



ALUNO: Dênis Rainer

1) (2,0 Pontos) Sobre conceitos básicos de sistemas operacionais, responda:

- 1,0 a) Cite três tipos de sistemas operacionais e cite um exemplo de cada um deles.
b) O que é o núcleo (*kernel*) do sistema operacional e quais as suas funções principais?

2) (2,0 Pontos) Sobre processos e threads, responda:

- 1,0 a) Um sistema possui três processos multithreads que utilizam threads de usuário. Onde são armazenados o PCB e os TCBs?
b) É conveniente utilizar threads em duas tarefas com característica CPU *bound* em uma máquina com apenas um núcleo? Justifique.

3) (2,0 Pontos) Sobre deadlock, responda:

- 2,0 a) O fato de um estado ser inseguro implica necessariamente que o sistema sofrerá deadlock? Justifique.
b) Como a utilização de *spool* pode evitar deadlocks?

2,0 4) (2,0 Pontos) Considere um sistema operacional que implemente escalonamento round-robin com fatia de tempo igual a 3 u.t. Em um determinado instante de tempo, existem apenas três processos (P1, P2 e P3) na fila de prontos, e o tempo de CPU de cada processo é 14, 6 e 10 u.t, respectivamente. Qual o estado de cada processo no instante de tempo T, considerando a execução dos processos P1, P2 e P3, nesta ordem, e que apenas o processo P1 realiza operações de E/S? Cada operação de E/S é executada após 5 u.t. e consome 10 u.t. Desconsidere o tempo de troca de contexto.

a) T = 18 u.t.

b) T = 25 u.t.

X 5) (2,0 Pontos) O problema dos leitores e escritores consiste em permitir que vários processos acessem simultaneamente a base dados para leitura, mas caso um processo esteja escrevendo na base de dados, nenhum outro processo poderá acessá-la nem para leitura nem para escrita. Os algoritmos abaixo propõem uma solução para o problema dos leitores e escritores. Existe algum erro nos algoritmos? Se existir, explique detalhadamente os problemas ocasionados por este(s) erro(s) e em seguida faça as devidas correções.

SEMAPHORE DB = 1; SEMAPHORE MUTEX = 1; INT RC = 0;

THREAD LEITOR:

```
WHILE (TRUE) {  
    DOWN(MUTEX);  
    RC = RC + 1;  
    UP(MUTEX);  
    DOWN(DB);  
    READ_DATA_BASE();  
    DOWN(MUTEX);  
    RC = RC - 1;  
    UP(DB);  
    UP(MUTEX);  
    USE_DATA_READ();  
}
```

THREAD ESCRITOR:

```
WHILE (TRUE) {  
    THINK_UP_DATA();  
    DOWN(DB);  
    WRITE_DATA_BASE();  
    UP(DB);  
}
```

BOA PROVA!

③ a) Não. O fato do estado ser inseguro quer dizer que há a possibilidade de deadlock, caso os recursos requisitados pelos processos sejam dados a estes processos.

b) Com a técnica de spool apenas um processo requisitará recurso, por exemplo uso da impressora, assim não haverá deadlock, pois não há deadlock com apenas um processo.

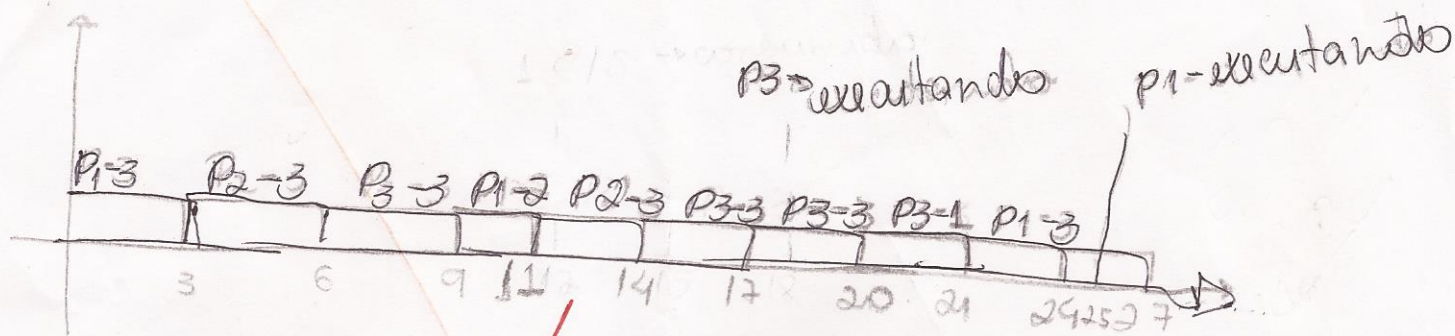
① b) O kernel ou núcleo é a parte de gerenciamento do S.O. Este tem a função de atender ou não as requisições dos usuários, usando o manter a segurança do sistema como um todo. Suas principais funções são: reembolso, fechar ou deletar ou criar processos, gerenciar o uso de recursos.

a) ~~Sistema de uso geral: windows~~

~~Sistema crítico: sistema utilizado em aviões~~

② b) Não, pois devido a limitação ser pela CPU não há tanta necessidade de trocas de contextos desnecessárias ou longas esperas por I/O. Como há apenas uma CPU as trocas de contexto que existirem não poderão ser evitadas, pois há apenas um núcleo.

④ $T_1 = 14$ $T_2 = 6$ $T_3 = 10$



a) P1 - blo queado (aguardará mais 3 u.t. p/ estar pronto)
 P2 - finalizado
 P3 - executando

b) P1 - executando
 P2 - finalizado
 P3 - finalizado

⑤ Não é possível ler simultaneamente, pois cada leitor entra busca os dados e sai, liberando novamente a base de dados para os escritores. É necessário garantir que só depois de todos os leitores terem entrado e saído da sala esta seja liberada para os escritores.

Garante-se isso fazendo, após `down(DB)`, no thread leitor, `if (R > 1)`

~~`DOWN(MUTEX)`
`READ-DATA-BASE()`
`UP(MUTEX)`
`DOWN(MUTEX)`
`RC = 0`~~