**Lista de Exercícios -Sistemas Operacionais -Entrega 13/06**

**Threads**

1. Forneça dois exemplos de programação em que a criação de múltiplos threads proporciona melhor desempenho do que uma solução com um único thread.

Um navegadorWebpode estar fazendodownloadde um vídeo e ao mesmo tempo quepode exibir as partes do vídeo já baixadas.Ou também um processador de texto permite que o usuário digite os caracteres e aomesmo tempo faz as correções ortográficas

1. Cite duas diferenças entre threads de nível de usuário e threads de nível de kernel. Sob que circunstâncias um tipo é melhor do que o outro?

As threads do usuário são escalonadas pela biblioteca de threads e o kernel escalona asthreads do kernel

1. Descreva as ações executadas por um kernel para mudar o contexto entre threads de nível de kernel.

A troca de contexto entre as threads do kernel normalmente exige salvar o valor dosregistradores da CPU da thread que está saindo e restaurar os registradores da CPU danova thread que está sendo escalonada

1. Que recursos são usados quando um thread é criado? Em que eles diferem daqueles usados quando um processo é criado?

Como uma thread é menor do que um processo, a criação de thread normalmenteutiliza menos recursos do que a criação de um processo. A criação de um processo exigea alocação de um bloco de controle de processo (PCB), uma estrutura de dados umtanto grande. O PCB inclui um mapa de memória, uma lista de arquivos abertos evariáveis de ambiente. A alocação e o gerenciamento do mapa de memórianormalmente é a atividade mais demorada. A criação de uma thread do usuário ou dokernel envolve a alocação de uma estrutura de dados pequena para manter um conjuntode registradores, pilha e prioridade

1. Suponha que um sistema operacional mapeie threads de nível de usuário para o kernel, usando o modelo muitos-para-muitos e que o mapeamento seja feito por LWPs. Além disso, o sistema permite que os desenvolvedores criem threads de tempo real para uso em sistemas de tempo real. É necessário vincular um thread de tempo real a um LWP? Explique.

Sim. A temporização é fundamental para aplicações de tempo real. Se uma thread formarcada como tempo real, mas não estiver ligada a um LWP, a thread pode ter deesperar para ser conectada a um LWP antes de sua execução. Considere se uma threadde tempo real está executando (está conectado a um LWP) e depois bloqueie (ou seja,precisa realizar E/S, foi permutado por uma thread de tempo real de maior prioridade,está esperando por um lock de exclusão mútua etc.). Enquanto a thread de tempo realestá bloqueada, o LWP ao qual foi conectado foi atribuído a outra thread. Quando athread de tempo real tiver sido escalonada para executar novamente, primeiro terá deesperar para ser conectada a um LWP. Vinculado um LWP a uma tread de tempo real, você estará garantindo que a tread poderá ser executada com

1. Qual dos seguintes componentes de estado de um programa são compartilhados pelos threads em um processo com múltiplos threads?

a. Valores do registrador

b. Memória do heap

c. Variáveis globais

d. Memória da pilha

1. Uma solução com múltiplos threads usando múltiplos threads de nível de usuário pode obter melhor desempenho em um sistema multiprocessador do que em um sistema uniprocessador? Explique.
2. É possível haver concorrência, mas não paralelismo? Explique.
3. Considere o segmento de código a seguir:

pid\_t pid;

pid = fork();

if (pid == 0) {/\*processo-filho \*/

fork();

thread\_create( … );

}

fork();

1. Quantos processos únicos são criados?

b. Quantos threads únicos são criados?

10. Especifique como se dá a implementação de Threads no Windows, Posix e Java.