a3: inc    %ecx
80483a4: inc    %ebx
80483a5: cmp    %esi,%ebx
80483a7: jl     8048393 <soma_conta+0x14>
...
```

a) Preencha a tabela em baixo, mapeando as variáveis locais e argumentos (do código C da função) aos registos. Indique também o endereço (no código apresentado acima) da primeira instrução no *assembly* que escreva no registo correspondente a cada variável.

Variável	Registo	Endereço da instrução
a	[r1]	[e1]
a[i\]	[r2]	[e2]
conta	[r3]	[e3]
i	[r4]	[e4]
n	%esi	-
s	[r5]	-

b) Pretende-se modificar a expressão no corpo do ciclo `for` no código fonte

```
*s += a[i\];
```

para

```
s = s + a[i\] + i * 2;.
```

Codifique esta expressão numa única instrução *assembly*:

[b1] [b2] (%edi, [b3]), [b4]

c) Pretende-se modificar a estrutura do ciclo `for` para um `do..while`.

Preencha os espaços em branco do código *assembly* equivalente ao ciclo `for` do código C apresentado.

Código C

```
do{
    ...
    i--;
}while(i >= n);
```

Código assembly

```
while: ...
    decl [c1]
    cmpl [c2] , %esi
    [c3] out
    jmp [c4]
out: ...
```


Resposta Especificada para r1 ✔ %edx


Resposta Especificada para e1 ✔ 8048393

Resposta Especificada para r2 ✔ %eax


Resposta Especificada para e2 ✔ 8048396


Resposta Especificada para r3 ✔ %ecx


Resposta Especificada para e3  80483a3


Resposta Especificada para r4  %ebx


Resposta Especificada para e4  80483a4

Resposta Especificada para r5  %edi


Resposta Especificada para b1  movl


Resposta Especificada para b2  %eax


Resposta Especificada para b3  %ebx, 2

Resposta Especificada para b4  %edi

Resposta Especificada para c1  %ebx

Resposta Especificada para c2  %ebx

Resposta Especificada para c3  [Sem Resposta]

Resposta Especificada para c4  8048393

Respostas Corretas para r1		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][dD][xX]\s*</code>	
Respostas Corretas para e1		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*0?[xX]?0?8048393\s*</code>	
Respostas Corretas para r2		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][aA][xX]\s*</code>	
Respostas Corretas para e2		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*0?[xX]?0?8048396\s*</code>	
Respostas Corretas para r3		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][cC][xX]\s*</code>	
Respostas Corretas para e3		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*0?[xX]?0?80483a3\s*</code>	
Respostas Corretas para r4		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][bB][xX]\s*</code>	
Respostas Corretas para e4		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*0?[xX]?0?80483a4\s*</code>	
Respostas Corretas para r5		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][dD][iI]\s*</code>	
Respostas Corretas para b1		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*[iL][eE][aA][iL]?s*</code>	
Respostas Corretas para b2		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][aA][xX]\s*</code>	
Respostas Corretas para b3		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][bB][xX]\s*,\s*2\s*</code>	
Respostas Corretas para b4		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*\%*[eE][dD][iI]\s*</code>	
Respostas Corretas para c1		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*%?[eE][bB][xX]\s*</code>	
Respostas Corretas para c2		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*%?[eE][bB][xX]\s*</code>	
Respostas Corretas para c3		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*[jJ][gG]\s*</code>	
Respostas Corretas para c4		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
 Correspondência de padrão	<code>\s*[wW][hH][iI][lL][eE]\s*</code>	



Considere o estado do programa no ponto de paragem (*breakpoint*) indicado na figura e um fragmento da compilação do código C para *assembly*.

Código da função em C	Assembly compilado com -O2 seguido de objdump -d	Breakpoint em soma_conta (compilado com -O2)
<pre>int soma_conta (int *a, int n, int *s) { int conta=0; int i; for(i=0; i < n; i++) { if (a[i]>20 && a[i]<30) { *s += a[i]; conta++; } } return conta; }</pre>	<pre>08048390 <soma_conta>: ... 8048396: 8b 75 0c mov 0xc(%ebp),%esi 8048399: 31 db xor %ebx,%ebx 804839b: 31 c9 xor %ecx,%ecx 804839d: 39 f3 cmp %esi,%ebx 804839f: 8b 7d 10 mov 0x10(%ebp),%edi 80483a2: 7d 16 jge 80483ba 80483a4: 8b 45 08 mov 0x8(%ebp),%eax 80483a7: 8b 14 88 mov (%eax,%ecx,4),%edx 80483aa: 8d 42 eb lea 0xfffffeb(%edx),%eax 80483ad: 83 f8 08 cmp \$0x8,%eax 80483b0: 77 03 ja 80483b5 80483b2: 01 17 add %edx,(%edi) 80483b4: 43 inc %ebx 80483b5: 41 inc %ecx 80483b6: 39 f1 cmp %esi,%ecx 80483b8: 7c ea jl 80483a4 ...</pre>	<pre>(gdb) info registers eax 0xbfffd720 ecx 0x1 edx 0xc ebx 0x1 esp 0xbfffd700 ebp 0xbfffd70a esi 0x4 edi 0xbfffd730 eip 0x80483b5 (gdb) x/8xw \$esp 0xbfffd700: 0x0016d51a 0x00241f60 0x00000005 0xbfffd700 0xbfffd710: 0x0804841e ?????????? ?????????? ??????????</pre>

- a) Indique os valores armazenados nas posições da pilha indicadas por `%ebp+12` [r2] e `%ebp+16` [r3];
- b) Indique o valor do *frame pointer* da função que chamou esta [r4] e o número de *bytes* que a função chamadora tem reservados para variáveis locais e para salvar registos (excluindo `%ebp`)? [r5];
- c) Indique qual o endereço da última instrução executada? [r6]; (nota: existem 2 caminhos...)

Resposta Especificada para r2 ☒ 0x4

Resposta Especificada para r3 ☒ 0xbfffd730

Resposta Especificada para r4 ☒ 0xd738

Resposta Especificada para r5 ☒ 12

Resposta Especificada para r6 ☒ 80483ba

Respostas Corretas para r2		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
<input checked="" type="checkbox"/> Correspondência de padrão	<code>*(0[xX])?0*4*</code>	
Respostas Corretas para r3		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
<input checked="" type="checkbox"/> Correspondência de padrão	<code>*(0[xX])?[bB][fF][fF][dD]730*</code>	
Respostas Corretas para r4		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
<input checked="" type="checkbox"/> Correspondência de padrão	<code>*(0[xX])?[bB][fF][fF][dD]738*</code>	
<input checked="" type="checkbox"/> Correspondência de padrão	<code>*(0[xX])?[fF][fF][bB][fF]0500*</code>	
Respostas Corretas para r5		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
<input checked="" type="checkbox"/> Correspondência de padrão	<code>*24*</code>	
Respostas Corretas para r6		
Método de avaliação	Resposta Correta	Diferenciação de maiúsculas e minúsculas
<input checked="" type="checkbox"/> Correspondência de padrão	<code>*(0[xX])?0?80483[bB]0*</code>	

Pergunta 4

14 em 30 pontos



Responda às duas alíneas seguintes no espaço disponível no final das questões.

Considere as figuras com a listagem da mesma função em C e um fragmento da compilação desse código para *assembly* do IA-32 com algumas otimizações:

Código C da função

Assembly compilado com -O2 seguido de objdump -d (parte 1)

```
int soma_conta (int *a, int n, int *s)
{
    08048390 <soma_conta>:
    8048390: 55     push    %ebp
```

{	8048391: 89 e5	mov	%esp,%ebp
int conta=0;	8048393: 57	push	%edi
int i;	8048394: 56	push	%esi
	8048395: 53	push	%ebx
for(i=0; i < n; i++)	8048396: 8b 75 0c	mov	0xc(%ebp),%esi
{	8048399: 31 db	xor	%ebx,%ebx
if (a[i]>20 && a[i]<30)	804839b: 31 c9	xor	%ecx,%ecx
{	804839d: 39 f3	cmp	%esi,%ebx
*s += a[i];	804839f: 8b 7d 10	mov	0x10(%ebp),%edi
conta++;	80483a2: 7d 16	jge	80483ba <soma_conta+0x2a>
}	...		
}			
return conta;			
}			

a) Introduza comentários/anotações para cada uma das instruções no código *assembly* que foram destacadas a **negrito** na figura.

b) Considere agora que (i) este código foi compilado para uma versão do MIPS também com 32-bits e que (ii) o *instruction set* deste MIPS não tem instruções de `mov` (só tem `load` e `store` para acessos à memória) e tem o mesmo suporte a estruturas de controlo que o IA-32. **Reescreva** este código *assembly* para esta versão do MIPS (pode usar a sintaxe do GNU para o IA-32, substituindo apenas o nome dos registos para `%r0`, `%r1`, ...), **justificando** todas as alterações que introduzir ao código.

Resposta a)

Selecioneada: `push %edi` # Salva-guarda do registo `%edi` para a stack
`mov 0xc(%ebp),%esi` # Estamos a colocar o argumento da função "n" no registo `%esi`
`xor %ebx,%ebx` # Coloca-se registo `%ebx` = 0 (i = 0)
`cmp %esi,%ebx` # Compara-se os registo `%esi` e `%ebx` (que será o equivalente a `i < n`; `%ebx-%esi=i-n`) `jge 80483ba` # Visto que se fez um `cmp` anteriormente, da-se o `jump` para o endereço `0x80483ba` caso o `i>=n` ou seja, caso o `i-n` for maior ou igual a 0.

b)

`store %r0, -4(%r1)`
`store %r1, %r0`
`store %r4, -8(%r1)`
`store %r2, -12(%r1)`
`store %r3, -16(%r1)`

`load 0xc(%r0),%r2`
`xor %r3,%r3`
`xor %ecx,%ecx`
`cmp %r2,%r3`
`load 0x10(%r0),%r4`
`jge 80483ba 0x2a>`

Em vez de dar `push`, dei `store` ao registo que queremos salvar-guardar no stack pointer com desfasamento (stack cresce)

Em vez de dar `mov`, dou `load` a algo da memória para algo do registo.

Resposta [Nenhuma]

Correta:

Domingo, 6 de Junho de 2021 13H19m BST

← OK

