

Exame

U.C. 21111

Sistemas Operativos

16 de Junho de 2011

INSTRUÇÕES

Para a resolução do teste, leia as seguintes informações e instruções, antes de responder

- O enunciado do teste tem 3 páginas, sendo constituído por dois grupos de questões, I e II, com a cotação total de 20 valores.
- Nas respostas, tenha a preocupação de utilizar uma **letra legível** por outra pessoa.
- O grupo I é constituído por questões do tipo resposta aberta, com a cotação total de 12 valores.
- O grupo II é constituída por questões de escrita de Software, com a cotação total de 8 valores.
- As cotações são indicadas nas próprias questões/alíneas.
- Todas as respostas devem ser escritas unicamente com caneta azul ou preta.
- É permitido utilizar máquina de calcular.
- O não cumprimento das instruções implica a anulação das respectivas questões ou do teste.
- O tempo de realização do teste é de 150 minutos.
- Verifique se o teste está completo e termina na palavra **FIM**.

Grupo I

- 1.1.** [1] Uma das principais funções de um SO é apresentar aos programas (aplicações) e programadores uma interface virtual baseada em abstrações, dita máquina extendida. Explique porquê.
- 1.2.** [1] Caracterize resumidamente o conceito de sistema operativo de tempo real e as suas variantes.
- 1.3.** [1] Em que consiste uma função de sistema ? Explique sucintamente como é implementada a chamada a uma função de sistema.
- 1.4.** [1] Explique o conceito de pseudoparalelismo.
- 1.5.** [1] Explique o conceito de processo.
- 1.6.** [1] Explique em que consiste o escalonamento de um processo e descreva quatro situações que possam levar o SO a efectuá-lo.
- 1.7.** [1] Explique os conceitos de recursos preemptíveis e não preemptíveis. Dê um exemplo de cada.
- 1.8.** Considere um sistema com memória virtual e paginação. O espaço de endereçamento virtual é de 22 bits (4MB) e em determinado momento a tabela de páginas (mononível) tem o seguinte conteúdo,

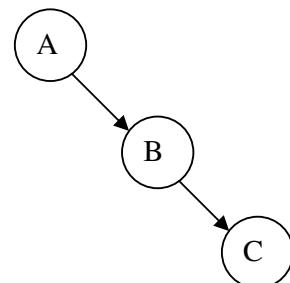
nº entrada (decimal)	conteúdo (binário)	bit presente/ausente (1/0)
1023	11011	0
1022	11010	1
...
4	01001	0
3	01000	1
2	00100	0
1	01011	1
0	00000	1

- 1.8.1.**[0,7] Considere as dimensões da página virtual e da moldura de página (page frame) iguais. Indique justificando a dimensão da página utilizada e do espaço de endereçamento físico.
- 1.8.2.**[0,7] Considere o endereço virtual 00 0000 0011 0101 0111 1111. Indique justificando o endereço físico correspondente.
- 1.8.3.**[0,6] Considere o endereço virtual 11 1111 1111 1111 0100 1101. Explique sucintamente o que acontece quando o programa efectua uma referência a este endereço.

- 1.9.** [1] Explique o significado de directória de trabalho (ou corrente) e relacione-o com o conceito de “pathname” relativa.
- 1.10.** [1] Para um sistema de ficheiros baseado em i-nodes, explique no que consiste e como é implementado um “hard link”.
- 1.11.** [1] Explique o conceito de I/O por mapeamento de memória.

Grupo II

- 2.1.** [3] Escreva um programa em linguagem C padrão que crie uma hierarquia de processos em árvore como mostra a figura. Cada processo deve imprimir na saída padrão uma mensagem do tipo “Processo X”, onde X significa A, B ou C conforme o caso. Cada processo deve esperar que os seus processos filhos (directos) terminem, antes dele próprio terminar. O programa não necessita de testar erros nas chamadas às funções.



- 2.2.1.** [4.5] Escreva um programa multitarefa em linguagem C segundo a norma POSIX que transfira por blocos os elementos de um vector `int x[1000]` para um vector `int y[1000]`. Os vectores constituem variáveis globais ao programa e admite-se que foram devidamente inicializados.

A tarefa principal deve criar 10 sub-tarefas em que cada uma em paralelo (ou em pseudo-paralelismo) copia um bloco de 100 elementos do vector `x[]` para o vector `y[]`. O bloco origem em `x[]` tem início no elemento de menor índice ainda não transferido. O bloco destino em `y[]` tem início no elemento de menor índice ainda não preenchido. A tarefa principal deve esperar que todas as sub-tarefas terminem.

Nota 1: com a estratégia indicada, a ordem dos elementos em `y[]` será diferente (por blocos) da ordem em `x[]`, dependendo da ordem do escalonamento das tarefas.

Nota 2: planeie cuidadosamente como é dividido o trabalho entre as sub-tarefas e como é efectuada a sincronização e a comunicação da informação necessária à resolução do problema entre todas as tarefas.

- 2.2.2.** [0.5] Considere que o programa da alínea anterior está num ficheiro de nome `transf.c`. Indique o comando Linux necessário à sua compilação e criação de um ficheiro executável.

FIM