

UFES - DEPARTAMENTO DE INFORMÁTICA

1ª. Prova de Sistemas Operacionais / Sistemas de Programação II

Período: 2006/1 – Data: 31/10/2006

Profa. Roberta Lima Gomes – Prof. José Gonçalves

ALUNO: _____

1) Considere um sistema que deva executar diversos tipos de processos, incluindo: interativos, em lotes, com uso intensivo da UCP, com uso intensivo de E/S, grandes, pequenos, etc. Considere ainda os métodos de escalonamento vistos em sala de aula (FIFO, SJF, SRTF, RR, prioridades, múltiplas filas, etc.). Dada a lista de requisitos a seguir, qual é a estratégia que melhor satisfaz a todos eles simultaneamente? Para cada uma das demais estratégias, explique porque pelo menos um requisito que não foi satisfeito. **(2,0)**

- a) O número de processos interativos é relativamente pequeno. Eles deverão ter alta prioridade, mas são aceitáveis pequenos atrasos.
- b) Processos com uso intensivo de E/S devem ter prioridade alta para manter ativos os processadores de E/S.
- c) Processos com uso intensivo da UCP devem retardar processos com uso intensivo de E/S, mesmo que esses já estejam esperando há muito tempo.
- d) Quando os processos com uso intensivo da UCP são os únicos que estão prontos, a sobrecarga do sistema operacional deve ser minimizada.

2) Compare e contraste a manipulação de chamadas ao supervisor (SVCs) e interrupções em um sistema operacional. **(2,0)**

3) Assumindo que todas as operações *fork()* retornarão com sucesso, quantos processos serão criados quando o seguinte programa executado? O que será impresso? **(2,0)**

```
main()
{
    int i=1;
    int ret_val;

    while(i <= 3)
    {
        if ((ret_val = fork()) == 0) {
            printf("In child %d. \n", i);
            i = i + 1;
        } else {
            printf("In parent %d. \n", i);
            exit(0);
        }
    }
}
```

4) Considere o problema dos leitores e escritores, onde existem diversos processos que eventualmente fazem acessos de leitura a uma base de dados e diversos processos que eventualmente fazem acessos de escrita à mesma base. Vários acessos de leitura podem ocorrer simultaneamente, mas um acesso de escrita não pode ocorrer simultaneamente com

acessos de nenhum tipo. Considere o código a seguir para os processos de leitura e de escrita. Suponha que todos os semáforos são iniciados com valor 1: **(2,0)**

```
Leitor:                               Escritor:
...                                   ...
1 while(1) {                           1 while(1) {
2 P(R);                                2 ...produz
3 P(M);                                3 P(R);
4 rc++;                                 4 P(W);
5 if (rc==1) P(W);                     5 ESCREVE;
6 V(M);                                6 V(W);
7 V(R);                                7 V(R);
8 LE;                                   8 }
9 P(M);
10 rc--;
11 if (rc==0) V(W);
12 V(M);
13 ... consome
```

(a) Essa solução pode levar a *starvation* de escritores? (pode acontecer de um escritor nunca conseguir o acesso à base devido à seguida chegada de novos leitores?) Explique sua resposta, usando os números das linhas de código para se referir aos passos do programa.

(b) Explique o papel do semáforo M. Dê um exemplo de problema que poderia ocorrer caso as operações sobre ele fossem retiradas.

5) Suponha que um grupo de N canibais come jantares a partir de uma grande travessa que comporta M porções. Quando alguém quer comer, ele(ela) se serve da travessa, a menos que ela esteja vazia. Se a travessa está vazia, o canibal acorda o cozinheiro e espera até que o cozinheiro coloque mais M porções na travessa. Utilizando semáforos, desenvolva o código para sincronizar as ações dos canibais e do cozinheiro. A solução deve evitar *deadlock* e acordar o cozinheiro apenas quando a travessa estiver vazia. Suponha um longo jantar, onde cada canibal continuamente se serve e come, sem se preocupar com as demais na vida de um canibal... **(2,0)**