

## Exercícios de preparação para teste

Nas questões de escolha múltipla, existe apenas uma resposta certa.

1. Considere que  $W3=0x8ABC0DEF$  antes de executar o código seguinte. Indique o valor final de  $W3$ .

```
str      W3, [X0]
ldrsb    W3, [X0]
```

- A.  $0x8ABC0DEF$       B.  $0xFFFFFFFF$       C.  $0x000000EF$       D.  $00000DEF$

2. Qual das seguintes instruções dá erro de compilação?

- A. `SUB X2,X10,X2,ASR #2`    B. `ANDS W1,W2,W3`    C. `SMULH X0,X0,X0`    D. `LDR W1,[W0,#4]`

3. Considere o seguinte programa composto por código C e *assembly* AArch64.

<pre>extern unsigned long int SUBR_T2(char *p);  int main(void) { char s[] = "30 de Abril"; unsigned long int res;  res = SUBR_T2(s); printf("0x%x\n", res); //Imprime em hexadecimal return EXIT_SUCCESS;}</pre>	<pre>.text .global SUBR_T2 .type SUBR_T2, "function"  SUBR_T2: LDRB W1, [X0]         CBZ  W1, FIM         ADD  X0, X0, #1         B    SUBR_T2 FIM:     RET</pre>
---	---

Após execução é escrito no ecrã  $0x00000000FEFFFFD3$ .

Qual o endereço de memória ocupado pelo primeiro elemento ('3') da cadeia de caracteres  $s$ ?

- A.  $0x00000000FEFFFFD2$       B.  $0x00000000FEFFFFC8$   
C.  $0x00000000FEFFFFC7$       D.  $0x00000000FEFFFFBC$

4. Suponha que  $x0=0x10000000$  e que a tabela representa o conteúdo da memória após executar a instrução `str w9, [x0]`. Qual o conteúdo inicial de  $w9$  ?

Endereço (hex)	Conteúdo (hex)	
10000003	0F	A. $0xF31600C$
10000002	31	B. $0xC00613F0$
10000001	60	C. $0xF01306C0$
10000000	0C	D. $0xC60310F$

5. Considere que  $SP=0x0805034A0$  e que após execução das instruções seguintes o estado da pilha é o indicado na tabela.

```
SMADDL X1, W1, W1, X1
STR     X1, [SP, #-16]!
```

Endereço (hex.)	Conteúdo (hex.)
0805034A8	0FF
0805034A0	001
080503498	000
080503490	00C

Nestas circunstâncias, pode afirmar-se que o valor inicial de  $X1$  é:

- A. 1      B. -1      C. 0      D. 3

6. Considere a declaração `extern int cp(int x);` e a respetiva rotina em *assembly* AArch64:

<pre>cp:    mov    w2,1       eor    w3,w3,w3 lpp:   cbz    w0,stop       and    w4,w0,w2       add    w3,w3,w4</pre>	<pre>      lsr    w0,w0,1       b      lpp stop:  and    w0,w3,w2       ret</pre>
---	---

Assumindo que a função `cp()` é aplicada a cada elemento da sequência (1,3,5,7) quais os resultados obtidos ?

- A. (1,1,1,1) B. (0,0,0,0) C. (1,0,1,0) D. (1,0,0,1)
7. Qual das seguintes variantes provoca um erro de compilação?
- A. `LDR X9,[X0]` B. `LDRB X9,[X0]` C. `LDRSW X9,[X0]` D. `LDRSB X9,[X0]`
8. A instrução `ROR X5, X5, #32` permuta as duas *words* de X5. Indique o fragmento de código que não realiza esta operação.
- A. `UBFX X6, X5, 32, 32`  
`BFI X5, X5, 32, 32`  
`ORR X5, X5, X6`
- B. `REV X5, X5`  
`REV32 X5, X5`
- C. `MOV X6, X5`  
`BFI X5, X6, 32, 32`  
`UBFX X5, X6, 32, 32`
- D. `MOV W6, W5`  
`LSR X5, X5, 32`  
`ADD X5, X5, X6, LSL 32`

9. Considere a declaração `extern int ism(int x, int y);` e respetiva rotina em *assembly* AArch64.

<pre>ism:  sdiv    w2,w0,w1       msub    w0,w1,w2,w0       cmp     w0,#0</pre>	<pre>      cset    w0,eq       ret</pre>
---	--

Qual o valor de `m = ism(34,7)` ?

A. 1 B. 4 C. 0 D. 6

10. Considere a declaração `extern int vsum(int *a, int n);` e respetiva rotina em *assembly* AArch64 que pretende calcular a soma dos elementos de um vetor.

<pre>vsum: eor     x10,x10,x10 nxt:  cbz     x1,stop       ldrsw   x9,[x0],#4       add     x10,x10,x9</pre>	<pre>      sub     x1,x1,#1       b      nxt stop: mov     x0,x10       ret</pre>
--	---

Qual das afirmações é verdadeira?

- A. A rotina está correta. B. A rotina está errada pois retorna  $n \times v[1]$ .  
 C. A rotina está errada pois retorna  $n \times v[0]$ . D. A rotina está errada pois retorna  $(n - 1) \times v[1]$ .

11. Considerar a execução do fragmento de código indicado abaixo.

```
L1:    cbz     w1, L2
      ldr     w2, [x6], 4
      sub     w1, w1, 1
      eor     w0, w0, w2
L2:    b      L1
L2:    ...
```

Inicialmente, `w0=0`, `w1=12`. Quantas instruções são executadas?

R: \_\_\_\_\_

12. Considere o seguinte programa composto por dois ficheiros.

Ficheiro em linguagem C (main.c):

```
extern int SUBROT1(int *a, int d);
int main()
{
    int n = 0, tam = 5;
    int seq[] = {1, 3, 6, 1, 9};
    n = SUBROT1(seq, tam);
    printf("%d\n", n);
    return EXIT_SUCCESS;
}
```

Ficheiro em linguagem *assembly*:

<pre>SUBROT1: STP    X29,X30,[SP, #-16]!  //&lt;1&gt; MOV    X29,SP  MOV    W10,0 LDR    W12,[X0] ADD    X0,X0,#4 SUB    X1,X1,#1 MOV    W11,#1 CICLO: CBZ    X1,FIM LDR    W13,[X0] CMP    W13,W12 B.LE   FSC                      //&lt;2&gt; ADD    W11,W11,#1 B      PROX</pre>	<pre>FSC: CMP    W10,W11 B.GE   PROX MOV    W10,W11 MOV    W11,#1  PROX: MOV    W12,W13 ADD    X0,X0,#4 SUB    X1,X1,#1 B      CICLO  FIM:   MOV    W0,W10 CMP    W10,W11 B.GE   TERMINAR MOV    W0,W11 TERMINAR: LDP    X29,X30,[SP],#16 RET</pre>
---	---

- Quanta informação, em número de palavras (*words*), é lida de memória ?  
A. 9   B. 5   C. 6   D. 7
- Assumindo que o valor inicial do registo SP é 0xE0, indique em que endereço da pilha se encontra armazenado o registo X30 após a execução da instrução assinalada com <1>.  
A. 0xE0   B. 0xE8   C. 0xD8   D. 0xD0
- Para os valores de invocação indicados em main.c, quantas vezes é tomado o salto assinalado com <2>?  
A. 2   B. 1   C. 3   D. 4
- Para os valores indicados no programa, qual é o resultado apresentado no ecrã?  
A. 5   B. 2   C. 3   D. 9

13. Uma memória *cache* com 64 B/bloco contém 32 KiB de dados. Quantos blocos tem?

- A. 512   B. 2048   C. 256   D. 1024

14. Qual das seguintes afirmações sobre uma memória *cache* do tipo *write-through* é falsa?

- A. O conteúdo da memória principal está sempre atualizado.

- B. No caso de uma falta num acesso de leitura (*read miss*) o valor é lido da memória principal e colocado na *cache* atualizando a etiqueta e alterando o valor de *v* para 1.
- C. No caso de uma falta num acesso de escrita (*write miss*) o valor é escrito na memória principal e na memória *cache* atualizando a etiqueta e alterando o valor de *v* para 1.
- D. No caso de um acerto num acesso de leitura (*read hit*), o valor é lido da memória *cache*.
15. Um CPU está equipado com uma memória *cache* unificada com taxa de faltas  $t_f = 0,1$ . Em média, 20 % das instruções de um programa acedem a dados. Qual das seguintes alternativas de *split cache* apresenta o mesmo desempenho?
- A. I-*cache* com  $t_f = 20\%$  e D-*cache* com  $t_f = 8\%$ ;
- B. I-*cache* com  $t_f = 8\%$  e D-*cache* com  $t_f = 20\%$ ;
- C. I-*cache* com  $t_f = 5\%$  e D-*cache* com  $t_f = 5\%$ ;
- D. I-*cache* com  $t_f = 10\%$  e D-*cache* com  $t_f = 20\%$ .
16. A memória principal usada com um CPU de 2 GHz tem um tempo de acesso de 60 ns. O acesso à memória *cache* demora 0,5 ns. Qual deve ser o valor máximo da taxa de faltas para que o tempo médio de acesso a memória seja 10 ciclos?
- A. 5 %   B. 7,5 %   C. 10 %   D. 2,5 %
17. Qual das seguintes afirmações sobre uma memória *cache* do tipo *write-back* é verdadeira?
- A. Existe a possibilidade de serem feitos 2 acessos a memória principal para apenas um acesso a memória *cache*.
- B. O conteúdo da memória principal está sempre atualizado.
- C. Pode haver um acesso a memória principal mesmo que o acesso a memória *cache* seja um acerto (*hit*).
- D. Não pode ser usada como D-*cache*.
18. O desempenho de uma memória *cache* unificada usada com um CPU de 1 GHz foi considerado insuficiente. Sabendo que o CPI de base é 1, indique a alteração que leva à maior redução do CPI efetivo.
- A. Baixar a taxa de instruções *load/store* de 50 % para 20 %.
- B. Baixar o tempo de acesso à memória externa de 100 ns para 80 ns.
- C. Baixar a taxa de faltas de 15 % para 10 %.
- D. Nenhuma das outras alterações indicadas reduz o CPI efetivo.
19. A tabela seguinte apresenta o conteúdo (em hexadecimal) de uma memória *cache* do tipo *write-back* com 8 blocos de 8 bytes usada como D-*cache* num CPU com endereços de 16 bits.

bloco	conteúdo								etiqueta	v	d
	7	6	5	4	3	2	1	0			
0	aa	cc	de	hf	34	33	11	01	235	1	0
1	bb	ad	45	4f	af	de	21	99	391	1	1
2	cc	34	ab	1f	56	cd	ff	ff	023	1	1
3	dd	67	22	2b	32	56	32	21	198	0	1
4	ee	32	11	9f	aa	ba	ab	bb	311	1	0
5	ff	10	00	04	01	02	03	04	278	0	0
6	11	03	41	32	cc	dd	ee	ff	212	1	1
7	22	01	65	01	05	06	07	08	387	0	1

- (a) Como é decomposto o endereço para acesso à memória *cache*? Justifique.
- (b) Indique (se possível) o valor (byte) em memória principal no endereço  $0xc467$ . Justifique.
- (c) Explique quais as alterações que ocorrem na *cache* e na memória principal durante a leitura do valor (byte) residente no endereço  $0xe48d$ .
20. Um CPU com endereços de 24 bits está equipado com uma memória *cache* de dados do tipo *write-through*. Etiqueta e índice têm, respetivamente, 14 e 6 bits de comprimento.
- (a) Determinar o número de blocos e o número de bytes por bloco desta memória *cache*.
- (b) Considerar a seguinte situação. São realizadas sucessivamente leituras (de uma palavra) das seguintes posições de memória:  
 $0x3B7C94$ ,  $0x3B6C90$ ,  $0x3B6C98$ ,  $0x3B6C94$   
 Quantos blocos são transferidos de memória principal para memória *cache* por causa dos três últimos acessos?
- (c) A memória *cache* é usada num sistema em que a penalidade de faltas é de 80 ciclos. Qual deve ser o valor máximo da taxa de faltas desta *cache* para que o número médio de ciclos de protelamento *no acesso a dados* não exceda 10?
21. A tabela seguinte apresenta o conteúdo de uma memória *cache* do tipo *write-through* com 8 blocos de 4 bytes usada como *D-cache* num CPU com endereços de 32 bits.

	Conteúdo	Etiqueta	v
0	aa ff cc 33	123456a	0
1	12 34 56 78	7bcd001	1
2	88 b0 3c 2b	7fffd55	1
3	71 ab 3f 6d	7fffd55	1
4	34 ff 13 aa	07f9910	0
5	78 00 9c 23	0000893	1
6	7a 10 9f a3	2900002	1
7	99 43 65 b4	7f01d12	0

- (a) Como é decomposto o endereço para acesso à memória *cache*? Justifique.
- (b) Apresente, justificando, o conteúdo da memória *cache* após a execução das seguintes operações (se em algum caso não for possível conhecer o conteúdo escreva "indeterminado"). Cada registo Rx tem 32 bits.

1:  $R1 \leftarrow 0xfafafafa$   
 2:  $R4 \leftarrow R3 + R2$   
 3:  $R4 \leftarrow 0x0ff32210$   
 4:  $R5 \leftarrow \text{MEM}[R4]$   
 5:  $R5 \leftarrow R4 - R1$   
 6:  $R6 \leftarrow 0xfffffaa8$   
 7:  $\text{MEM}[R6+4] \leftarrow R1$

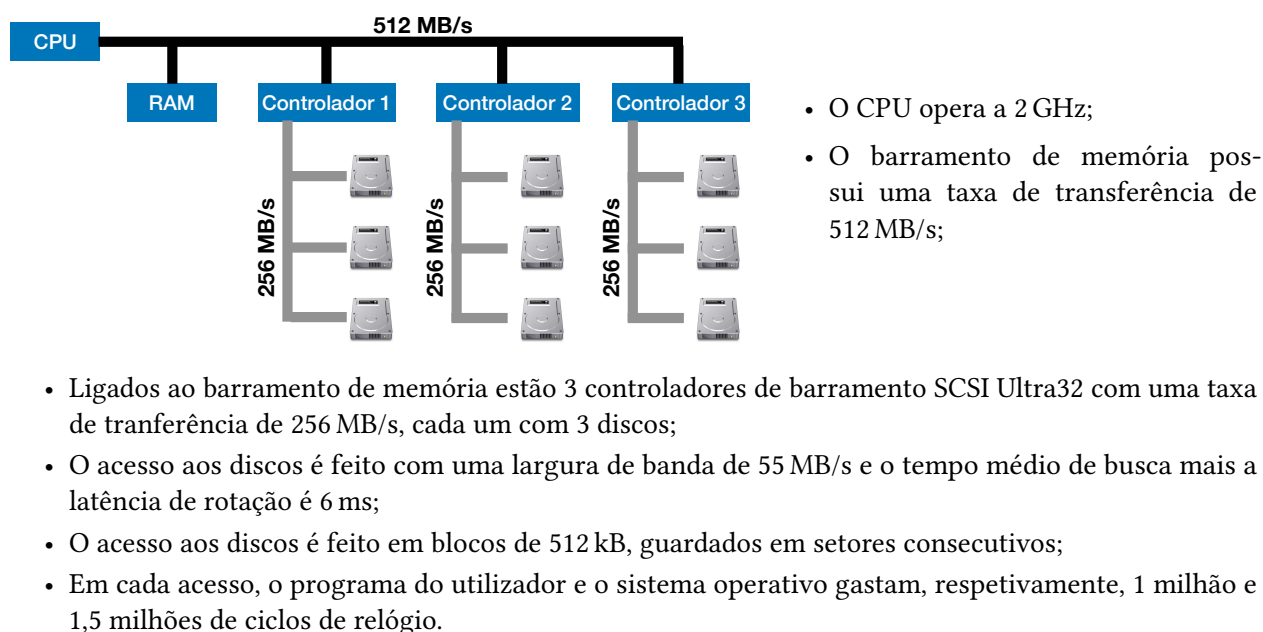
	Conteúdo	Etiqueta	v
0			
1			
2			
3			
4			
5			
6			
7			

22. Um CPU tem endereços de 20 bits e usa uma memória *cache* com 1 palavra/bloco. A memória *cache* do tipo *write-back* usa 6 bits para cada índice e 12 bits para cada etiqueta. Uma parte do conteúdo da memória *cache* está indicada (em hexadecimal) na tabela seguinte:

	conteúdo	etiqueta	v	d
0	12345678	ABC	1	1
1	6548FEAB	123	0	0
2	3C1F56FD	678	1	0
3	AFD12498	567	1	1
4	6198FA34	B7C	1	0
5	1929AAAA	8D1	0	1
...	...	...	...	...

- Determinar o número de blocos e o número total de bits usados na memória *cache*.
- Determinar, se possível, a posição em memória do valor  $0x6198FA34$ .
- Determinar, se possível, qual o valor atualmente armazenado na posição de memória  $0x5670C$ .

23. Considere o computador indicado na figura e que tem as seguintes características:

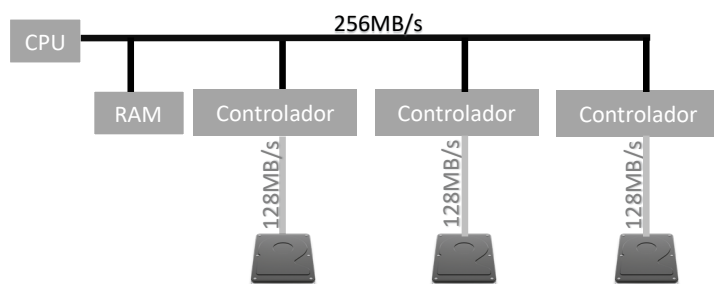


Determine qual dos recursos (CPU, barramento de memória ou discos) limita o desempenho expresso em blocos processados por unidade de tempo. [Considere  $\text{kB} = 10^3 \text{ B}$ ,  $\text{MB} = 10^6 \text{ B}$ .]

24. Um computador possui um CPU que opera a 2,5 GHz e está equipado com um disco que transfere grupos de 8 palavras (4 B cada) a uma taxa de 16 MB/s. [Considere  $\text{kB} = 10^3 \text{ B}$ ,  $\text{MB} = 10^6 \text{ B}$ .]
- O acesso ao disco é feito pela técnica de *polling*. Determinar a fração de tempo de CPU consumida, assumindo que cada operação de *polling* gasta 1000 ciclos de relógio.
  - O disco do computador só transfere dados em 20% do tempo. Calcular a fração de tempo de CPU consumida recorrendo à técnica de interrupção. Assumir que o *overhead* de cada transferência, incluindo o atendimento da interrupção, é de 2000 ciclos de relógio.
  - Tendo em consideração os resultados das alíneas anteriores, determinar a partir de que percentagem de ocupação do disco é mais vantajoso o uso da técnica de *polling*.

question[30]

Considere o computador indicado na figura e que tem as seguintes características:



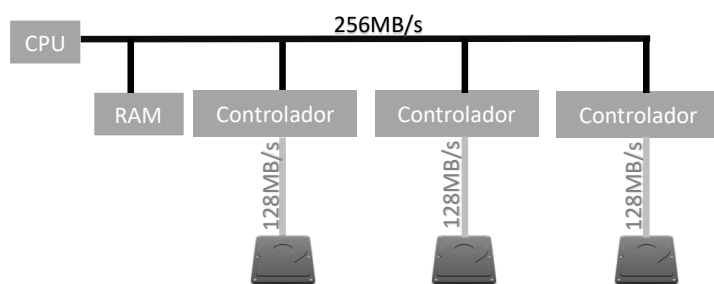
- O CPU opera a 3 GHz;
- O barramento de memória possui uma taxa de transferência de 256 MB/s;

- Ligados ao barramento de memória estão 3 controladores, cada um com o seu disco;
- O acesso aos discos é feito com uma largura de banda de 128 MB/s e o tempo médio de busca mais a latência de rotação é 3 ms;
- O acesso aos discos é feito em blocos de 256 kB, guardados em setores consecutivos;
- Em cada acesso, o programa do utilizador e o sistema operativo gastam, respetivamente, 1 milhão e 2 milhões de ciclos de relógio.

Determine qual dos recursos (CPU, barramento de memória ou discos) limita o desempenho expresso em blocos processados por unidade de tempo. [Considere kB =  $10^3$  B, MB =  $10^6$  B.]

25. Um sistema possui um CPU que opera a 4 GHz. Este sistema possui ainda um disco que transfere dados para o processador em grupos de 4 palavras (2 bytes cada) e tem uma taxa de transferência de dados de 40 MB/s. [Considere kB =  $10^3$  B, MB =  $10^6$  B.]
- Se pretendermos utilizar *polling* como técnica de gestão de periféricos e sabendo que uma operação de *polling* consome 400 ciclos de relógio, qual é a fração de tempo de CPU consumida?
  - Se em vez de *polling* fosse utilizada a técnica de interrupções, qual seria a fração de tempo de CPU consumida? Admita que o *overhead* de cada transferência, incluindo o atendimento da interrupção, é de 600 ciclos de relógio e que o disco está sempre potencialmente ocupado.
  - Comparando os resultados das alíneas anteriores, em que situações é que a técnica de interrupções pode ser vantajosa?

26. Considere o computador indicado na figura e que tem as seguintes características:



- O CPU opera a 3 GHz;
- O barramento de memória possui uma taxa de transferência de 256 MB/s;

- Ligados ao barramento de memória estão 3 controladores, cada um com o seu disco;
- O acesso aos discos é feito com uma largura de banda de 128 MB/s e o tempo médio de busca mais a latência de rotação é 3 ms;
- O acesso aos discos é feito em blocos de 256 kB, guardados em setores consecutivos;
- Em cada acesso, o programa do utilizador e o sistema operativo gastam, respetivamente, 1 milhão e 2 milhões de ciclos de relógio.

Determine qual dos recursos (CPU, barramento de memória ou discos) limita o desempenho expresso em blocos processados por unidade de tempo. [Considere  $\text{kB} = 10^3 \text{ B}$ ,  $\text{MB} = 10^6 \text{ B}$ .]

27. O computador de bordo de uma boia oceanográfica envia por rádio, a cada  $n$  minutos, a posição da boia e um conjunto de informações meteorológicas relevantes, num total de 2500 Bytes. O rádio tem uma potência de 180 W e envia informação a uma cadência de 10 kbit/s; o restante hardware tem uma potência de 1 W. A bateria tem capacidade para armazenar 1200 W h de energia. A boia é lançada ao mar com a bateria totalmente carregada.
- (a) Determine qual deve ser a periodicidade do envio de informação (valor de  $n$ ) por forma a que a boia possa navegar durante 25 dias. Note que o número de envios por hora é dado por  $60/n$ .
  - (b) A cadência de envio da informação é agora de minuto a minuto e a boia foi equipada com painéis fotovoltaicos capazes de fornecer, em média, 160 W h de energia por dia. Determine quantos dias a boia se manterá em funcionamento.
28. Um sistema composto por um CPU, que opera a 1 GHz e um disco duro que transfere dados em grupos de 4 palavras (8 bytes cada) a uma taxa de 8 MB/s, utiliza o método de comunicação com periféricos conhecido como interrupções. Assumindo que o *overhead* de cada transferência, incluindo o atendimento da interrupção, é de 2000 ciclos de relógio e que o disco duro transfere dados durante 10 % do tempo, calcule a percentagem de tempo médio de CPU consumido nas transferências. [Considere  $\text{kB} = 10^3 \text{ B}$ ,  $\text{MB} = 10^6 \text{ B}$ .]