

03 – Processos

- Processo é um programa em execução;
- Normalmente são independentes;
- Possuem espaços de endereçamento separados;
- Interage com outros processos por meio de IPCs (mecanismo de comunicação entre processos);

03 – Threads

- Fluxos de execução dos processos (subconjunto de processos);
- Compartilham o mesmo espaço de endereçamento e alguns dados da tabela de processo (isso se fizerem parte do mesmo processo);
- Thread significa (fio, linha), portanto linhas de execução dentro de um mesmo processo;

03 – Revisão de Processo

- Quando falamos em cliente e servidor estamos falando em hardware e software, portanto estamos falando de processo;
- Um processo servidor de arquivos com um único fluxo faz uma requisição do disco e aguarda o resultado;
- O mesmo servidor de arquivos com múltiplos fluxos pode atender a solicitação de outros usuários. Aumento do throughput e do desempenho;

03 – Revisão de Processo

Processo e Threads não distribuídos:

- Chaveamento de Processo é um processo caro;
- Tem que salvar e carregar as informações de contexto;

03 – Processo e Threads

Processo e Threads em sistemas distribuídos:

- Threads do kernel permitem chamadas bloqueadas sem bloquear todo o processo;

Vantagens das Threads em SD's:

- Permitem criar múltiplas conexões, cada uma sendo implementada por uma thread;
- Assim, oculta-se a latência da comunicação na rede (transparência);
- Pode-se iniciar com a comunicação e logo depois realizar uma outra tarefa

03 – Clientes Multithreads

Clientes Multithreads – Browsers Web

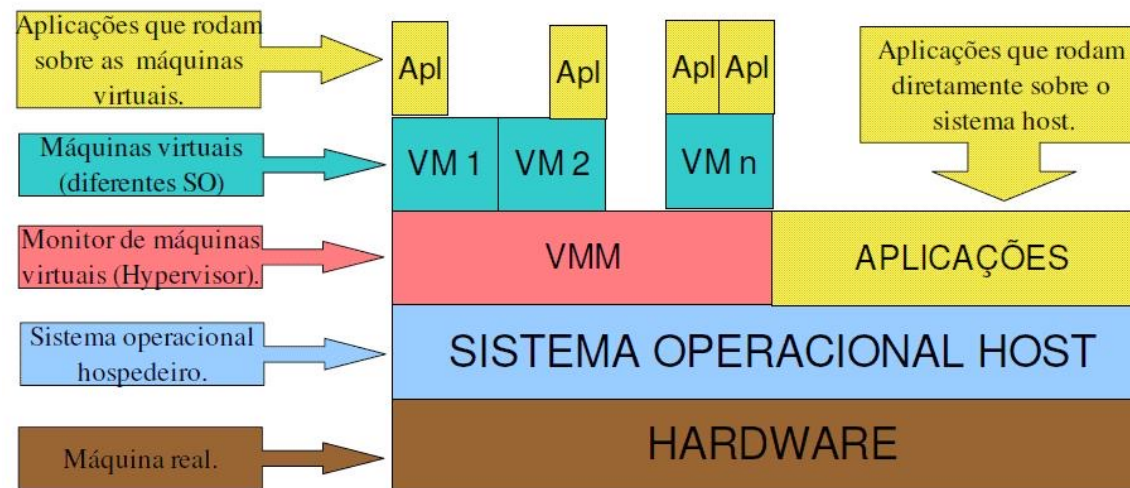
- Considerando uma conexão persistente com paralelismo;
- Requisições são feitas sem que todos os objetos tenham chegado na máquina do cliente;
- Cliente pode manipular diversos fluxos em paralelo usando as threads;
- Vantagem: usuário não necessita esperar até que todos os componentes da página cheguem (transparência);
- Thread despachante atribui tarefas para as threads operárias, caso todas estejam ocupada ela mesma realiza;

03 – Virtualização

- Threads e processos dão ilusão de que várias tarefas estão sendo feitas ao mesmo tempo;
- Em computadores com uma CPU, a execução simultânea é uma ilusão:
 - Única CPU – somente uma thread ou processo será executado por vez;
- Virtualização de Recursos: Fingir que um determinado recurso está replicado no sistema;

03 – Virtualização

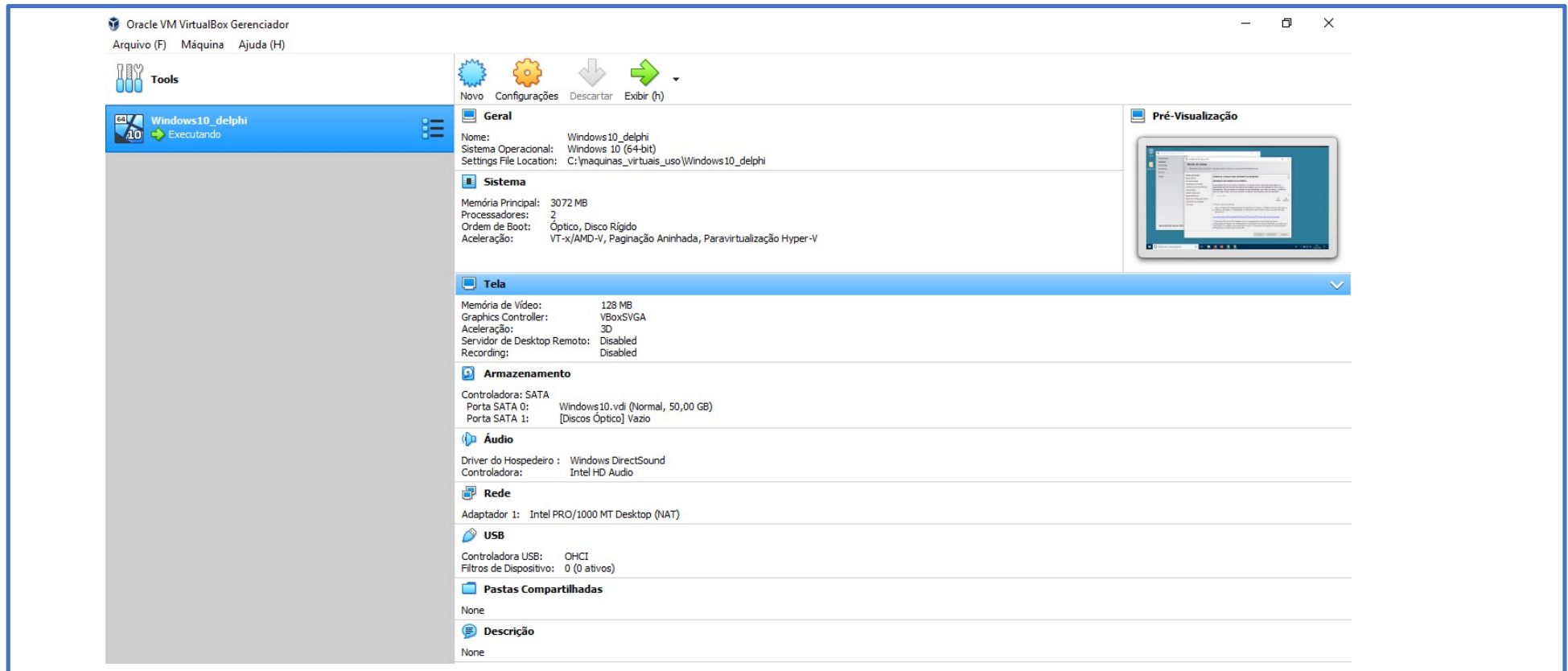
- Sistema de Virtualização:
 - Estende ou substitui uma interface existente para imitar o comportamento de um outro sistema;



03 – Virtualização

- VirtualBox é um software de virtualização desenvolvido pela empresa Innotek depois comprado pela Sun Microsystems que posteriormente foi comprada pela Oracle que, como o VMware Workstation, visa criar ambientes para instalação de sistemas distintos. Wikipédia

03 – Virtualização



03 – Virtualização

- VMware, Inc. é uma empresa subsidiária da EMC Corporation que produz um software de máquina virtual que permite a instalação e utilização de um sistema operacional dentro de outro, dando suporte real a software de outros sistemas operativos. Wikipédia

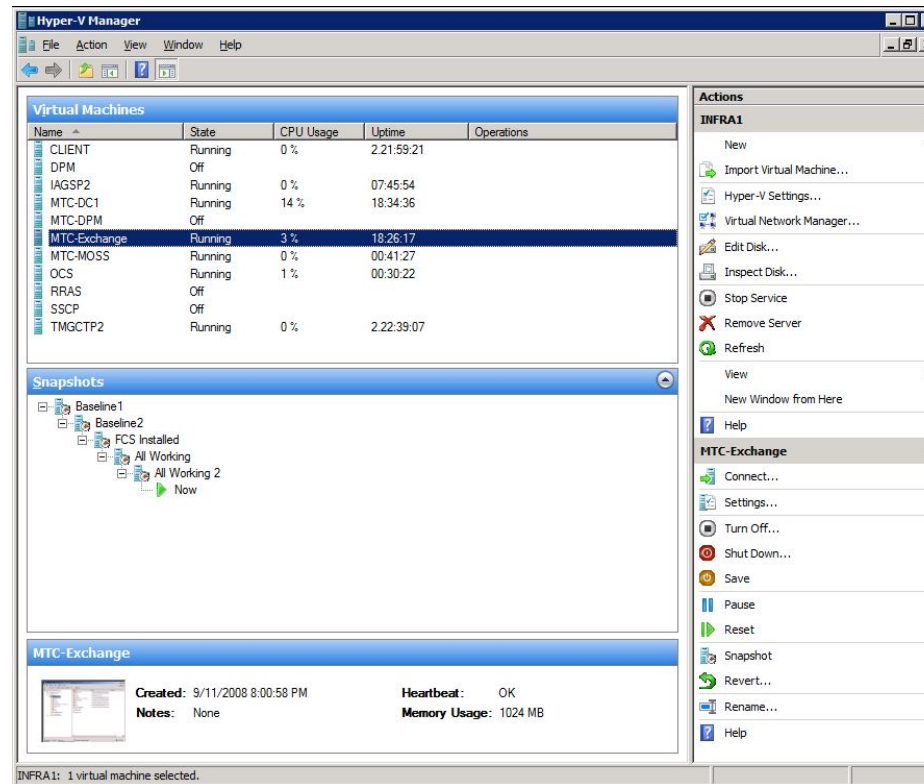
03 – Virtualização



03 – Virtualização

- Microsoft Hyper-V, codinome Viridian e anteriormente conhecido como Virtualização no Windows Server, é uma tecnologia de virtualização baseada em Hipervisor nativo. Wikipédia

03 – Virtualização



03 – Virtualização

Virtualização em meados de 70...

- A virtualização foi aplicada com sucesso nos mainframes da IBM;
- Estes acompanhavam uma máquina virtual para executar os softwares;
 - Executavam aplicações e Sistemas Operacionais dentro do IBM 370;
- Não é um conceito novo;
- Até hoje essas VM's são utilizadas;

03 – Virtualização

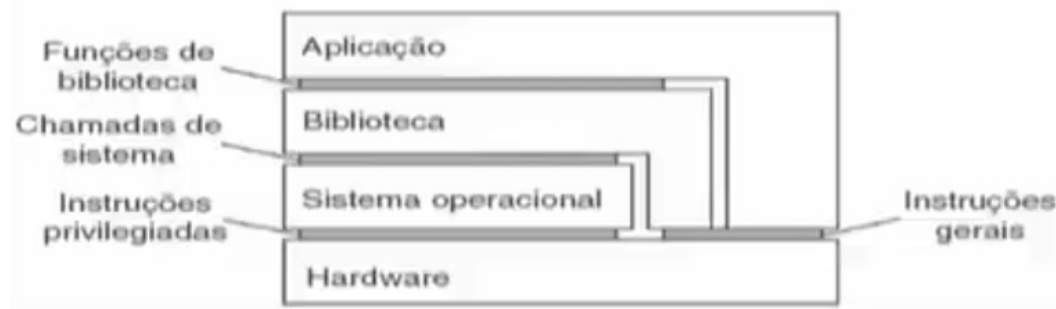
Virtualização Portabilidade

- Softwares em nível mais alto são mais estáveis (não mudam muito) do que o hardware e sistemas de baixo nível;
- Virtualização (middleware) pode ajudar transportando as interfaces de softwares para novas plataformas;
- Novas plataformas são capazes de executar softwares existentes anteriormente;

03 – Virtualização

Arquitetura de máquinas virtuais:

- Existem 4 tipos e níveis diferentes de interfaces;
- Níveis em que a virtualização pode ser implementada;



03 – Virtualização

Em resumo, virtualizar é imitar o comportamento das interfaces (instruções de máquina, chamadas de sistema – system calls);

Existem dois tipos de virtualização:

- Máquina Virtual de processo;
- Monitor de máquina virtual;

03 – Virtualização

Máquina Virtual de Processo:

- Aplicações desenvolvidas pra um Sistema Operacional são executadas em outro Sistema Operacional;
- Virtualização feita apenas para um único processo;

Exemplo:

Wine, diretório/application.exe

Serve para execução de aplicativos do Windows no Linux.

03 – Virtualização

Monitor de máquina virtual:

- Fornece o conjunto de instruções completo do hardware;
- Vários sistemas operacionais diferentes executando independente e concorrentemente na mesma plataforma;
- Segurança: Isolamento de uma aplicação e seu ambiente;
- Falhas não afetam a máquina inteira;

Exemplos:

VMWare e Virtualbox;

03 – Implementações dos Processos Distribuídos

Clientes: Modo de Interação com o Servidor.

Processos dos clientes – permitem que os meios interajam com servidores remotos a partir de dois modos:

- Fat Client: Para cada serviço remoto – Um servidor separado com que possa estabelecer contato;
 - A aplicação é mais robusta e possui seus próprios protocolos;
- Thin Client: Cliente provê apenas uma interface;
 - O middleware auxilia na interação;

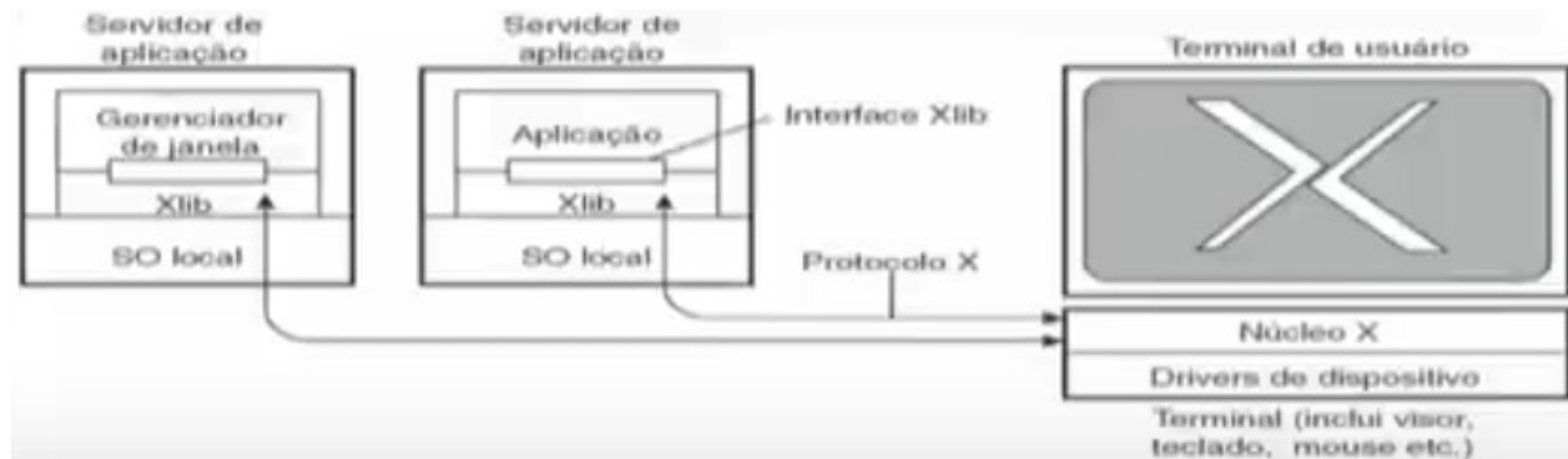
03 – Implementações dos Processos Distribuídos

Dois modos de Interação com o Servidor:

- Protocolo de Interação é implementado no nível da aplicação e é específico a uma aplicação;
- Protocolo de Interação é implementado no nível do middleware e é genérico;

03 – Implementações dos Processos Distribuídos

- Implementação de um cliente minimizado thin cliente;
- X Windows é uma implementação de interface de usuário em rede;



03 – Implementações dos Processos Distribuídos

Implementação no lado do cliente:

- Transparência de Migração: O middleware no cliente oculta do usuário a corrente localização do servidor, caso esse migre para outro sitio;
- Transparência a falha: O middleware pode tentar a conexão com o servidor repetida vezes ou tentar um outro servidor após n tentativas.
 - Uso de cache: falha na conexão com o servidor.
- Transparência de Replicação: várias réplicas de requisição e uma resposta é enviada à aplicação;

03 – Implementações dos Processos Distribuídos

Vamos falar de Servidores:

- Como tratar as requisições?
- Como os clientes contatam um servidor?
- Como interromper a comunicação com o servidor?
- Servidor deve guardar estado da comunicação dos clientes?

03 – Implementações dos Processos Distribuídos

Servidor Iterativo

- Próprio servidor manipula a requisição;
- Responde ao cliente quando necessário;

Servidor Concorrente

- Não manipula por si próprio a requisição;

Ex. Servidor Multithread.

- Processos ou threads respondem ao cliente;

03 – Implementações dos Processos Distribuídos

Servidores: Como os clientes contatam?

- Um daemon informa o # da porta;
- Um superservidor bifurca a chamada de uma porta para o servidor correto;

03 – Implementações dos Processos Distribuídos

Servidores: Interrompendo a Comunicação.

- Abordagem mais simples: o usuário sai abruptamente da aplicação cliente (servidor encerrará a conexão antiga).

Abordagem mais completa: dados “urgentes” (fora da banda)

- Pacotes TCP possuem o campo URG no header;
- Servidor ao receber um pacote com o campo URG setado é interrompido para trata-lo;

03 – Implementações dos Processos Distribuídos

Servidores: Manutenção de Estado

Três implementações

- Sem estado;
- Estado flexível;
- Com estado;

03 – Implementações dos Processos Distribuídos

Servidores sem estado:

- Não mantem informações sobre os estados dos cliente, por exemplo, o que acontece com o protocolo HTTP.
- Servidor Web: Após processar uma requisição, esquece o cliente (e.g. não guarda os acessos dele).
- Mesmo sem estado, servidor web guarda informações de clientes;
- Uma alternativa para armazenar estados é com os cookies;

03 – Implementações dos Processos Distribuídos

Servidores de estado flexível:

- Servidor promete manter estado em nome do cliente por tempo limitado;
- Após o tempo limitado, o servidor descarta estas informações;

Ex.: Servidor promete manter um cliente informado sobre as atualizações;

03 – Implementações dos Processos Distribuídos

Servidores com estado

- Mantém informações persistentes sobre seus clientes;
- Ex.: Servidor de Arquivos que permite ao cliente manter “cópia extra” local de um arquivo.
- Servidor mantém uma tabela com entradas (cliente, arquivo);
- Monitora as permissões sobre um arquivo, versão atual, etc;
- Desafio: tolerância a falhas para manter todas as informações;

03 – Implementações dos Processos Distribuídos

Migração de Código: (migra o processo em caso de lentidão / processo é o programa em execução).

- Principal razão (aumento de desempenho);
- Envio de processos para máquinas menos sobrecarregadas;
- Enviar grande quantidade de mensagens trocadas entre aplicações cliente-servidor:
 - Ex1: Operações de banco de dados que envolvem uma grande quantidade de troca de dados;
 - Ex2: Formulários enviados do servidor para o cliente;

03 – Implementações dos Processos Distribuídos

Tipos de Migração:

- Mobilidade Fraca: Transfere apenas o segmento de código e alguns dados de inicialização;
 - Requer somente que a máquina-alvo possa executar o código (portabilidade);
- Mobilidade Forte: Segmento de execução.
 - O processo em execução pode ser parado, movido para uma outra máquina e retomar a execução no ponto original;