

Arquitetura de Computadores
Licenciatura em Engenharia Informática
1º teste (C) – 2008/04/23 – Duração: 1h30m + 30m

Nome: _____ Número: _____
Classificação: _____

Teste sem consulta.

A interpretação do enunciado faz parte da avaliação. Explícite nas suas respostas todas as hipóteses assumidas. Por favor, tente focalizar a sua respostas para que estas se enquadrem na zona delimitada.

1 Teórica - Escolha Múltipla

Deve assinalar com um **X** a resposta correcta. Cada resposta errada desconta 25% da cotação da pergunta.

Para todas estas questões considere que está no contexto de um computador Intel com o ISA IA-32 e que o assembly utilizado é o NASM.

Q-1 [1.0 val.] Qual é o comportamento do seguinte código C?

```
char a[] = ``hello``, *p;  
for (p = a; *p != ``\0``; p++)  
    *p = (*p) + 1;
```

1. O ciclo corre indefinidamente, pois a condição de paragem não está bem definida, devia ser `p != ``\0```
2. O ciclo executa normalmente, colocando em cada posição do array `a` o símbolo que, na tabela ASCII, sucede ao guardado nessa posição, por exemplo se `a[i] = 'a'` então o novo valor de `a[i]` será 'b'
3. O ciclo executa normalmente, colocando em cada posição do array `a` o valor do elemento seguinte, por exemplo coloca em `a[1]` o valor de `a[2]`
4. A condição inicial (`p = a`) está incorrecta, devia ser `p = &a`
5. Nenhuma das anteriores

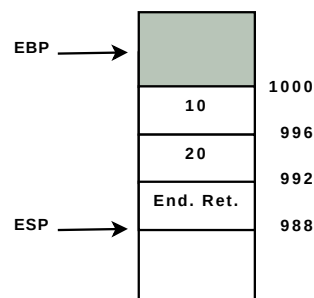
Q-2 [1.0 val.] Considere a definição da seguinte variável: `a: resw 20`

Qual das seguintes sequências de instruções coloca o conteúdo do registo `cx` na última posição do array `a`.

1. `mov ebx, a` `mov esi, 19` `mov [ebx+esi*4], cx`
2. `mov ebx, [a]` `mov esi, 19` `mov [ebx+esi*2], cx`
3. `mov ebx, a` `mov esi, 19` `mov [ebx+esi*2], cx`
4. `mov ebx, [a]` `mov esi, 19` `mov [ebx+esi*4], cx`
5. Nenhuma das anteriores

Q-3 [1.0 val.] Considere que está a executar uma subrotina com o ambiente de execução ilustrado na figura à direita. Indique que sequência de instruções é a correcta para terminar a execução da subrotina, de forma a voltar para o ponto de onde foi chamada e retornar o valor 10, de acordo com a convenção adoptada para o retorno de resultados.

1. `mov eax, 10` `pop ebp` `ret`
2. `mov eax, 10` `ret`
3. `pop ebp` `ret 10`
4. `mov eax, 10` `pop esp` `ret`
5. Nenhuma das anteriores



Q-4 [1.0 val.] Um endereço de memória:

1. tem se ser sempre guardado em registos específicos, como o EIP
 2. pode ter 8, 16 ou 32 bits
 3. tem sempre 32 bits
 4. pode ter 16 ou 32 bits
 5. Nenhuma das anteriores
-

Q-5 [1.0 val.] No contexto da unidade de vírgula flutuante

1. O topo da pilha de operações é guardado no registo FLAGS
 2. O estado da última operação efectuada é guardado no registo FSW
 3. Existem instruções da família **jump** que testam o conteúdo do registo FSW
 4. Os registos ST0 a ST7 têm 64 bits
 5. Nenhuma das anteriores
-

Q-6 [1.0 val.] Considere o seguinte programa C guardado num ficheiro nome `prog.c`.

```
int g(int);
int main() {
    int x = 2;
    return g(x);
}
```

Qual é o resultado da seguinte linha de compilação: `gcc prog.c -o prog`

1. Um erro de compilação porque o código da função `g` não está disponível
 2. Um erro de compilação porque o protótipo da função `g` não está definido
 3. Um erro de compilação porque a utilização da função `g` não está de acordo com o seu protótipo
 4. Um erro de ligação (“linkagem”) porque o código da função `g` não está disponível
 5. Nenhuma das anteriores
-

Q-7 [1.0 val.] Dadas as seguintes variáveis (à esquerda) diga quais das instruções (à direita) são correctas quanto ao tipo de dados utilizado:

a: **db** 32
b: **dd** 57
c: **dd** 1.0

a) **mov byte** a, 0
b) **mov ah**, [a]
c) **mov ebx**, [b]
d) **fild dword** [a]
e) **fstp word** [c]

1. c) e d)
2. b) e c)
3. b), c) e d)
4. b), c) e e)
5. Nenhuma das anteriores