Sistemas de Informação

Disciplina: Sistemas Operacionais

Professor: Luciano Coelho

Aluno: Gabriel do Nascimento Pires

Jantar do Filósofos

Thread:

Thread é um pequeno programa que trabalha como um subsistema, sendo uma forma de um processo se autodividir em duas ou mais tarefas. É o termo em inglês para Linha ou Encadeamento de Execução. Essas tarefas múltiplas podem ser executadas simultaneamente para rodar mais rápido do que um programa em um único bloco ou praticamente juntas, mas que são tão rápidas que parecem estar trabalhando em conjunto ao mesmo tempo.

Primitivas Atómicas(Semáforos)

As operações de incrementar e decrementar devem ser operações atômicas, ou indivisíveis, ou seja, enquanto um processo estiver executando uma dessas duas operações, nenhum outro processo pode executar outra operação sob o mesmo semáforo, devendo esperar que o primeiro processo encerre a sua operação sob o semáforo. Essa obrigação evita condições de disputa entre vários processos. O valor de um semáforo indica quantos processos (ou *threads*) podem ter acesso a um recurso compartilhado.

Exclusão Mutua(Mutex)

Exclusão mútua (também conhecida pelo acrônimo mutex para *mutual exclusion*, o termo em inglês) é uma técnica usada em programação concorrente para evitar que dois processos ou *threads* tenham acesso simultaneamente a um recurso compartilhado, acesso esse denominado por seção crítica.

Um meio simples para exclusão mútua é a utilização de um semáforo binário, isto é, que só pode assumir dois valores distintos, 0 e 1. O travamento por semáforo deve ser feito antes de utilizar o recurso, e após o uso o recurso deve ser liberado. Enquanto o recurso estiver em uso, qualquer outro processo que o utilize deve esperar a liberação.

Escalonadores

O escalonamento de processos ou agendador de tarefas (em inglês *scheduling*) é uma atividade organizacional feita pelo escalonador (*scheduler*) da CPU ou de um sistema distribuído, possibilitando executar os processos mais viáveis e concorrentes, priorizando determinados tipos de processos, como os de I/O Bound e os CPU Bound.

O escalonador de processos de 2 níveis escolhe o processo que tem mais prioridade e menos tempo e coloca-o na memória principal, ficando os outros alocados em disco; com essa execução o processador evita ficar ocioso.

Escalonador de Processos escolhe o processo que será executado pela CPU; O escalonamento é realizado com o auxílio do hardware; O escalonador deve se preocupar com a eficiência da CPU, pois o chaveamento de processos é complexo e custoso: Ele afeta desempenho do sistema e satisfação do usuário; O escalonador de processo é um processo que deve ser executado quando da mudança de contexto (troca de processo);

Deadlocks

Deadlock (interbloqueio, blocagem, impasse), no contexto de sistemas operacionais (SO), refere-se a uma situação em que ocorre um impasse, e dois ou mais processos ficam impedidos de continuar suas execuções - ou seja, ficam bloqueados, esperando uns pelos outros. Trata-se de um problema bastante estudado em sistemas operacionais e banco de dados, pois é inerente à própria natureza desses sistemas.

O *deadlock* ocorre com um conjunto de processos e recursos não-preemptíveis, onde um ou mais processos desse conjunto está aguardando a liberação de um recurso por um outro processo, o qual, por sua vez, aguarda a liberação de outro recurso alocado ou dependente do primeiro processo.

A definição textual de *deadlock* por ser muito abstrata, é mais difícil de se compreender do que a representação por grafos, que será resumida mais adiante. No entanto, algumas observações são pertinentes:

- O *deadlock* pode ocorrer mesmo que haja somente um processo no SO, considerando que este processo utilize múltiplos *threads* e que tais *threads* requisitem os recursos alocados a outros *threads* no mesmo processo;

- O *deadlock* independe da quantidade de recursos disponíveis no sistema;

- Normalmente o *deadlock* ocorre com recursos, tais como dispositivos, arquivos, memória etc. Apesar de a CPU também ser um recurso para o SO, em geral é um recurso facilmente preemptível, pois existem os escalonadores para compartilhar o processador entre os diversos processos, quando trata-se de um ambiente multitarefa.

Starvation

Starvation é quando um processo não consegue ser executado, de forma alguma, pois sempre existem processos de prioridade maior para serem executados, de forma que o processo "faminto" nunca consiga tempo de processamento.

O Starvation ocorre quando processos de maior prioridade aparecem sempre que tem um processo de menor prioridade ativo e querendo chamar o kernel. Nesse caso, os processos mais prioritários ficam com acesso ao kernel e o outro processo fica apenas esperando permissão, que nunca chega.

Ele não consegue agir e fica parado, sem fazer nada, ocupando memória e tempo de processamento (visto que o SO tem que fazer o escalonamento dos processos!) do processador sem fazer nada.