

## Escolha múltipla para 1º Teste

- 1) Um SO multiprogramado melhora a utilização do CPU porque se pode:  
☒ Executar vários processos no mesmo instante  
☐ Executar vários processos de diferentes utilizadores  
☐ Atribuir o CPU a um outro processo caso o processo corrente tenha de ser suspenso
- 2) Um encravamento (deadlock) pode acontecer quando se tem 2 ou mais processos:  
☐ A tentarem ler ou escrever num mesmo disco  
☒ A tentarem aceder a recursos não interrompíveis
- 3) Um descritor de ficheiros em Unix é representado em C por:  
☒ Um inteiro  
☐ Um apontador para estrutura  
☐ Uma estrutura
- 4) A instrução `fork()` cria um processo filho, suponha então que se executou o seguinte código:  
    `x = 1;`  
    `if(fork()==0)`  
        `x=2;`  
    `else`  
        `x=3;`  
Após a instrução `if_else` podemos dizer que:  
O processo pai tem valor 3 e o processo filho tem valor 2.
- 5) O modo kernel de operações num SO é responsável por:  
☒ Proteger o acesso aos recursos do sistema  
☐ Executar os processos de Sistema  
☐ Executar os processos de utilizador
- 6) Um pedido I/O síncrono:  
☐ Demora menos tempo a ser processado  
☐ Demora mais tempo a ser processado  
☒ Não possibilita a sobreposição de I/O com o uso de CPU
- 7) O uso de DMA tem o efeito de:  
☐ Reduzir o nº de vezes que os dados passam no bus do sistema  
☒ Reduzir a intervenção do SO nas operações de I/O  
☐ Facilitar o acesso directo dos processos à memória da máquina
- 8) A função `wait()` do sistema permite uma forma limitada de comunicação entre processos pai e filho, nomeadamente:  
☐ Permite que o pai espere até que um filho em concreto termine  
☒ Permite que o pai espere até que um filho termine
- 9) A implementação de redireccionamento de output de um comando para um ficheiro requer o uso da função do sistema `dup2` por:  
☐ Associar o descriptor de ficheiro ao descriptor 0 (input)  
☒ Associar o descriptor de ficheiro ao descriptor 1 (output)  
☐ Duplicar o valor do descriptor

- 10) Qual das seguintes não é condição para que se verifique um encravamento (deadlock):
- ☒ Posse exclusiva de um recurso por um processo
  - ☐ Espera por atribuição de um recurso
  - ☐ Cadeia de espera circular de processos e recursos
- 11) Num SO é indispensável proteger o acesso aos recursos ou hardware. Tal é conseguido:
- ☐ Usando-se dois modos distintos de operação: User e Kernel
  - ☐ Usando-se dois tipos ou processos: de Sistema e de Utilizadores
  - ☐ Gerando-se interrupts por operação de I/O
- 12) O que entende por buffering, caching e spooling?
- Buffering permite sobrepor comutações com operações de entrada e saída
- Caching permite armazenar dados que serão utilizados no futuro
- Spooling permite processar tarefas sobrepondo
- 13) Comparando com um pedido I/O síncrono, um pedido assíncrono
- ☐ Demora menos tempo a ser processado
  - ☐ demora mais tempo a ser processado
  - ☒ Assegura maior eficiência do sistema porque sobrepõe I/O e uso de CPU
- 14) Num sistema que suporta multiprogramação e time\_sharing, vários utilizadores podem partilhar o sistema simultaneamente. Se este suporte não for bem implementado, esta situação pode causar vários problemas de segurança. Um possível problema de segurança pode ser:
- ☒ Processos podem invadir a área de memória de outros processos
  - ☐ processos podem executar em modo não protegido
  - ☐ processos podem utilizar a CPU de forma caótica
- 15) Um pedido de I/O sem buffering
- ☐ Demora menos tempo a ser processado
  - ☐ Demora mais tempo a ser processado
  - ☒ Não possibilita a sobreposição de I/O ou o uso de CPU
- 16) A função do sistema kill() serve:
- ☒ Para enviar sinais explícitos a um processo filho
- 17) A função de sistema execl() permite substituir o programa do processo que executa a função por um outro, isto acontece porque:
- ☒ Se modifica o segmento de dados e texto do processo
- 18) A instrução fork() cria um processo filho. Supondo que executou o código abaixo. Podemos dizer que:
- ```
for(i=0; i<2; i++)  
    if (fork()==0)  
        escreve(i);
```
- ☒ Cria 3 processos filhos
- 19) Uma instrução que verifique e modifique uma posição de memória de forma atómica:
- ☒ é fundamental para implementar sincronização entre processos
  - ☐ não é necessária para implementar sincronização entre processos
  - ☐ torna a implementação de sincronização entre processos mais fácil

20) Com a técnica de DMA, numa operação de leitura de dados de um periférico, permite:

- ☒ Ao controlador colocar directamente na memória do computador os dados lidos
- ☐ Ao SO transferir os dados no Buffer do periférico para a memória após sinalização feita pelo controlador
- ☐ Ao processo que efectuou o pedido aceder directamente à memória para transferir os dados que lhe são enviados pelo controlador

21) A função do sistema signal() serve para:

- ☐ Definir uma acção a tomar sobre um tipo concreto de sinal
- ☒ Enviar um sinal a um processo
- ☐ Activar um novo sinal num processo

22) Qual o resultado da execução do trecho de programa abaixo:

```
execpl("ls","s","-la",NULL);  
printf("Terrível");
```

- ☐ O conteúdo do directório corrente e a string "Terrível"
- ☒ O conteúdo do directório corrente
- ☐ Este trecho de programa está errado

23) O que é um PCB?

- ☒ É a estrutura de dados da identificação e estado de um processo
- ☐ É um modelo de hardware usado na interrupção de processos
- ☐ É um modelo de SO responsável pela troca de contexto de processos