Lista preparatória prova AIA.

**1 – Defina Processo e Threads.**

**Processo**: É um programa que está sendo executado em um dos processadores virtuais do sistema operacional no momento em questão. **Thread**: Sub programa de um processo virtual ou de um programa virtual que tem o objetivo facilitar estruturas de gestão.

2 – **De exemplos de elemento que devem ser gerenciados por um processo pelo sistema operacional.**

Devem ser gerenciados o grupo de dados em execução garantindo a conformidade de dados, as condições de onde o processo se encontra para que possa voltar o processo em execução e a terceira, processos remotos originados.

**3 – Explique com suas palavras a importância da construção de threads em um processo servidor.**

Trabalhar os aspectos de cada elemento contido em um processo maior de forma única, estabelecendo uma necessidade baseada nas disponibilidades de um serviço. Utilizando menos memória pois lida apenas com um processo, e é mais eficiente enquanto ao gerenciamento. Isso diminui a carga do servidor enquanto a lidar com as transações. Além de que o usuário consegue atuar mais sob os sub processos.

**4 – Qual a importância da thread despachante e das threads operárias.**

O despachante é o elemento capaz de determinar a composição do grupo e as funcionalidades dos operários e os operários são os elementos que estabelecem as movimentações, a ideia funcional do ambiente. Despachante faz a conexão, o operário processa.

**5 – Discuta o papel de virtualização em sistemas distribuídos.**

Capacidade de estabelecer vários processos ao mesmo tempo, e esses processos podem ter características que o diferem tanto que não pode ser usados no mesmo sistema operacional, nesse caso é necessário uma camada de software (middleware) que tem a capacidade de adaptar as características de um software que trabalha com um sistema operacional para outro, a ideia de pegar uma forma de execução e transformar em outra.

**6 – Determine a razão para migrar códigos em um SD.**

A principal razão é estabelecer a possibilidade que o servidor não tenha que processar tudo ou seja tirar carga de processamento do servidor.

**7 – Discuta as mobilidade de um modelo de migração de código em um SD.**

Na mobilidade fraca é possível transferir somente segmentos de código, junto com dados de inicialização, um programa transferido é sempre inicializado de acordo com as várias posições de partida predefinidas, na mobilidade forte o segmento de execução também é transferido, um processo em execução pode ser parado, movido e retomar de onde parou.

**8 – Descreva as etapas de uma chamada a procedimento remota.**

Determinar uma interface, acionar a interface passando parâmetros através de uma mensagem, consequentemente espera uma resposta ou exceções no modelo síncrono ou continua o trabalho no modelo assíncrono.

**9 – Qual a importância da RPC assíncrona?**

Não precisa esperar O modelo assíncrono na primitiva de requisição você não espera pela resposta, a máquina gera um aceite da requisição, enquanto isso a máquina processa a requisição e devolve quando finalizado.

**10 – Explique com suas palavras o que é e para que serve uma interface de Berkeley.**

Interface de Berkeley é um terminal de comunicação, que permite que você inter-relacione as suas necessidades com elementos locais ou globais, serve para que as transações possam trocar primitivas de envio e recebimento de informações.

**11 – Qual a importância das comunicações orientadas a fluxo?**

É a importância e a capacidade de trafegar fluxos de dados em tempo real.

**12 – Explique a necessidade de módulos que implementam QoS em um sistema de comunicação orientada a fluxo.**

Garantir que através de uma portabilidade de pacotes comum você consiga estabelecer prerrogativas de controle de fluxo de dados, consequentemente consiga ter um fluxo orientado em um tempo adequado para que isso seja considerado um fluxo de dados.

Extras

**Virtualização de Recursos**

Capacidade de falsear um paralelismo através do rápido chaveamento entre thread e processo e outros recursos.

**Tabela de Processos**

Monitorar o processadores virtuais que são criados para executar cada programa do sistema operacional, essa tabela contém entradas para armazenar valores de CPU, mapas de memória, arquivos abertos, privilégios, etc.

**Contexto de Thread**

Consiste no contexto da CPU junto com informações para gerenciamento da thread.

**Processos Leves**

Executa no contexto de um único processo e pode haver vários LWP’s, veio para resolver problemas de implementação de thread, pois se implementar thread no nível de usuário a chamada bloqueadora bloqueia todo o processo que a thread pertence, se colocar a thread no nível do núcleo o custo é alto pois a criação, encerramento e sincronização terá de ser executada pelo núcleo, o que requer uma chamada de sistema

**Ativações de escalonador**

Outra abordagem para implementação de thread, a diferença dos LWP’s é que quando um thread bloqueia uma chamada de sistema, o núcleo faz uma upcall para o pacote, a vantagem é que ela poupa gerenciamento de LWP’s pelo núcleo.

**Multithread**

Facilita o desenvolvimento de servidores que exploram o paralelismo para obter alto desempenho, simplifica o código nos servidores

**Virtualização**

Há várias formas como uma virtualização pode ser implementada, em geral sistemas de computadores podem ter quatro tipos diferentes de interface:

* Uma interface entre o hardware e o software, o qual consiste em instruções de máquinas que podem ser invocadas por qualquer programa
* Uma interface entre o hardware e o software, o qual consiste em instruções de máquinas que podem ser invocadas somente por programas privilegiados como o sistema operacional
* Uma interface que consiste em chamadas de sistema como oferecidas por um sistema operacional
* Uma interface que consiste em chamadas de biblioteca que, em geral formam o que é conhecido como interface de aplicação de interface (API)

**Máquina virtual de processo**

O emulador imita o comportamento de chamadas de sistema, em essência essa virtualização é feita somente para um processo.

**Monitor de máquina virtual**

Camada que gerencia e executa diferentes sistemas operacionais independentemente e concorrentemente.

**Thin Clients**

Interface de rede de usuário em que tudo é processado e armazenado no servidor

**Sistema X Windows**

Sistema usado para controlar terminais mapeados em bits (controla o terminal)

**Nucleo X**

Contém todo o cerne do sistema(driver,etc – altamente dependente do hardware)

**Protocolo X**

Protocolo de comunicação de camada de aplicação pelo qual uma instancia pode trocar dados e eventos com o núcleo

**Documento Composto**

Conjunto de documentos, possivelmente de vários tipos bem diferentes, integrado no nível de interface de usuário

**Servidor iterativo e servidor concorrente**

Iterativo: é o próprio servidor que manipula a requisição.

Concorrente: não manipula por si próprio a requisição, passa para uma thread fazer ou para um processo

OBS: Clientes enviam requisições a terminais denominadas portas

**Fora da banda**

Dados que venham a ser processados pelo servidor antes de quaisquer outros dados daquele cliente

**Servidor sem estado**

Servidor que não mantem informações sobre o estado de seus clientes e pode mudar seu próprio estado sem informar a ninguém.

**Estado flexível**

Servidor promete manter estado em nome do cliente, mas por tempo limitado

**Servidor com estado**

Persiste informações dos seus clientes