

AULA 7(03-12): THREADS, QUESTIONÁRIO E DISCUSSÃO

A aula teve um foco maior em discussão e roda de conversa. O início foi direcionado para uma revisão de conteúdo de threads. Após isso, respondemos um questionário a respeito do assunto. Por fim, foi sugerido pelo professor que pesquisássemos vagas de administrador de sistemas operacionais, motivado pelo curso da RedHat.

Abaixo, seguem algumas das definições apresentadas no início da aula:

Tipos de threads: há dois tipos de threads: de kernel e de usuário. Threads de kernel são totalmente gerenciadas pelo sistema operacional, mais especificamente pelo escalonador do kernel. Cada thread é conhecida pelo SO, que controla sua criação, destruição, mudança de estado e troca de contexto. Isso permite que threads de kernel usem múltiplos núcleos de forma nativa e que uma thread bloqueada não interrompa as demais. Por outro lado, criar e alternar entre threads de kernel é mais custoso, pois exige chamadas ao kernel, o que envolve mais overhead. Geralmente são usadas em sistemas que exigem alta responsividade e trabalho paralelo real. Enquanto isso, Threads de usuário são gerenciadas por uma biblioteca na aplicação, não pelo kernel. Para o sistema operacional, todo o processo parece ser apenas uma única thread. A troca de contexto entre threads dentro da aplicação é muito rápida, porque não envolve chamadas ao kernel. No entanto, isso traz limitações: se uma thread de usuário bloqueia, todo o processo bloqueia; além disso, threads de usuário não conseguem explorar múltiplos núcleos simultaneamente sem apoio do kernel.

Comunicação entre threads: threads podem ser entendidas como execuções concorrentes compartilhando um mesmo espaço de endereçamento. Isso garante velocidade e eficácia no processamento de atividades, haja vista que as threads podem acessar e fazer escrita no espaço de dados.

Benefícios do uso de threads: o uso de thread permite responsividade. Isto é, que o programa pare de funcionar por algum motivo, a thread que não teve seu funcionamento afetado segue com o processamento. Além disso, o compartilhamento de recursos pode ser visto como algo benéfico a depender do contexto. A escalabilidade em threads também é convidativa, além de uma maior economia de recursos e poder de processamento.

Desvantagens do uso de threads: problemas de condição de corrida, isto é, duas threads compartilhando recursos e o mesmo espaço em memória podem subscrever dados uma da outra. Além disso, o número de threads que podem ser criadas é limitado e seu excesso pode tornar o processamento lento.

Exemplos de uso de threads: há inúmeras aplicações para as threads, dentre as quais destacam-se: navegador web(para exibir imagens e recuperar dados da rede), uso em processamento de textos e uso em servidores web.

CONCLUSÃO

A partir dos pontos discutidos em aula, fica evidente que o uso de threads desempenha um papel fundamental na construção de sistemas modernos, oferecendo ganhos significativos em responsividade, desempenho e aproveitamento eficiente dos recursos do hardware. Entretanto, também exige atenção cuidadosa, especialmente quanto aos desafios associados à concorrência e ao compartilhamento de memória, que podem introduzir erros difíceis de detectar. A diferenciação entre threads de kernel e de usuário reforça a importância de compreender como cada modelo opera e suas respectivas implicações no comportamento do sistema. Em síntese, dominar o conceito de threads não apenas amplia a compreensão sobre o funcionamento interno dos sistemas operacionais, mas também prepara o estudante para lidar com situações práticas presentes no mercado de tecnologia e em áreas como administração de sistemas, desenvolvimento e otimização de software.

Aluno: João Victor Oliveira

Matrícula: 20240008468