

Princípios da Computação – Exame da Época Especial de Setembro

Licenciatura em Engenharia Informática
4 de Setembro de 2023

NOTE BEM:

- **Duração: 60 minutos. Sem consulta.**
- **Identifique o seu teste antes de iniciar a prova. Responda no enunciado.**
- Indique inequivocamente a sua resposta a cada questão.
- A indicação de uma resposta errada resulta na avaliação da questão em **zero (0)** valores.
- A indicação exclusiva da resposta correcta resulta na avaliação da questão em **um (1)** valor.

Número: _____ Nome: _____

1. **Admita que tem um ficheiro de texto com um programa escrito numa determinada linguagem de programação de alto-nível.**
 - (a) As instruções nesse ficheiro podem ser imediatamente executadas pelo processador.
 - (b) Se a linguagem de programação for interpretada, será sempre necessário ter um compilador específico à linguagem de programação para executar o programa linha-a-linha.
 - (c) Um compilador específico para essa linguagem de programação irá executar as instruções em código-máquina, necessárias para levar a cabo cada instrução do programa.
 - (d) Um compilador específico à linguagem de programação irá traduzir o código-fonte, produzindo um ficheiro executável em código-máquina.
2. **Os constituintes básicos de um processador são a Unidade de Controlo, a Unidade Aritmética e Lógica, e os registos.**
 - (a) A Unidade de Controlo realiza as operações lógicas.
 - (b) Os registos são de uso exclusivo da Unidade de Controlo.
 - (c) Os registos são de uso exclusivo da Unidade Aritmética e Lógica.
 - (d) Os registos de uso geral são partilhados pela Unidade de Controlo e pela Unidade Aritmética e Lógica.
3. **Observe a seguinte representação de um valor inteiro em 8 bits: 10001100.**
 - (a) Se for uma representação com sinal em complemento para 2, este número é negativo.
 - (b) Se for uma representação sem sinal, este número é negativo.
 - (c) Se for uma representação com sinal em complemento para 2, este número é positivo.
 - (d) Se for uma representação sem sinal, é impossível determinar se o número é positivo ou negativo.
4. **Na arquitectura de Von Neumann...**
 - (a) a memória ROM faz parte do subsistema de I/O.
 - (b) o processador executa instruções directamente obtidas do disco.
 - (c) o processador é fabricado com o programa já implementado na sua construção.
 - (d) as instruções e os dados de um programa em execução coabitam em memória.
5. **A protecção de memória é muito importante em sistemas multiprogramados, porque...**
 - (a) impede que um processo aceda às instruções e dados de outro processo.
 - (b) impede que o sistema operativo remova um processo da RAM para o disco.
 - (c) evita que os processos possam partilhar bibliotecas ligadas dinamicamente.
 - (d) previne que um processo altere as suas instruções e dados.

6. Num computador, dois processos concorrem para operar o mesmo dispositivo (e.g., um disco) praticamente ao mesmo tempo.
- (a) O sistema operativo disponibiliza a configuração do dispositivo, e os processos coordenam entre si as operações do dispositivo.
 - (b) Todos os pedidos de operação são aceites pelo sistema operativo, que os organiza e executa (em nome dos processos) de acordo pela ordem que definir.
 - (c) Os processos acedem directamente ao dispositivo. O dispositivo organiza autonomamente as operações concorrentes solicitadas pelos processos.
 - (d) Todos os pedidos de operação são aceites pelo sistema operativo, que os organiza. Após estabelecida uma ordem, cada processo opera directamente o dispositivo.
7. O processamento por lotes (*batch processing*)...
- (a) é ideal para trabalhos interactivos.
 - (b) é utilizado, por exemplo, em supercomputadores para correr longos trabalhos não-interactivos.
 - (c) já caiu em desuso, não sendo actualmente utilizado.
 - (d) é o modelo utilizado nos sistemas computacionais que controlam automaticamente outros sistemas físicos (processos industriais, sistemas automotivos, sistemas médicos, etc.).
8. Quando um computador é iniciado, o processador vai sempre buscar a primeira instrução para executar...
- (a) a um sector específico no primeiro disco do sistema.
 - (b) ao endereço da última instrução executada, antes de ter sido desligado/reiniciado.
 - (c) a um endereço específico, em memória ROM.
 - (d) a um endereço específico, em memória RAM.
9. A técnica de *swapping* permite...
- (a) parar a execução de um processo, cedendo a CPU a um outro processo *Ready* com maior prioridade.
 - (b) parar a execução de um processo que pediu uma operação de I/O, cedendo a CPU a um outro processo *Ready*.
 - (c) libertar memória RAM transferindo (parcial ou totalmente) um processo da RAM principal para a memória secundária.
 - (d) libertar memória RAM transferindo (parcial ou totalmente) da RAM para a memória secundária, o processo que está a ser executado pela CPU.
10. Qual é a operação que não é necessariamente realizada durante uma mudança de contexto (*context switch*)?
- (a) Copiar os valores dos registos do processador para o bloco de controlo (*Process Control Block*, PCB) do processo que estava a ser executado (*Running*).
 - (b) Admitir um processo novo (*New*) para o conjunto de processos prontos (*Ready*).
 - (c) Seleccionar o próximo processo a executar, do conjunto de processos prontos (*Ready*).
 - (d) Restaurar os valores nos registos do processador a partir do bloco de controlo (*Process Control Block*, PCB) do processo que foi escalonado para execução.
11. Assuma dois processadores *P1* e *P2* substancialmente diferentes um do outro, nomeadamente no número de componentes electrónicos que cada um indica ter. Ambos implementam a mesma arquitectura do conjunto de instruções (*Instruction Set Architecture*, ISA). Qual das seguintes afirmações é falsa?
- (a) Ambos os processadores são capazes de executar o mesmo código-máquina, produzindo os mesmos resultados.
 - (b) Os diferentes números de componentes reflectem diferenças nas decisões de projecto, devidas a vários tipos de optimizações como, por exemplo, desempenho, caches, uso de energia ou preço.
 - (c) O conjunto de registos do processador *P1* é exactamente igual ao conjunto de registos do processador *P2*.
 - (d) O menor número de componentes de um dos processadores significa que nem todas as instruções da arquitectura estão implementadas em *hardware*, tendo que ser emuladas por *software*.

12. Em Unix, a *shell* é uma interface da linha de comando que permite ao utilizador carregar e executar programas de uma forma simples. O Unix traz um conjunto de programas utilitários que realizam tarefas específicas muito frequentes. Qual das seguintes afirmações é falsa?
- (a) O comando `chmod` é utilizado para redefinir as permissões de acesso a um ficheiro para os vários tipos de utilizador.
 - (b) O comando `find` localiza e apresenta uma expressão regular no interior de ficheiros de texto.
 - (c) O comando `fg` traz para primeiro plano (*foreground*) um processo que está a correr em segundo plano (*background*).
 - (d) O comando `mv` permite alterar o nome de um ficheiro.
13. Um *shell script* é um programa interpretado pela interface da linha de comando...
- (a) escrito em *assembly*, para resolver problemas de carácter genérico.
 - (b) desenvolvido numa linguagem de alto nível, muito prático para automatizar tarefas de manutenção do sistema.
 - (c) desenvolvido numa linguagem de alto nível, sobretudo para resolver problemas de computação numérica.
 - (d) que é utilizado para carregar o sistema operativo no arranque do computador.
14. Os computadores digitais contemporâneos operam exclusivamente símbolos binários.
- (a) A conversão de um inteiro decimal para binário pode resultar numa representação inexacta (i.e. com erro).
 - (b) Um número racional com representação finita em decimal é sempre convertido num número racional também com representação finita em binário. Desta forma, não há introdução de erro desde que haja um número de bits suficientes.
 - (c) Um número racional com representação finita em decimal pode ser convertido num número racional com representação infinita periódica em binário. Neste caso, há introdução de erro porque não há um número de bits suficientes para representar o valor de forma exacta.
 - (d) Um número racional tem de ser convertido em número inteiro, pois não existe representação para a vírgula decimal num computador.
15. Num sistema com escalonamento preventivo (*preemptive scheduling*)...
- (a) um processo no estado *Running* pode passar para o estado *Waiting*.
 - (b) um processo no estado *Waiting* pode passar para o estado *Running*.
 - (c) um processo no estado *Ready* pode passar para o estado *Terminated*.
 - (d) um processo no estado *Running* pode monopolizar o uso do processador, se não o ceder voluntariamente a outros processos.
16. Considere um sistema com escalonamento não-preventivo (*non preemptive*). Quando um processo solicita uma operação I/O...
- (a) o seu estado passa para *Terminated*.
 - (b) o seu estado continua em *Running*, ficando o processador parado (*idle*) durante todo o tempo da operação I/O.
 - (c) o seu estado passa para *Ready* e o processador fica parado (*idle*) se não houver outros processos prontos para executar.
 - (d) o seu estado passa para *Waiting* e o processador é cedido a outro processo que se encontre pronto para executar.

17. É necessário um script que permita reiniciar um conjunto de contentores docker, indicados pelo utilizador. A aplicação docker identifica cada contentor pelo seu nome, que é uma palavra: por exemplo `contentor_X`. O comando `docker` permite realizar operações com contentores, genericamente através da forma:

```
docker ACÇÃO CONTENTOR.
```

Como exemplo, para reiniciar o contentor `contentor_X` deve ser executado o comando:

```
docker restart contentor_X
```

Desenvolva um script (na caixa de resposta) que cumpra os requisitos enumerados em baixo.

R1: O utilizador insere os nomes de contentores a reiniciar na linha de comandos.

R2: Os contentores deverão ser reiniciados exactamente pela ordem indicada na linha de comando.

R3: Para cada contentor, o script deverá escrever uma mensagem no ecrã a informar se a operação teve sucesso ou falhou.

Nota: A variável `$?` guarda o estado de saída de último comando executado. Se o comando teve sucesso, o estado de saída é zero.

FIM