

Sistemas Operativos

Primeira Frequência, 19 novembro 2012, 14:00

Docente: Eduardo Marques (emarques@uma.pt)

Nome: _____ Número: _____

Curso: _____

Este exame é **sem consulta**. Acetatos, livros, computadores, calculadoras, PDA's, telemóveis e acesso à Internet **não são permitidos**. Apenas são necessárias esferográficas azuis e/ou pretas.

A duração da frequência é de **90 minutos**, para uma cotação máxima de **20 valores**. Para outras informações sobre a avaliação deve, após a prova, consultar a página da cadeira.

Leia as questões com atenção e responda nas folhas do enunciado. Muita atenção para o tempo despendido em cada uma delas. Quando terminar, entregue as suas respostas ao docente/vigilante, tendo a certeza que preencheu corretamente a sua identificação.

Questão	1	2	3	4	5	6	7	8	Total
Valores	4	1½	1½	2	2	2	2	5	20
Cotação									

Escolha Múltipla

1. (4 val's) Assinale em cada uma das questões seguinte uma opção. A opção correta é apenas uma e é aquela que responde ao pedido completamente. Cada questão correta vale 0,5 valores. Uma questão errada diminui em 0,125 valores a cotação desta secção (se o valor final desta secção for negativo, a sua cotação passará para zero).

- (a) No Sistema Operativo Linux, o comando 'cd projeto', executado a partir da raiz do sistema de ficheiros, terá o mesmo resultado que qual dos seguintes comandos:

- ☐ cd /projeto
- ☐ cd ../projeto
- ☐ cd /home/<utilizador>/projeto
- ☐ cd /usr/projeto
- ☐ nenhum dos anteriores

- (b) A dois processos a aceder a uma variável partilhada, o algoritmo de *Peterson* providencia:

- ☐ exclusão mútua
- ☐ progresso
- ☐ ausência de minguia
- ☐ todos os anteriores
- ☐ nenhum dos anteriores

-
- (c) Quando um processo é criado usando a chamada de sistema `fork()`, qual dos seguintes itens do processo pai NÃO é herdado pelo processo filho?
- ☐ ID do utilizador
 - ☐ ficheiros abertos
 - ☐ ID do processo
 - ☐ espaço de endereçamento do processo
 - ☐ nenhuma das opções anteriores
- (d) A exclusão mútua pode ser implementada com um semáforo, se este for inicializado a:
- ☐ 0
 - ☐ 1
 - ☐ n
 - ☐ a exclusão mútua apenas é garantida com trincos
 - ☐ todas as anteriores
- (e) Na estrutura interna do *Windows 2000* o HAL representa:
- ☐ *Hardware Advert Limit*
 - ☐ *Hardware Abstraction Layer*
 - ☐ *Hardware Abstraction Level*
 - ☐ *Hacking Avoidance Layer*
 - ☐ nenhuma das anteriores
- (f) Um processo bloqueado num semáforo poderá avançar no caso de:
- ☐ o sistema operativo o desbloquear
 - ☐ um outro processo assinalar sobre o semáforo
 - ☐ o utilizador premir a tecla **Esc**
 - ☐ ultrapassar um dado tempo de bloqueio
 - ☐ todas as anteriores
- (g) Qual das seguintes transições de estados de um processo não é possível:
- ☐ bloqueado para executável
 - ☐ execução para executável
 - ☐ execução para bloqueado
 - ☐ executável para execução
 - ☐ nenhuma das opções anteriores
- (h) Qual dos seguintes termos NÃO é uma operação do objeto Processo:
- ☐ Criar
 - ☐ Adormecer
 - ☐ Juntar
 - ☐ Eliminar
 - ☐ nenhuma das anteriores

Conceitos e Definições

Defina cada um dos termos apresentado abaixo, indique o seu contexto e apresente, de forma clara e concisa, as principais diferenças.

2. ($1\frac{1}{2}$ val's) 'fechar' e 'esperar'

3. ($1\frac{1}{2}$ val's) pseudoparalelismo' e 'multiprogramação'

Processos e tarefas

4. (2 val's) O que é uma tarefa e quais são as suas vantagens? Complemente com um exemplo prático que ilustre a sua utilidade.

5. (2 val's) O que é o contexto de um processo?

Gestor de Processos

6. (2 val's) Qual o objetivo do Despacho? Descreva o seu funcionamento.

7. (2 val's) Descreva os comportamentos genéricos dos processos indicados nas aulas. Relacione cada um deles com a política de escalonamento com prioridades.

Sincronização

8. (5 val's) Considere o serviço de refeições de uma cantina. Existem n alunos para almoçar e apenas uma mesa de M lugares ($n \gg M$). As refeições são servidas por turnos, ou seja, os alunos de cada turno têm um tempo definido de refeição. No final do tempo, todos se devem levantar e, apenas após todos saírem, os alunos do turno seguinte se poderão sentar. Um turno apenas se inicia após existirem M alunos à espera.

De seguida são apresentados os códigos dos processos *aluno* e do processo *Mesa*.

```
aluno() {
    INICIAREFEICAO();
    refeicao();
    TERMINAREFEICAO();
}

Mesa() {
    for (;;) {
        SENTAGRUPO();
        temporefeicao();
        SAIDAGRUPO();
    }
}
```

Programe em pseudo-código as seguintes funções que os processos podem invocar: *INICIAREFEICAO*, *TERMINAREFEICAO*, *SENTAGRUPO* e *SAIDAGRUPO*. Utilize semáforos e/ou trincos para coordenar os processos e não se esqueça de indicar também as declarações iniciais das variáveis que utilizar. **Não se preocupe** com os procedimentos *refeicao* e *temporefeicao*; apenas se sabe que são procedimentos executados por ambos os processos, que o *temporefeicao* termina ao fim de um intervalo de tempo finito e que é sempre superior à duração de *refeicao*.