1. Explique por que a programação mutithreading é considerada multiprogramação leve?

Principalmente porque a estrutura da thread é menos robusta do que a de um processo, sendo assim a troca de contexto entre uma thread e outra é mais leve do que a de um processo e outro.

2. Cite exemplos de programação multithreading com:

a. desempenho melhorado em relação a uma solução de thread único

b. desempenho não melhorado em relação a uma solução de thread único

1. Quando várias tarefas similares devem ser executadas várias vezes
2. Quando várias tarefas diferentes devem ser executadas várias vezes

3. Caracterize o descritor de um processo e o descritor de um thread. A partir desta caracterização, justifique o porquê threads são chamados de processos leves

4. Diferencie threads de usuário e threads de kernel.

Aparentemente elas são a mesma coisa, a diferença é que threads de usuário são criadas dentro do espaço de código do usuário e threads de kernel não está associada ao código do usuário e rodam dentro do contexto do kernel.

5. Quais os possíveis estados de um thread?

New, Runnable, Running, Non-Runnable e Terminated.

6. Quais recursos são usados quando um thread é criado? E quando um processo é criado?

7. Sobre os modelos multithreading, caracterize cada um deles, apresentando seus aspectos positivos e negativos.