

Princípios da Computação – Exame da Época de Recurso

Licenciatura em Engenharia Informática
2 de Fevereiro de 2023

NOTE BEM:

- **Duração: 60 minutos. Sem consulta.**
- **Identifique o seu teste antes de iniciar a prova. Responda no enunciado.**
- Uma questão é classificada com **um** valor mediante a indicação exclusiva da resposta correcta; e classificada com **zero (0)** valores nos outros casos.

Número: _____ Nome: _____

1. A norma IEEE 754 permite...

- (a) representar conjuntos de caracteres que permitem formar texto.
- (b) representar números inteiros com sinal, com uma única representação do valor zero.
- (c) representar números inteiros com sinal, havendo duas representações para o valor zero.
- (d) representar números reais com sinal em vírgula flutuante.

2. Relativamente a uma sistema computacional de 64 bits, qual das seguintes afirmações é falsa?

- (a) Os registos de uso genérico da CPU têm largura de 64 bits.
- (b) A Unidade Aritmética-Lógica está construída para operar palavras de 64 bits.
- (c) A maior representação possível de um valor inteiro tem 64 bits.
- (d) Permite endereços de memória com largura de 64 bits.

3. Observe a seguinte representação de um valor inteiro de 8 bits: 10001100.

- (a) Se for uma representação em complemento para 2, este número é positivo.
- (b) Se for uma representação em sinal e grandeza, este número é positivo.
- (c) Se for uma representação sem sinal, este número é negativo.
- (d) Se for uma representação sem sinal, este número é positivo.

4. Uma CPU com tecnologia (optimização) *superescalar* permite executar...

- (a) instruções de programas distintos, em paralelo, em unidades de execução distintas.
- (b) múltiplas instruções do mesmo programa, em paralelo, em unidades de execução distintas.
- (c) as três fases do ciclo *fetch-decode-execute* em paralelo, para uma única instrução.
- (d) múltiplas instruções do mesmo programa, em paralelo, na mesma unidade de execução.

5. Nas operações lógicas binárias com cadeias de bits (*bitwise operations*)...

- (a) o operador AND é seleccionado para forçar o bit UM (1) em determinadas posições da cadeia.
- (b) o operador OR é seleccionado para forçar o bit UM (1) em determinadas posições da cadeia.
- (c) o operador NOT é seleccionado para negar os bits em determinadas posições da cadeia.
- (d) o operador XOR é seleccionado para forçar o bit ZERO (0) em determinadas posições da cadeia.

6. Há dois tipos de operações na memória principal: *leitura e escrita*.
- (a) A memória ROM não permite operações de escrita.
 - (b) Uma operação de escrita na RAM conserva o conteúdo nos endereços acedidos.
 - (c) Uma operação de leitura na RAM destrói o conteúdo nos endereços acedidos.
 - (d) Uma operação de acesso à memória, quer seja de leitura ou de escrita, destrói o conteúdo nos endereços acedidos.
7. Suponha dois processadores de fabricantes diferentes.
- (a) Os dois processadores só podem suportar a mesma arquitectura do conjunto de instruções (ISA) se forem exactamente iguais.
 - (b) Os dois processadores podem suportar a mesma arquitectura do conjunto de instruções (ISA) mesmo tendo microarquitecturas diferentes.
 - (c) Para suportarem a mesma arquitectura do conjunto de instruções (ISA), é necessário que ambos os processadores apresentem a mesma configuração de caches.
 - (d) Os dois processadores podem implementar a mesma arquitectura do conjunto de instruções (ISA), e apresentar conjuntos de registos diferentes.
8. Identifique a afirmação falsa.
- (a) Uma interface da linha de comandos (a *shell*, por exemplo) é uma aplicação que permite ao utilizador invocar serviços do sistema operativo de forma simples.
 - (b) O núcleo (ou *kernel*) do sistema operativo estabelece uma camada de abstracção do *hardware* que impede as aplicações de acederem directamente aos dispositivos do sistema.
 - (c) O *kernel* gere todos os recursos do sistema, seguindo estratégias para uma utilização eficaz do sistema.
 - (d) O sistema operativo expõe o *hardware* às aplicações, que podem aceder directamente aos contro-ladores dos dispositivos.
9. Numa *mudança de contexto*...
- (a) um novo processo é criado, e o código do programa é carregado de um ficheiro para a memória.
 - (b) o processador suspende temporariamente a execução de um processo para executar código do *kernel* para atender uma interrupção de hardware.
 - (c) o processador pára de executar as instruções de um processo para passar a executar as instruções de outro processo.
 - (d) um processo é transferido do disco para a memória.
10. Identifique a afirmação falsa, relativa à *shell* do Unix.
- (a) Numa linha de comando, a primeira palavra é o nome do comando.
 - (b) O operador > (símbolo “maior”) redirecciona a saída de um comando (STDOUT) para a entrada do comando seguinte (STDIN).
 - (c) O comando `ls` permite visualizar os conteúdos de um directório, sendo possível filtrar os resultados pelo nome.
 - (d) O comando `rm` permite apagar ficheiros do sistema de ficheiros.
11. Um sistema operativo multiprogramado/multitarefa...
- (a) executa os programas em sequência, do primeiro até ao último.
 - (b) não tem o seu desempenho limitado pela capacidade de memória RAM.
 - (c) caracteriza-se por intercalar no tempo a execução de vários programas pelo processador.
 - (d) não é indicado para conjuntos de aplicações com muitas operações de I/O.

12. O *Process Control Block* (PCB) é a estrutura de dados em que um sistema operativo guarda toda a informação relevante de um processo. Qual a afirmação falsa?
- (a) O PCB permite determinar quais os ficheiros que o processo tem abertos.
 - (b) O PCB permite obter informação de outros processos concorrentes.
 - (c) O PCB mantém o registo actual dos blocos de memória que o processo ocupa.
 - (d) O PCB regista o tempo de execução do processador já utilizado pelo processo.
13. A gestão de memória é um conjunto de técnicas fundamentais para manter a integridade e eficácia de um sistema multiprogramado.
- (a) A protecção de memória impede que o sistema operativo remova um processo da RAM para o disco.
 - (b) A técnica de *swapping* permite que um processo possa alterar as suas instruções durante a execução.
 - (c) O espaço de memória de cada processo é dividido em páginas do mesmo tamanho. O *kernel* mantém um registo da localização de cada página, que pode estar na memória principal ou na memória secundária.
 - (d) A técnica de *swapping* impede que um processo possa alterar as suas instruções durante a execução.
14. Num sistema com escalonamento sem preempção um processo termina a sua execução.
- (a) O processo passa do estado *Running* para o estado *Waiting*.
 - (b) O escalonador selecciona um processo no estado *Waiting* para executar a seguir.
 - (c) O escalonador selecciona um processo no estado *Running* para executar a seguir.
 - (d) O processo encontrava-se no estado *Running* antes de terminar.
15. Num sistema com escalonamento preemptivo, um processo no estado *New* pode passar ao estado...
- (a) *Ready*.
 - (b) *Running*.
 - (c) *Waiting*.
 - (d) *Terminated*.
16. Um sistema tem 8 CPU exactamente iguais (simétricos) num *chip* (multicore). Qual a afirmação falsa?
- (a) Este sistema pode executar, no máximo, 8 processos em paralelo.
 - (b) Um CPU pode aceder e alterar os registos de outro CPU.
 - (c) Este sistema pode ter muitos (mais de 8) processos concorrentes em simultâneo, sendo as suas execuções intercaladas no tempo, nos núcleos disponíveis.
 - (d) Todos os CPU concorrem no acesso aos restantes recursos do sistema.
17. Pretende-se desenvolver um *script* que descarregue automaticamente uma série de ficheiros a partir da *web*. Este *script* deverá receber o nome de um ficheiro de texto indicado pelo utilizador como parâmetro na linha de comando. O ficheiro terá que conter uma lista de localizadores uniformes de recursos (*Uniform Resource Locator*, URL), um URL por linha, conforme o exemplo em baixo:
- ```
1 http://api.ipma.pt/open-data/forecast/meteorology/cities/daily/1131200.json
2 https://isep.sport.pt/atleta/1220000/act-20230120-1830.json
3 https://isep.sport.pt/atleta/1220000/act-20230121-1900.json
4 ...
```

(Continua na próxima página.)

Desenvolva um script (na caixa de resposta) que cumpra os requisitos enumerados em baixo. O script deve terminar sempre que um dos requisitos não se verifique.

R1: Recebe exactamente um argumento passado como parâmetro na linha de comandos: o nome do ficheiro de texto.

R2: O ficheiro de texto indicado como parâmetro tem que existir.

R3: O utilizador deve ser informado dos URL que não foram descarregados (por erro).

R4: O script deverá informar o número de URL descarregados com sucesso.

DICA 1: O comando `curl` permite descarregar um URL criando uma cópia local, utilizando a opção `-o`, como no exemplo em baixo. O `curl` retorna o código de saída 0 (zero) em caso de sucesso.

```
1 curl -o um_URL_aqui
```

DICA 2: Um ficheiro pode ser lido linha-a-linha para uma variável do seguinte modo:

```
1 while read -r linha do
2 echo $linha
3 done < nome_do_ficheiro
```

**FIM**