

DISCIPLINA: Sistemas Operacionais – DCC403

Prazo de Entrega: 29/05/2018

ALUNO(A): TALLER BEZERRA DE ASSUNÇÃO

[Questão-1] Utilizando o simulador SOSim (disponível em <http://www.training.com.br/sosim>) apresente os resultados do simulador e uma análise para cada item abaixo.

**(PRÁTICA - A)**

Questão teórica para responder com a ajuda do simulador. Observe que em alguns momentos existem processos no estado de pronto, porém nenhum em estado de execução. Explique o porquê dessa situação.

Este momento ocorre porque o escalonador de processos está encerrando a execução de um processo, mudando seu estado para pronto, o enviando para o final da fila de prontos e carregando o próximo processo para ser executado pela CPU. Na simulação podemos ver que os dois processos formam a fila de prontos, o primeiro e carregado, executado e mandado para o final da fila, para que o próximo processo possa ser utilizado pela CPU.

**(PRÁTICA - B)**

Questões teóricas para responder com a ajuda do simulador

- **Por que o problema do starvation pode ocorrer?**

Como os dois processos querem utilizar a CPU, e um processo tem maior prioridade que o outro, o de menor prioridade pode nunca ser escolhido para executar, pois o sistema operacional irá escolher sempre o de maior prioridade quando a CPU estiver disponível.

- **Cite duas ações que o administrador do sistema pode realizar quando é identificada a situação de starvation em um processo?**

Uma solução seria utilizar uma fila de pedido de recurso, assim, quando um processo deseja utilizar um recurso, um pedido associado a ele é colocado no final da fila. Quando o recurso for liberado, o sistema operacional seleciona o processo de acordo com a fila de pedido de recurso. Outra solução seria suspender temporariamente o processo de maior prioridade, para que os de menor possam ser executados.

[Questão-2] Com relação ao problema de Deadlock. Pesquisa e descreva o algoritmo do banqueiro (criado por Dijkstra) que pode ser utilizado para evitar impasses. Sempre que recursos são solicitados, o algoritmo avalia se atender à solicitação levará a um estado inseguro e se isso ocorrer, ela não é atendida. Adicionalmente, escreva o algoritmo do banqueiro em C/C++ e apresente alguns exemplos de sua execução.

O algoritmo do banqueiro pode ser entendido como um banqueiro, que possui alguns clientes que estão pedindo um empréstimo, mas o banqueiro tem dinheiro limitado para emprestar, então ele



busca distribuir esse dinheiro da melhor forma, analisando para quem ele deve emprestar para que todos tenham a quantidade desejada, sem que ele fique zerado e ninguém fique sem a quantidade total requerida.

Seria um algoritmo que distribui recursos compartilhados entre processos, os processos desejam uma quantidade de recursos para completar seu processamento, e assim que conseguem e finalizam, esses recursos são liberados e podem ser utilizados por outros processos, e se ocorre uma situação em que os recursos disponíveis não são suficientes para nenhum processo, é alertado que o estado é inseguro e a requisição não executada, evitando um deadlock.

O algoritmo do banqueiro está disponível na pasta “banqueiro”, onde contêm o código fonte e exemplos de implementação.

Desenvolvido no sistema operacional Linux Ubuntu, para compilá-lo, basta através do terminal Linux, navegar até a pasta onde os dois arquivos, “banqueiro.c” e “banqueiro.h”, se encontram e passar o comando:

```
gcc banqueiro.c -o banqueiro
```

Para executar, passar em seguida o comando:

```
./ banqueiro
```

**[Questão-3] Com relação a problemas clássicos de comunicação entre processos. Escreva o algoritmo do barbeiro (visto em sala de aula), usando threads, em C/C++ e apresente alguns exemplos de sua execução.**

O algoritmo do barbeiro está disponível na pasta “barbeiro”, onde contêm o código fonte e exemplos de implementação.

Desenvolvido no sistema operacional Linux Ubuntu, para compilá-lo, basta através do terminal Linux, navegar até a pasta onde o código se encontra e passar o comando:

```
gcc -pthread barbeiro.c -o barbeiro
```

Para executar, passar em seguida o comando:

```
./barbeiro
```

