

**Exame da Época de Recurso**  
2 de fevereiro de 2024

**Duração: 60 minutos. Sem consulta. Responda no enunciado.**

**Número:** \_\_\_\_\_ **Nome:** \_\_\_\_\_

**Selecione a resposta correcta para cada uma das questões. (Resposta errada: 0 valores.)**

1. Qual é a forma direta mais precisa (ou seja, com menor erro) de representar o número  $-2,6$  num computador?
  - (a) Número inteiro sem sinal de 32 bits.
  - (b) Número inteiro com sinal (complemento para 2) de 32 bits.
  - (c) Número em vírgula flutuante (IEEE 754) de 32 bits.
  - (d) Número em vírgula flutuante (IEEE 754) de 64 bits.
2. Superscalar é uma técnica de optimização do desempenho de uma CPU em que...
  - (a) as três fases do ciclo *fetch-decode-execute* são executadas simultaneamente para uma única instrução.
  - (b) instruções de programas distintos são executadas simultaneamente em núcleos de processamento (*cores*) diferentes.
  - (c) se pode observar uma diminuição na taxa de instruções concluídas por unidade de tempo.
  - (d) duas ou mais instruções do mesmo programa são executadas simultaneamente em unidades de execução distintas da CPU.
3. As linguagens de programação de alto nível surgiram na década de 1960 e evoluíram desde então. Qual das seguintes opções é uma característica fundamental das linguagens de alto nível?
  - (a) São diretamente executáveis pela CPU, sem necessidade de tradução.
  - (b) Possuem um conjunto limitado de instruções de baixo nível para otimizar o desempenho dos programas.
  - (c) Exigem a escrita de código em linguagem máquina para obter maior eficiência.
  - (d) Oferecem uma maior abstração, permitindo que os programadores se concentrem na lógica do problema.
4. A arquitetura de von Neumann é um modelo fundamental para a construção de computadores. Qual das seguintes opções é uma característica da arquitetura de von Neumann?
  - (a) Estabelece uma separação física entre a memória de instruções e a memória de dados.
  - (b) Os dados e as instruções partilham espaço na mesma memória física.
  - (c) A execução de múltiplos programas simultaneamente em núcleos distintos.
  - (d) A Unidade Aritmética e Lógica coordena as operações do processador.
5. Qual é a principal função dos registos de uma CPU num computador?
  - (a) Manter um registo de eventos do sistema operativo.
  - (b) Facilitar a comunicação entre diferentes periféricos.
  - (c) Gerir o armazenamento de longo prazo no disco rígido.
  - (d) Armazenar temporariamente dados e instruções durante a execução de programas.

6. É necessário colocar a zero os 16 bits menos significativos de uma palavra de 32 bits, deixando os restantes 16 mais significativos inalterados. Uma solução é aplicar à palavra...
- (a) 16 deslocamentos lógicos para a direita seguidos de 16 deslocamentos lógicos para a esquerda.
  - (b) a operação lógica AND com a máscara de bits 0x0000FFFF.
  - (c) a operação lógica OR com a máscara de bits 0xFFFF0000.
  - (d) a operação lógica AND com a máscara de bits 0x00000000.
7. Os dispositivos de entrada e saída (I/O devices)...
- (a) informam o processador de que terminaram uma operação através de uma interrupção.
  - (b) não têm a capacidade de armazenar dados temporariamente.
  - (c) têm velocidades de operação semelhantes às dos processadores.
  - (d) são controlados directamente pelo processador durante toda a sua operação.
8. Quando abordamos os termos *arquitectura do conjunto de instruções (ISA)* e *microarquitetura*...
- (a) ambos os termos referem-se ao mesmo conceito.
  - (b) a microarquitetura é a especificação que determina o conjunto de instruções suportado por uma família de processadores.
  - (c) a ISA estabelece a organização física (electrónica) do processador.
  - (d) a microarquitetura consiste numa implementação física que atende aos requisitos especificados pela ISA.
9. O que caracteriza um sistema operativo multiprogramado?
- (a) Um programa tem acesso exclusivo ininterrupto à CPU.
  - (b) Apenas um programa é carregado na memória de cada vez.
  - (c) Vários programas podem residir na memória ao mesmo tempo e serem executados concorrentemente.
  - (d) A execução de múltiplos programas só é possível em sistemas com múltiplos processadores.
10. Numa mudança de contexto (*context switch*)...
- (a) o processador pára de executar as instruções de um processo para passar a executar as de outro processo.
  - (b) um novo processo é criado quando o programa é carregado de um ficheiro para a memória.
  - (c) o processo em execução é parado temporariamente para que o SO possa tratar uma interrupção de hardware.
  - (d) um processo é transferido da memória para o disco.
11. Qual a característica associada ao processamento por lotes (*batch processing*)?
- (a) A execução de tarefas de forma interativa em tempo real.
  - (b) A necessidade de uma resposta imediata do utilizador para cada tarefa.
  - (c) O processamento de uma sequência de tarefas sem intervenção direta do utilizador.
  - (d) Diversos programas intercalam a sua execução no processador ao longo do tempo.
12. Uma interface de linha de comando é uma aplicação...
- (a) que, embora não integrada no kernel do sistema operativo, expõe os seus serviços ao utilizador.
  - (b) que corre no espaço do kernel, executando comandos ordenados pelo utilizador.
  - (c) tornada obsoleta pelas interfaces gráficas do utilizador, nos SO contemporâneos.
  - (d) que funciona apenas em linguagens de programação específicas.

13. Um sistema operativo actual...
- (a) deve permitir o acesso directo ao hardware às aplicações do sistema.
  - (b) oculta os pormenores do hardware através de um conjunto de serviços que podem ser requisitados pelas aplicações.
  - (c) estabelece uma interface para as aplicações, que é específica à arquitetura do sistema.
  - (d) requer que as aplicações coordenem entre si a utilização dos recursos da máquina.
14. Identifique a afirmação verdadeira, relativa à *shell* do Unix.
- (a) A primeira palavra numa linha de comando pode não ser um comando.
  - (b) O operador > (símbolo “maior”) redirecciona a saída de um comando (STDOUT) para a entrada do comando seguinte (STDIN).
  - (c) O operador | (*pipe*) redirecciona a saída de um comando (STDOUT) para um ficheiro novo.
  - (d) A saída de um comando pode ser redireccionada para a entrada do comando seguinte através do operador | (*pipe*).
15. Num sistema operativo, o termo *preempção* refere-se a:
- (a) A capacidade de um processo ser voluntariamente suspenso.
  - (b) A execução simultânea de múltiplos processos.
  - (c) A capacidade de um processo ser retirado da CPU involuntariamente pelo sistema operativo.
  - (d) O escalonamento de processos em fila.
16. Os sistemas operativos mantêm um conjunto de registos, designados por *Process Control Block (PCB)*, que lhes possibilita a gestão dos diversos processos no sistema.
- (a) O PCB de um dado processo permite obter informação de outros processos concorrentes.
  - (b) O PCB de um processo não inclui a lista de ficheiros abertos.
  - (c) O PCB mantém o registo atualizado dos blocos de memória que o processo ocupa.
  - (d) Um PCB permite determinar o tempo de execução restante para a conclusão do processo.

**Desenvolva um shell script que resolva o seguinte problema.**

17. Está matriculado num curso de programação em C, no qual lhe foi proposto resolver um conjunto de exercícios. Cada exercício é solucionado mediante o desenvolvimento de um programa que atenda aos requisitos do problema.

Para este propósito, criou um diretório no seu disco onde guardou todos os ficheiros com o código-fonte das soluções que desenvolveu. De forma criteriosa, atribuiu a cada ficheiro um nome que lhe permite identificar prontamente a qual exercício se refere, como por exemplo, `exercicio_23.c`.

O comando `gcc` é utilizado para compilar programas em C, seguindo a forma genérica:

```
1 gcc -o EXECUTAVEL FONTE.c
```

Por exemplo, para construir o programa executável `exercicio_23` a partir da compilação da fonte `exercicio_23.c`, deve ser executado o comando:

```
1 gcc -o exercicio_23 exercicio_23.c
```

No entanto, compilar os vários programas que já desenvolveu pode ser uma tarefa fastidiosa que pode ser aliviada com a assistência de um *shell script*. Este *script* será guardado e executado no mesmo diretório onde se encontram as soluções dos exercícios.

Desenvolva então um *script* (na caixa de resposta) que permita compilar um conjunto de programas em C indicados pelo utilizador, cumprindo os requisitos enumerados abaixo:

- R1: O utilizador introduz uma sequência de nomes de programas (executáveis) na linha de comandos. Esta sequência de nomes é passada como parâmetros para o *script*.
- R2: Os programas deverão ser compilados exatamente pela ordem indicada na linha de comandos.
- R3: Deve ser emitida uma mensagem de erro se o ficheiro de código-fonte de um programa não estiver acessível no directório de trabalho actual (*working directory*).
- R4: O script deverá informar quantos programas foram compilados sem erros.

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

**FIM**