

PERGUNTA 35 (THREAD)	PERGUNTA 35 (THREAD)
O que é um Thread?	Todo processo pode ser dividido em linhas de instruções ou grupo de instruções de código fonte. Essas linhas de instrução são chamadas de Thread, "Fio" em inglês. Podemos ter 2 tipos de Thread: Monothread e Multithread.
Qual é a grande vantagem de se usar threads no processamento? E quais cuidados devem ser tomados?	A vantagem das Threads é o ganho na alocação de recursos , afinal, threads podem compartilhar Contexto de Software e Espaço de Endereçamento . Além do ganho de tempo , quando um processo com thread é iniciado ele faz várias aplicações sem perder tempo iniciando e encerrando processos . Porém, é importante fazer uma boa sincronização e comunicação entre as threads para que elas não misturem os dados enquanto compartilham recursos .
O que é um Monothread?	Um processo Monothread é um único processo que divide o seu espaço de endereçamento com uma thread que se divide em processos independentes e sub-processos . Esses sub-processos e processos independentes são executados simultaneamente em um multiprocessador . Isso faz com que aquela única thread seja capaz de executar várias aplicações simultaneamente .
O que é um Multithread?	Um processo Multithread é um processo que aloca um thread de execução que compartilha o seu espaço de endereçamento com outros threads . Apesar de na Multithread não ser possível processar várias threads simultaneamente , só o fato de não ter que ficar iniciando e finalizando processos só para abrir novas instruções já é um grande ganho de tempo de processamento .
O que é um Thread de Usuário?	Na Thread Usuário a geração e implantação das threads vêm das aplicações e não do SO. Isso faz com que os programas sejam responsáveis por gerenciar e sincronizar threads ou multithreads, tirando essa tarefa do SO . Vantagens: possibilita a implementação de multithreads , mesmo que o SO não dê suporte para isso. Além da velocidade na execução das threads , que não necessitarão de acessar o kernel para executar instruções.
O que é um Thread de Kernel?	Diferente das Threads de Modo Usuário, as Threads de Modo Kernel são implementadas pelo kernel . O problema desses pacotes é quanto a baixa velocidade de execução , pois todas as threads em modo kernel precisam que o sistema operacional primeiro aceite o pedido pelo modo Kernel por através do System Call e depois mude o acesso da aplicação de Usuário para Modo Kernel , o que leva tempo muito mais tempo.
O que é um Thread Híbrido?	Nos Threads em Modo Híbrido é feita uma combinação entre as vantagens de threads usuário e kernel . Um processo pode ter threads Usuário e Kernel no mesmo processo . Além disso, o SO pode escalonar todos os threads no modo kernel, evitando fique entrando e saindo do modo kernel . Desvantagem: se um thread kernel solicitar System Call , todos os threads usuário ficam em estado de espera até o thread kernel desbloquear.