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

**Threads**

**O livro base para responder aos exercícios é:**

**SILBERSCHATZ, Abraham; Peter Baer Galvin, Greg Gagne. Fundamentos de Sistemas Operacionais. 9 ed. Rio de Janeiro: LTC, 2015.**

1. Forneça dois exemplos de programação em que a criação de múltiplos threads proporciona melhor desempenho do que uma solução com um único thread.
2. 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?
3. Descreva as ações executadas por um kernel para mudar o contexto entre threads de nível de kernel.
4. Que recursos são usados quando um thread é criado? Em que eles diferem daqueles usados quando um processo é criado?
5. 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.
6. 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?

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