

Nome: _____ RA: _____

Questão 1 (2 pontos)

- a) Explique porque servidores em geral são mais eficientes quando implementados com threads.

São mais eficientes porque permitem o servidor atender várias requisições concorrentemente, evitando o bloqueio do servidor quando, para atender uma requisição, o servidor executa uma operação de E/S (neste caso apenas a thread associada à requisição bloqueia).

- b) Descreva com o auxílio de pseudo-código as operações **down** e **up** para semáforos. Suponha que o código destas operações é executado pelo sistema operacional com interrupções desabilitadas. Comente cada instrução do seu pseudo-código.

```
down(s)
s.valor--;
if(s.valor < 0) {
    adiciona(s.ListaBloqueados, esteProcesso); // adiciona processo no final da lista
    suspend(esteProcesso); // bloqueia processo
}

up(s)
s.valor++;
if(s.valor <= 0) {
    processo = remove(s.ListaBloqueados); // remove processo do topo da lista
    wakeup(processo); // desbloqueia processo
}
```

Questão 2 (2 pontos) – Descreva o funcionamento de um algoritmo de escalonamento de processos apropriado para sistemas operacionais de uso geral (onde a maioria das aplicações são interativas). Justifique porque o algoritmo descrito é apropriado para o escalonamento de aplicações interativas.

O melhor algoritmo para esta classe de aplicações é o de filas múltiplas (ver descrição no livro texto), pois este algoritmo eleva a prioridade de processos bloqueados em E/S fazendo com que retomem a CPU assim que se tornem prontos (o usuário percebe uma resposta rápida do sistema).

Questão 3 (2 pontos) - Suponha que a CPU inicia a execução de uma instrução cujo acesso à memória irá gerar uma falta de paginação (page fault). Descreva a sequência

temporal de eventos desde a detecção da falta de paginação até a instrução iniciada ser executada por completo.

1. *Página ausente na TLB.*
2. *SO é interrompido pela CPU (page fault).*
3. *CPU identifica a interrupção e acessa a tabela de páginas do processo.*
4. *Se a página está mapeada na tabela, atualiza TLB e reinicia a execução do processo.*
5. *Caso contrário (página não mapeada):*
 - a) *Bloqueia o processo.*
 - b) *Aloca uma página livre e programa DMA para trazer página da área de swap (ou do executável).*

Questão 4 (2 pontos) - Explique porque o algoritmo de troca de páginas “Menos Recentemente Usada” (LRU – Least Recently Used) é eficiente mas de difícil realização. Descreva uma alternativa mais prática ao LRU.

LRU exige a manutenção do tempo exato que a página foi acessada, o que demanda uma leitura e gravação do relógio na tabela de páginas a cada acesso à memória. Uma alternativa é o algoritmo do relógio ou de envelhecimento (ver descrição no livro texto).

Questão 5 (2 pontos) - As perguntas abaixo exigem que o aluno estabeleça certas hipóteses que são parte das respostas.

- a) Onde os programas que listam informações detalhadas sobre arquivos (tamanho, permissões, data de criação, etc.) buscam tais informações ?
Considerando FAT, na entrada do diretório. Considerando inodes, no próprio inode.

- b) Quantos acessos a disco são necessários para ler o primeiro byte do arquivo /usr/local/arq1 ?
Considerando que todo o diretório cabe em um único bloco, um sistema de arquivos FAT requer 3 acessos (bloco com diretório usr, bloco com diretório local, e primeiro bloco do arquivo arq1). Considerando um sistema de arquivos baseado em inodes e apenas o inode do diretório raiz já em memória, são necessários 7 acessos (bloco do diretório raiz, inode do diretório usr, bloco do diretório usr, inode do diretório local, bloco do diretório local, inode do arquivo arq1 e primeiro bloco do arquivo arq1).

- c) Um sistema de arquivos possui blocos de 1 Kbytes. Qual o tamanho máximo de arquivo que este sistema comporta ?
Considerando FAT com n bits para endereçar blocos, 2^n Kbytes. Considerando inodes, ver, por exemplo, questão 4 da prova do ano anterior.