

PERGUNTA 34 (PROCESSO DOS SISTEMAS OPERACIONAIS)	PERGUNTA 34 (PROCESSO DOS SISTEMAS OPERACIONAIS)
Por que os Sistemas Operacionais de hoje executam "processos" e não "tarefas"?	Computadores que antes tinham apenas “uma rotina” para ser executada por vez, agora tinham várias tarefas a serem executadas simultaneamente . Sendo assim, para que os SO’s garantissem que cada tarefas fossem feitas com sucesso simultaneamente , os SO’s tiveram que começar a dividir as "Tarefas" em “Processos”, alocando registradores específicos para cada tarefa .
De que informações o Sistema Operacional precisa para executar um processo?	O SO precisa conhecer toda a estrutura necessária para que um programa solicitado seja executado. Ele precisa saber: como a alocação no processador está sendo usada? como a memória está sendo usada? e como os dispositivos de E/S estão sendo usados? Com essas informações ele consegue dividir esses recursos habilmente para cada programa em execução ("usuário") executando todos simultaneamente .
Qual a sensação que um "usuário" do processador precisa ter para que não haja conflito das aplicações?	O usuário (como chamamos um programa em execução) precisa ter a impressão de que todos os recursos do CPU estão exclusivamente disponíveis para ele , apesar de estarem sendo utilizados simultaneamente por outros usuarios. O CPU executa um usuário num intervalo de tempo e no instante seguinte estará executando outro usuário. (Time-Sharing) Se 2 usuários se encontrarem na mesma alocação, isso poderia ser um problema.
Como um processo está estruturado?	Um processo é formado por três partes : Contexto de Software; Contexto de Hardware; Espaço de endereçamento;
Como se dá o processo de um Contexto de Software durante um processamento?	Antes mesmo do SO iniciar um processo , ele especifica os limites e características para que um processo aconteça de acordo com dos recursos disponíveis na CPU . Ele faz isso definindo 3 características aos processos: Identificação, Quotas e Privilégios . Obs: às vezes essas características podem ser definidas, ou mudadas, durante a execução de um processo.
Como o Contexto de Software usa a Identificação de um processo para executá-lo?	Quando um processo é criado, ele recebe 2 Identificações: - O PID (Process Identification) , uma numeração que o SO usa para identificar e gerenciar o processo do início ao fim; - E o UID (User Identification) , uma numeração de segurança que os usuários usam para identificar somente as rotinas que pertencem ao programa que as originou , evitando conflito de rotinas.
Como o Contexto de Software usa as Quotas de um processo para executá-lo?	As Quotas, refere-se aos limites de recurso estipulados pelo SO para cada tarefa . Esses limites são estipulados com base na capacidade de alocação, memória, operações e processos disponíveis no momento . Quando o SO percebe que não tem o suficiente para que um processo seja executado de uma vez, ele fraciona o processo , fazendo ele ser executado lentamente, ser interrompido ou não ser executado .
Como o Contexto de Software usa os Privilégios de um processo para executá-lo?	Os Privilégios de um processo são identificados por através de um código que habilita a CPU a tomar ações especiais em relação a execução de um processo . Assim como tomar ações em relação a execução de outros processos , como interrompê-los para priorizar a si mesmo. Ou até mesmo privilégios concedidos pelo SO , como mudar estado de usuário para Modo Kernel e ter acesso as rotinas privilegiadas.
Como se dá o processo de um Contexto de hardware durante um processamento?	Como o nome já sugere, esse contexto gerencia o uso de hardware da CPU entre os multiprocessos em execução. Entre esses hardwares temos: Registradores, PC (Program Counter), Stack Pointer e Registrador de Status . Para utilizá-las simultaneamente o SO trabalha com interrupções , em segundos, um processo é interrompido e salvo num registrador, enquanto outro executado e assim sucessivamente .
Como se dá o processo de um Espaço de Endereçamento durante um processamento?	Cada processo tem instruções e dados que precisam ser armazenados juntos e alocados na mesma ordem e tamanho entre si . Quando essas instruções e dados são deslocados de um lugar para outro, eles precisam manter o mesmo espaço e localização . Isso é definido pelo Espaço de Endereçamento . Caso as instruções e dados se misturem , isso poderia gerar um conflito durante a execução do programa .

PERGUNTA 34 (PROCESSO DOS SISTEMAS OPERACIONAIS)	PERGUNTA 34 (PROCESSO DOS SISTEMAS OPERACIONAIS)
O que são PCB's e como eles são utilizados durante a execução de um processo?	Todos os processos implementados pelo SO são organizados em Blocos ou PCB (Process Control Block - Bloco de Controle de Processos) . No PCB os SO's mantém as informações, dados, contextos de hardware e software bem organizados . Todos PCBs em execução ficam ativos na memória RAM na área reservada para o SO. Essa área é limitada por parâmetros do SO , que limitam o máximo de processos simultâneos.
Por que um processo precisa ser dividido em estados de processo?	Como as CPU's executam vários processos simultaneamente e cada processo só pode ocupar o processador um de cada vez , raramente um processo será executado de uma vez. Havendo então a necessidade de dividir os processos em estados de processamento , até que eles sejam encerrados.
Quais os estados que um processo pode passar e quais as suas características?	Novo (New) : ao ser criado; Pronto (Ready) : o processo foi definido e escalonado, agora aguarda para ser executado; Em execução (Running) : está fazendo uso do processador; Em espera (Waiting) : aguardando algum evento externo ou recurso computacional. Por exemplo, o clique em um botão; Encerrado (Terminated) : processo é destruído pelo SO.
Por que duas razões um processo muda de estado?	Ele vai mudar por: - Eventos Voluntários : o próprio processo solicita a entrada de informações para realizar a execução ou a gravação de informações na memória e etc; - Eventos Involuntários : Como uma chamada para execução escalonada pelo SO, uma interrupção e etc;
Quais são as mudanças de estado de processo que podem ocorrer em um processo?	Pronto → Execução : Mudança Involuntária causada pelo SO; Execução → Espera : Mudança Voluntária onde o processo aguarda uma interação externa ou computacional; Espera → Pronto : Mudança Involuntária quando uma solicitação é atendida por um meio externo ou computacional; Execução → Pronto : Mudança Involuntária quando o tempo de execução expira e ele volta para a fila escalonamento;
Resumidamente, como ocorre um processo desde a sua criação até a sua destruição?	A criação ocorre a partir do momento em que o SO adiciona um novo PCB á um espaço de endereçamento na memória RAM . Assim que o PCB é criado, o SO já detecta sua existência gerencia o programa ao seu contexto , para que ele seja executado. Após a sua execução, o SO elimina o processo deletando o PCB da memória principal .