PERGUNTA 38 (SINCRONIZAÇÃO DE PROCESSOS)	PERGUNTA 38 (SINCRONIZAÇÃO DE PROCESSOS)
Qual a grande importância da sincronia entre os processos?	O kernel de um SO pode **executar simultaneamente as solicitações de uma multithread,** mas isso exige bons **mecanismos de sincronização para proteger as estruturas de dados internas e das threads.** Como por exemplo num serviço onde uma **aplicação multithread** faz uma solicitação **remota** enquanto as outras threads fazem outras **solicitações internas,** exige-se sincronismo entre processos para que a **aplicação ocorra corretamente.**
O que é o Padrão Pthreads?	Em **1995**, foi criado um padrão para **definir o uso de threads em SO´s,** seu nome é **POSIX P1003.1c, (Portable Operating System Interface ou Interface Portátil de Sistemas Operacionais) mas ficou mais conhecido como Pthreads.** Até hoje, ele é amplamente usado em **Unix e aplicações em C.** O Pthreads não apenas padroniza o uso de threads, como também o **uso de mecanismos de sincronização, como: semáforos, mutexes e variáveis condicionais.**
O que é o mecanismo de sincronização Mutexes?	**Mutex (Mutual Exclusion), ele implementa um "race conditions", (ou uma exclusão mútua) onde processos ficam esperando usar um recurso exclusivo que está sendo usado. Esse mecanismo evita que 2 processos tentem entrar no recurso ao mesmo tempo gerando deadlock. Funciona bem em threads, mas em multithreads que processam simultaneo, ainda gera deadlock, pois os processos são concorrentes e podem entram no recurso ao mesmo tempo.**
O que é o mecanismo de sincronização de Variáveis Condicionais?	**As Variáveis Condicionais** são mecanismos que **sincronizam processos usando condições variáveis para evitar conflitos, deadlocks e etc.** Por exemplo: um **contador de processos** que **toda vez que atinge determinado valor** obedece uma determinada ação. Esse mecanismo se faz **maior importância em aplicações multithread,** onde a sincronia entre as threads **precisa obedecer determinadas condições.**
O que é o mecanismo de sincronização Semáforo?	Os **Semáforos** são mecanismos **semelhantes aos Mutexes, mas melhorados.** Enquanto os mutexes só disponibilixam o uso de um recurso para um processo por vez, **Semáforos podem permitir que mais de um processo utilize um recurso exclusivo desde que obedeça a determinadas condições que garantam que um deadlock não será gerado.** Esse mecanismo é padrão **exclusivo do POSIX ou extensões para tempo-real como o POSIX.1b.**
O que são ordens de sincronização e qual a importância de estabelecê-las?	Como é **praticamente impossível determinar a ordem de ocorrência de um evento durante um processamento,** em especial quando essa ordem depende de eventos externos, como no caso das **redes distribuídas,** que solicitar ou atrasar processos a qualquer momento e em várias situações. Por isso, se viu **necessária a definição de ordem para que as sincronias entre os processos acontecessem.**
O que é uma Ordem Causal em sincronia de processos?	A **ordem Causal** define que todos os **processos reconheçam que se um evento depende que uma determinada ação aconteça para que se dê continuidade ao processamento.** Ou seja, ele **depende de "causa",** sem a causa ele se mantém em **estado de espera.** Acontece assim: Se os eventos **A** e **B pertencerem ao mesmo processo,** e A **precisa de receber B** antes de continuar, **A fica em espera até B chegar.**
O que é uma Ordem Total em sincronia de processos?	**A Ordenação Total** possibilita a ordenação de eventos segundo **relógios lógicos sem perder a Ordenação Causal**. Os relógios lógicos são **implementados pelo SO que usa uma lógica computacional para que determinar** que processos ocorram numa determinada **"Timestamp" ou "Marca de tempo"**. Esse mecanismo também é **usado em sistemas distribuídos,** onde **relógios lógicos escalares sincronizam relógios lógicos nos hospedeiros remotos**.