UNIVERSIDADE ABERTA
www.uab.pt

Ministério da Educação e Ciência

U.C. 21111

Sistemas Operativos
[Exame da Época de Recurso]

5 de setembro de 2011 *[<= o ano aqui tem uma gralha: esta prova foi realizada em 5 de setembro de **2012** e não 2011]*

*[Nas págs. 3 e 4 aparece a **resolução** dada pelo Prof. José Coelho em 22-set-2012]*

-- INSTRUÇÕES --

- O tempo de duração da prova de exame é de 2 horas, acrescida de 30 minutos de tolerância.
- O estudante deverá responder à prova na folha de ponto e preencher o cabeçalho e todos os espaços reservados à sua identificação, com letra legível.
- Verifique no momento da entrega da(s) folha(s) de ponto se todas as páginas estão rubricadas pelo vigilante.
- Exclui-se, para efeitos de classificação, toda e qualquer resposta apresentada em folhas de rascunho.
- Os telemóveis deverão ser desligados durante toda a prova e os objetos pessoais deixados em local próprio da sala de exame.
- A prova é constituída por **8** perguntas, e termina com a palavra **FIM**. Verifique o seu exemplar e, caso encontre alguma anomalia, dirija-se ao professor vigilante nos primeiros 15 minutos da mesma, pois qualquer reclamação sobre defeito(s) de formatação e/ou de impressão que dificultem a leitura não será aceite depois deste período.
- Utilize unicamente tinta azul ou preta. As respostas na folha de ponto podem ter as perguntas por qualquer ordem, mas tem de identificar o número da pergunta.
- A cotação é indicada junto de cada pergunta.
- A interpretação dos enunciados das perguntas também faz parte da sua resolução, pelo que, se existir alguma ambiguidade, deve indicar claramente como foi resolvida.

Pergunta 1 (2 valores)
Indique o que é uma instrução TRAP, e qual a sua utilidade nos sistemas operativos.

Pergunta 2 (2 valores)

Explique o que é e para que serve um mutex. Exemplifique uma situação em que seja necessário.

Pergunta 3 (2 valores)

Descreva e compare os algoritmos de substituição de páginas: LRU e Aging.

Pergunta 4 (2 valores)

Suponha que a informação sobre os blocos livres do disco foi completamente perdida devido a um crash. Existe alguma maneira de recuperar esta informação? Discuta esta questão para o sistema de ficheiros UNIX e FAT-16.

Pergunta 5 (2 valores)

Compare RAID 0 até RAID 5, relativamente à performance de leitura/escrita.

Pergunta 6 (2 valores)

Descreva o algoritmo do banqueiro para um recurso.

Pergunta 7 (4 valores)

Suponha que tem uma função "Trabalho" com um argumento que pode variar de 0 a N-1, indicando o trabalho a realizar. Escreva um programa que crie N processos em paralelo, a partir de um só processo, e em cada um chame a função Trabalho com argumento **i** (sendo **i** o número de processo a variar entre 0 e N-1). O processo principal deve terminar quando todos os processos criados terminarem.

Pergunta 8 (4 valores)

Escreva um programa multitarefa em linguagem C segundo a norma POSIX que permite ao utilizador introduzir números, um número por linha. Para cada número N, cria uma tarefa para obter o número de divisores, testando de 2 até N/2. Após a tarefa calcular o valor, soma o número de divisores encontrados aos valores já calculados e guardados numa variável global (atenção, podem haver outras tarefas em paralelo) e imprime o valor da variável global na consola, terminando. Caso o utilizador introduza o valor 0, a tarefa principal deve aguardar que todas as tarefas terminem, e terminar.

FIM

Resolução (critérios de correção / pontos esperados / comentários) indicada pelo Prof. José Coelho em 22-set 2012:

- P1 (secção 1.6, figura 1-17)
  - é uma instrução que permite passar do modo de utilizador para o modo kernel
  - no modo kernel existem mais privilégios, com a troca de modos sempre que uma função do sistema operativo é chamada, garante-se que apenas código do sistema operativo corre em modo kernel
- P2 (secção 2.3.6)
  - serve para guardar uma região crítica (à entrada lock, à saída unlock), quando existem várias threads (tarefas), de modo a garantir que apenas uma tarefa está em cada momento na região crítica
  - quando várias tarefas têm de ler e alterar o valor de uma mesma variável, tanto o acesso como a edição da variável, têm de estar protegidas por um mutex
  - os mutexes não são utilizados para fazer exclusão mútua entre processos, apenas no mesmo processo entre várias threads (tarefas) desse processo. Penalização de 0,5 para quem referiu processos em vez de threads (tarefas).
- P3 (secção 3.4.6 e 3.4.7)
  - LRU (least recently used), remove a página que não foi utilizada à mais tempo.
  - O LRU é o ideal, mas o problema é manter uma lista atualizada das páginas utilizadas por ordem, de modo a identificar rapidamente a que não foi utilizada à mais tempo.
  - O aging é uma forma de simular o LRU por software, embora seja diferente
  - Por cada página é guardado um registo. Sempre que há uma interrupção de clock, os registos de cada página "envelhecem" fazendo um shift para a direita, e as páginas que foram referenciadas no período do clock, recebem o bit mais à esquerda a 1. Quando há uma "page fault", é escolhida a página com o menor valor, significando que foi a menos referenciada nos últimos instantes.
  - O algoritmo aging, trata páginas a zero como sendo iguais, apenas significando que, desde a última vez que foram acedidas, já passaram mais de X interrupções de clock, sendo X o número de bits do registo. Essas páginas são tratadas de igual forma, e também não destingue acessos entre interrupções de clock, pelo que, em caso de empate pode-se não estar a libertar a página acedida à mais tempo.
  - Resposta relacionada com a matéria: 0,5 valores
  - Descrição do LRU e Aging parecida (embora incompleta e algumas partes no sentido contrário dos algoritmos): 1 valor
- P4 (secção 4.3.2 e 4.4.1)
  - Em ambos os sistemas tem que se inicializar os blocos livres para todo o disco, e identificar os blocos utilizados em ficheiros e directorias, a partir da directoria raiz.
  - Os ficheiros relativos às directorias têm de ser abertos para ver o seu conteúdo.
  - Os blocos ocupados por ficheiros, no sistema FAT não forçam qualquer leitura em disco, já que a lista de blocos utilizados em cada ficheiro está no FAT (ver fig. 4-12), que se encontra em memória.
  - No UNIX cada ficheiro tem um i-node com os blocos onde se encontra o ficheiro (e pode ter mais que um), pelo que o i-node tem de ser lido do disco, tendo portanto mais acessos ao disco.
  - Não é suficiente consultar apenas a tabela FAT, já que assim não se sabe onde começam os ficheiros, nem quais os blocos não utilizados.
  - Nesta questão era irrelevante os utilitários que se poderiam utilizar, que seriam interessantes na UC de Administração de Sistemas Informáticos.

- P5 (secção 5.4, figura 5-20)
    - RAID 0 - divide os blocos pelos discos, blocos consecutivos ficam em discos distintos, para assim um ficheiro em vários blocos seguidos, seja lido/escrito utilizando todos os discos em paralelo. Maximiza a velocidade de leitura/escrita, mas não tem redundância, pelo que quantos mais discos, maior é a probabilidade de um deles falhar.
    - RAID 1 - idêntico ao RAID 0, mas com metade dos discos com uma cópia redundante dos outros discos. Devido a metade dos discos serem utilizados, acaba por ter metade da performance na escrita que o RAID 0, mas na leitura tem uma performance idêntica já que qualquer um dos discos pode ser utilizado. Se um disco falhar, existe sempre uma cópia do outro disco.
    - RAID 2 - em vez de um bloco é colocado em cada disco um bit e o resultado é a concatenação dos bits lidos de cada disco, que têm de ler as mesmas posições em simultâneo, para reconstruir a informação. Neste caso um erro de um disco pode ser reconstruído dado que são guardados bits a mais (Hamming code). A velocidade de leitura/escrita é dependente do número de discos, e como são todos utilizados, tem-se a eficiência máxima, mas normalmente são necessários muitos discos, todos iguais, e têm de estar sincronizados na leitura.
    - RAID 3 - igual ao RAID 2, mas apenas um disco para um bit de paridade (permite apenas que um disco falhe). Como os discos têm de estar sincronizados, o seek time de todos os dicos é igual a um só disco, pelo que se existirem muitos ficheiros ou pedidos de blocos em locais distintos, o tempo de leitura é idêntico a um só disco (já que é mais o tempo de seek que o tempo de leitura do bloco), sendo apenas rápido no caso dos blocos estarem seguidos.
    - RAID 4 - igual a RAID 0, mas um dos discos fica com um bloco que é um OU-exclusivo dos restantes blocos no disco, para assim ficar com redundância. A leitura é idêntica ao RAID 0 (apenas tem menos um disco), mas a escrita é mais lenta, já que é necessário atualizar o disco com o OU-exclusivo.
    - RAID 5 - idêntico ao RAID 4 mas não existe um disco com os blocos com o OU-exclusivo, esses blocos são repartidos por todos os discos, para balancear a carga.
    - Informação genérica sobre RAID: 0,5 valores
- P6 (secção 6.5.3, figura 6-11)
    - Cada processo tem de especificar o volume de unidades máximo que pode precisar.
    - Existe uma quantidade de recursos disponíveis.
    - A cada momento, se forem pedidos mais recursos, estes são fornecidos excepto se o estado ficar inseguro.
    - Um estado seguro significa que pelo menos um processo, mesmo que peça o volume máximo de unidades especificado, existem ainda recursos disponíveis para que esse processo acabe e liberte os recursos que tem no momento, e de seguida um segundo processo, e assim sucessivamente, sendo possível evitar deadlocks.
    - Um estado inseguro, significa que, se todos os processos pedirem recursos ao mesmo tempo, não existe nenhum do pedidos que possa ser satisfeito, e assim fica-se num deadlock.
- P7 (programa com processos)
    - Utilização do fork com condicional: 1 valor
    - Criação de mais que dois processos mas não N processos: 2 valores
    - Número de processos corretos: nenhum trabalho
- P8 (programa com tarefas)
    - Tentativa com processos: 0,5 valores
    - Tentativa com threads e utilização de um mutex: 1 valor