
**CONVERSA INICIAL**

Olá, seja bem-vindo a terceira aula de Sistemas Operacionais!

Hoje, vamos explorar o conceito de concorrência, fundamental para os sistemas multiprogramáveis, e as principais técnicas que permitem a implantação desse conceito. Vamos entender o conceito de processo, subprocessos, *threads* e sua aplicação.

**CONTEXTUALIZANDO**

O conceito de concorrência é o princípio básico para o projeto e a implementação dos sistemas operacionais multiprogramáveis.

Nos sistemas multiprogramáveis, vários programas podem estar residentes em memória, concorrendo pela utilização do processador. Quando uma aplicação solicita uma operação de entrada e saída, outra aplicação pode entrar em execução. Essa técnica diminui a ociosidade da CPU e torna o uso da memória mais eficiente.

Nos sistemas que possuem múltiplas unidades de processamento, seja fisicamente, ou núcleo que simulam mais de uma unidade (*dual-core*, *quad-core*) a concorrência se torna fundamental para garantir o compartilhamento de recursos e um bom aproveitamento da capacidade de processamento.
As técnicas de concorrência foram sendo aperfeiçoadas com o objetivo de permitir um alto grau de compartilhamento e a otimização dos recursos computacionais.

**PESQUISE**

A técnica de *spooling* está presente na maioria dos sistemas operacionais, sendo utilizada no gerenciamento de impressão. Quando um comando de impressão é executado, as informações são gravadas em uma área de disco, denominada arquivo de *spool*, liberando a CPU para outras tarefas enquanto é realizada a operação de entrada e saída (I/O – imput/output).

Pesquise mais sobre o funcionamento dos arquivos de *spool* em impressoras modernas com capacidade de armazenamento interno.
http://mrmsistemas.com.br/mendes/so/InterrupcaoExcecao.pdf

**Conceito de processo**

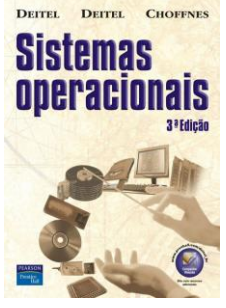
O conceito de processo foi introduzido para facilitar a multiprogramação, cada programa a ser executado está associado a um ou mais processos: um mesmo programa pode ter vários processos concorrendo entre si e compartilhando recursos do sistema computacional. Cada processo possui o seu endereçamento de memória, suas informações do contexto de *software* e do contexto de *hardware*, simulando uma CPU exclusiva para cada um.

Os processos se alternam no uso da CPU fazendo a troca de contexto: hora em espera, pronto ou em execução. Para aumentar a performance das aplicações ou a função específica de uma aplicação, foi introduzido o conceito de *threads* que também são conhecidas como processos leves.

A grande vantagem em relação aos processos é que as *threads* compartilham o mesmo endereço de memória das demais *threads* do processo e o mesmo contexto de *hardware*, isso facilita a comunicação entre as *threads*, aumenta o grau de compartilhamento de recursos e a velocidade de troca de contexto.

Para utilização do conceito de *threads*, o sistema operacional e a linguagem de programação adotada deverão dar suporte a sua implementação.

Aproveite e consulte também, o livro "Sistemas Operacionais", disponível na Biblioteca Virtual, no portal Único.



**TROCANDO IDEIAS**

Chegou a hora de participar do fórum da nossa aula!

- Por que as *threads* são tão importantes em arquiteturas multiprocessadas?
- Como uma aplicação pode implementar concorrência sem utilizar o conceito de *threads*?

Responda à essas questões, também aproveite e comente a postagem dos seus colegas!

**NA PRÁTICA**

É responsabilidade do programador, no seu projeto de *software*, definir quando utilizar o recurso de *threads* e para que tipo de função, normalmente em situações em que é necessário rodar rotinas simultaneamente da mesma aplicação, as *threads* podem ser uma boa alternativa.

Acesse as classes da linguagem JAVA, clicando no ícone abaixo, e verifique que a ferramenta oferece suporte às *threads*. Pesquise também outras linguagens que permitem a utilização de *threads*.

http://docs.oracle.com/javase/6/docs/api/

**SÍNTESE**

Hoje, pudemos entender o conceito de concorrência e as diferentes técnicas que permitem sua implementação. Vimos que o aumento da concorrência e o compartilhamento de recursos tem soluções de *software*, como exemplo o *boffering* e o *spooling*; ou soluções de *hardware*, como os controladores.

Em um ambiente multiprogramável, um programa a ser executado deve estar sempre associado a um processo. Esse processo é formado pelo contexto de *software* e o contexto de *hardware*.

Também conhecemos o conceito de *threads* ou processos leves, que tem como principal vantagem em relação aos processos uma maior capacidade de compartilhamento.

**COMPARTILHANDO**

Acesse o gerenciador de tarefas do Windows (CTRL + ALT + DEL) e observe na guia de processos, quais são os processos em execução na sua máquina.

Observe que habilitando "mostrar todos os processos de usuários", poderá visualizar os processos do usuário que você logou na máquina e os processos do sistema. Abra o terminal do *prompt* de comando e utilize o comando *tasklist,* verifique como os processos são listados e quais as informações do contexto de *software* são exibidas.

Depois dessas experiências, compartilhe com seus colegas e pessoas interessadas o que você aprendeu hoje.
Até a próxima!