

30-769

Sistemas Operacionais II

MSc. Fernando Schubert

Visão Geral

- Sistemas Operacionais
- Sistemas Operacionais em Rede
- Sistemas Operacionais Distribuídos
- Virtualização
- Tendências Atuais

Sistemas Operacionais

- O que é um sistema operacional?
- Um sistema operacional é um gerenciador de recursos
- Fornece uma interface de computação abstrata
- O SO arbitra o uso de recursos entre processos
 - CPU, memória, sistema de arquivos, rede, teclado, mouse, monitor
 - Outros dispositivos de hardware
- Isso torna possível ter vários processos no mesmo sistema
 - Se dois processos solicitarem o mesmo recurso
 - O SO decide quem o utilizará, quando e por quanto tempo

Sistemas Operacionais

Operating System Components

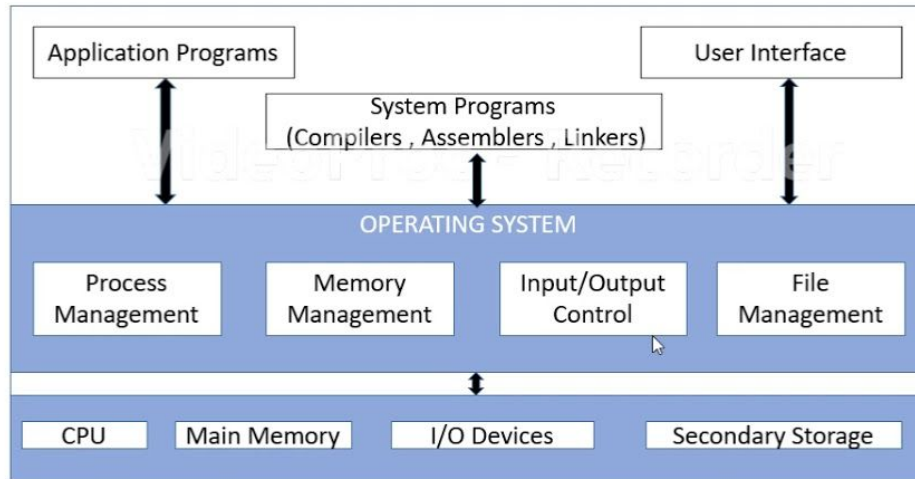
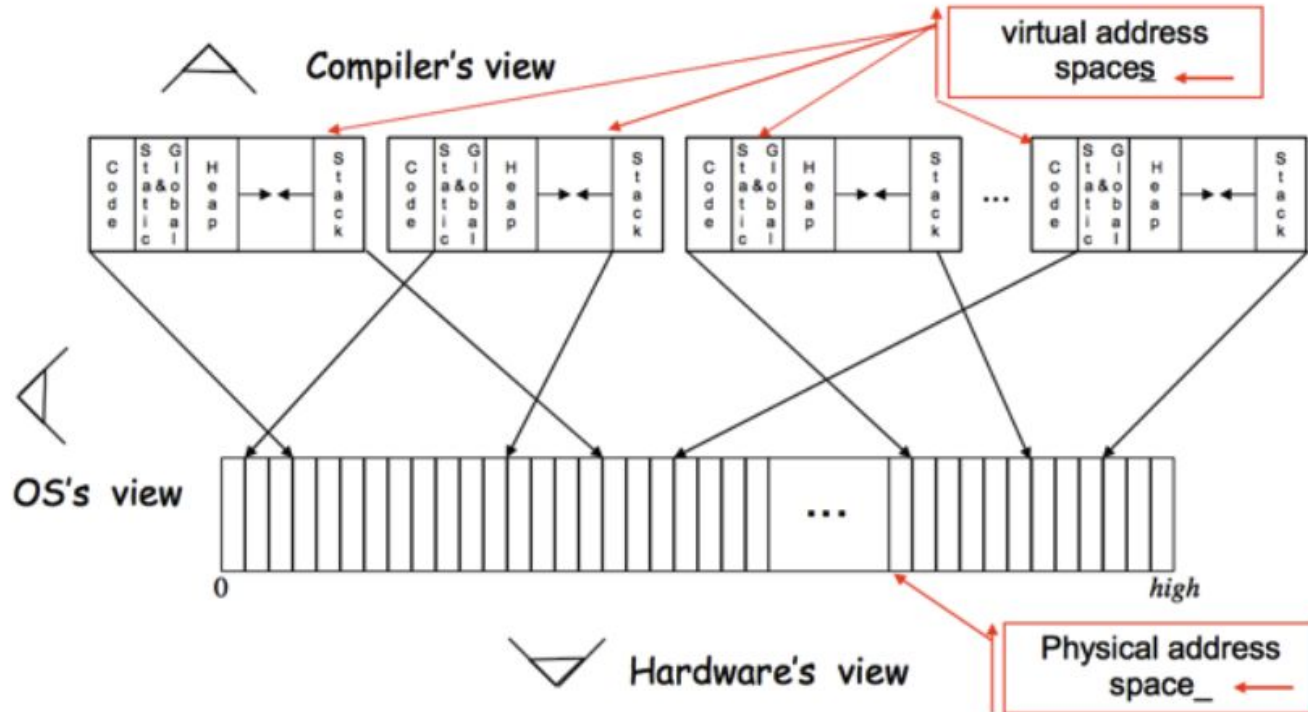


Figure : OS Components

Como o SO Gerencia Diferentes Recursos

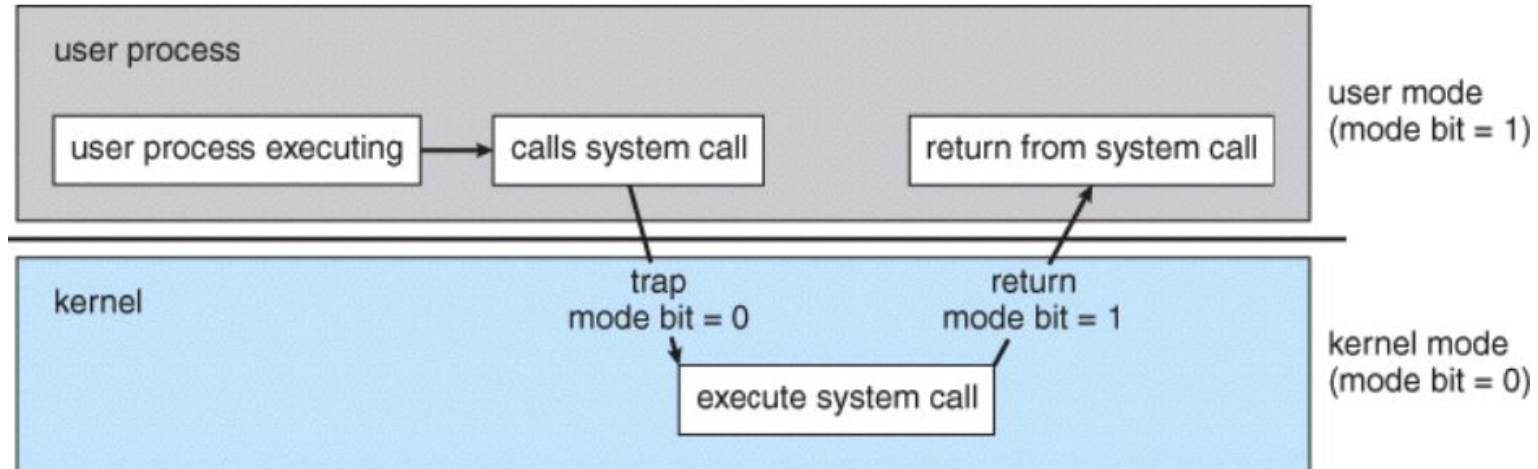
- Memória:
 - Cada processo recebe uma parte diferente da memória para usar e não pode acessar a memória de outro
 - Se precisar de mais memória, o SO a alocará do armazenamento de memória não utilizada
- Sistema de Arquivos:
 - O SO verifica se o processo tem direitos de leitura/gravação no arquivo
 - Garante que dois processos não escrevam no mesmo arquivo ao mesmo tempo
- Rede:
 - O SO recebe mensagens dos processos e as envia para a placa de rede uma de cada vez
 - Quando as mensagens são recebidas, o SO as entrega aos processos adequados

Memória Virtual



Operação em Modo Kernel/Usuário

- Modo Kernel: Tem controle total sobre o hardware, executa instruções privilegiadas e gerencia recursos críticos do sistema.
- Modo Usuário: Executa aplicativos comuns com permissões limitadas, sem acesso direto ao hardware, garantindo segurança.

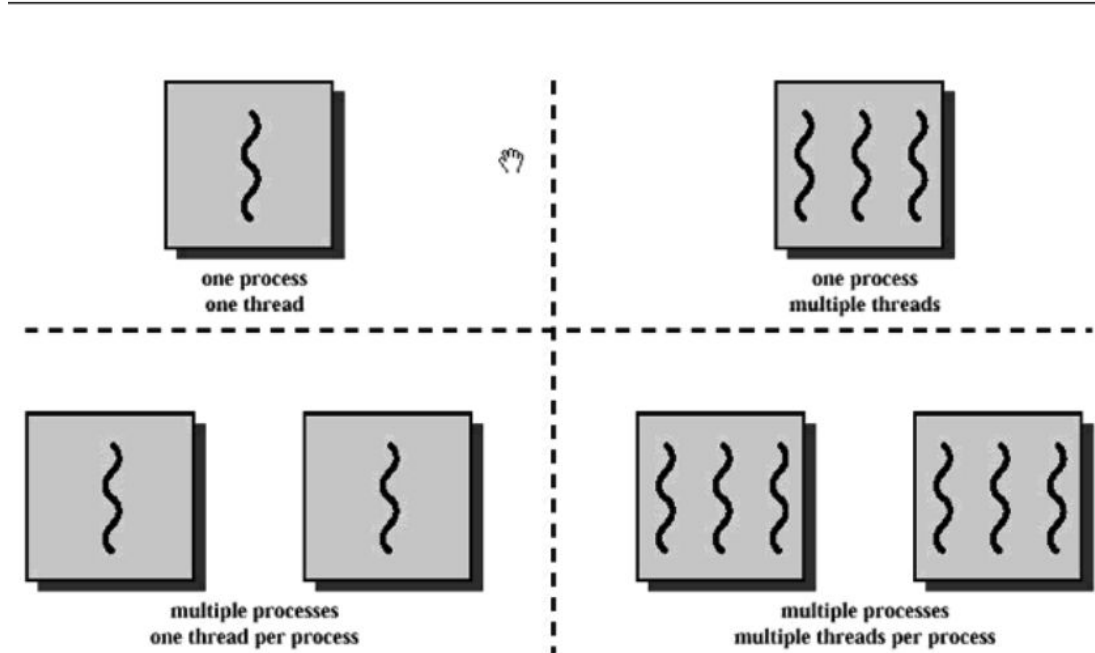


Abstração do SO

- O processo não conhece os detalhes do hardware
- O processo não sabe sobre outros processos (a menos que se comuniquem entre si)

Threads

Threads



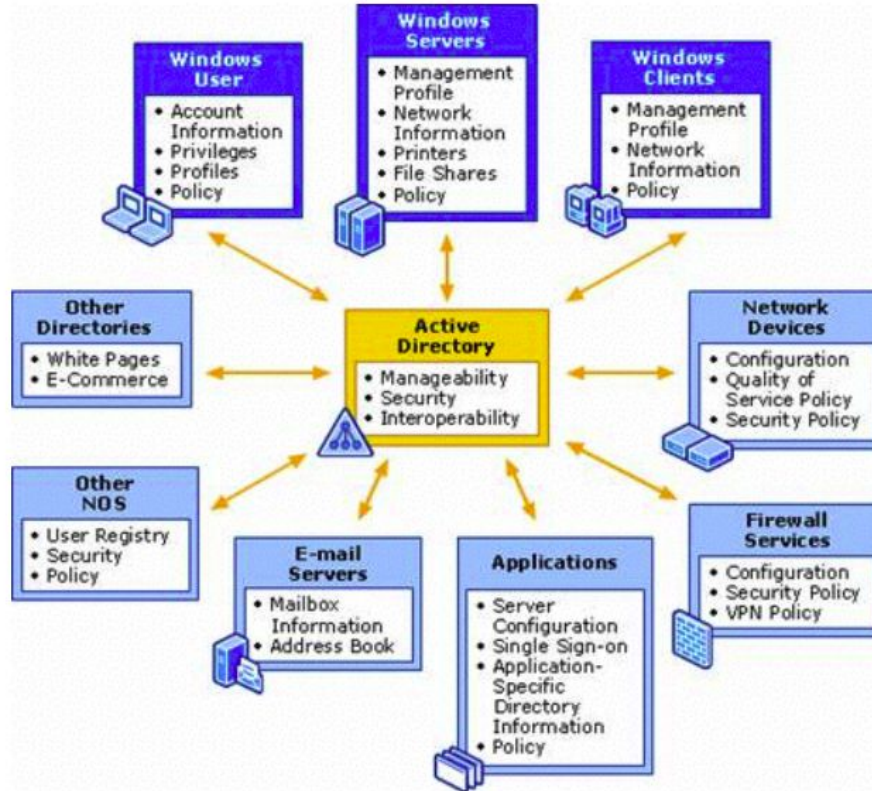
Benefícios de Threads

- Responsividade: mesmo que parte do programa esteja bloqueada ou realizando uma operação demorada, o multithreading permite que o programa continue.
- Compartilhamento de Recursos: threads compartilham a memória e os recursos do processo dentro do mesmo espaço de endereço.
- Economia: a alocação de memória e recursos para criação de processos é custosa. Threads compartilham os recursos do processo ao qual pertencem, sendo a criação e a troca de contexto de threads mais econômica.
- Utilização de Arquiteturas Multicore: em sistemas multicore, as threads rodam em paralelo em diferentes núcleos.

Sistema Operacional em Rede

- Um sistema operacional em rede tem consciência de que está conectado à rede.
- Cada nó tem um SO rodando.
- Cada nó gerencia os recursos naquele nó.
- Um processo pode solicitar comunicação com processos em outros nós, mas deve estar explicitamente ciente de que está solicitando serviço em um nó diferente e qual nó (por exemplo, endereço IP).

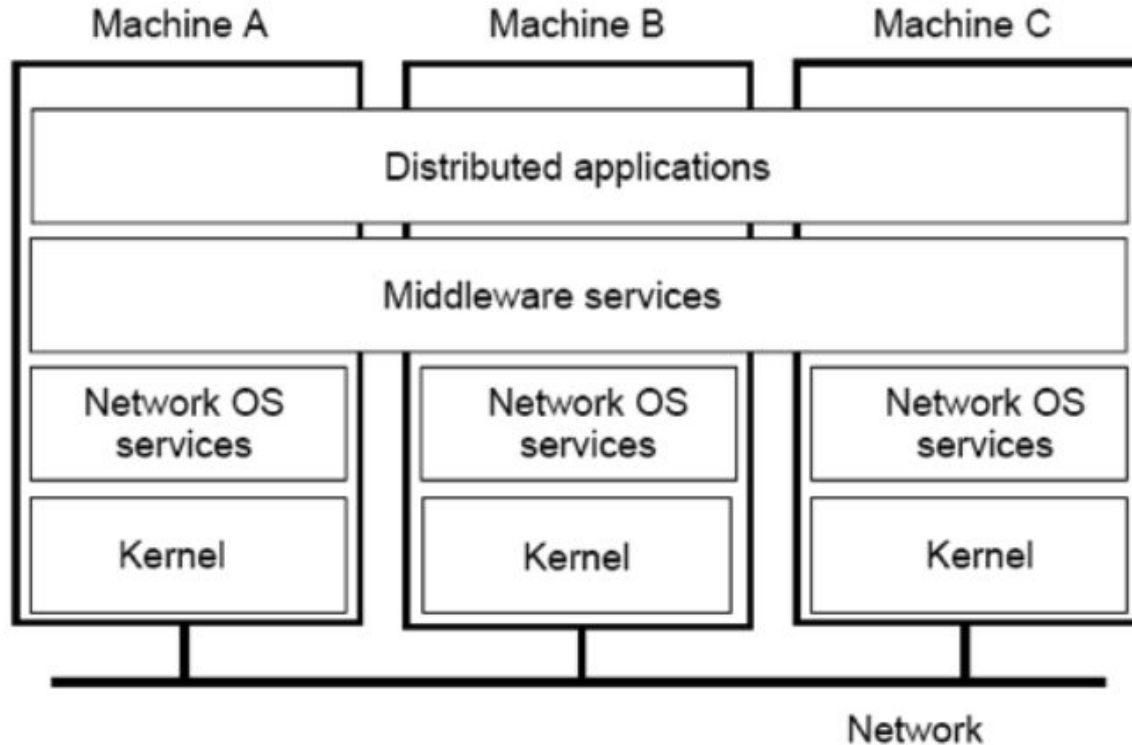
Sistema Operacional em Rede



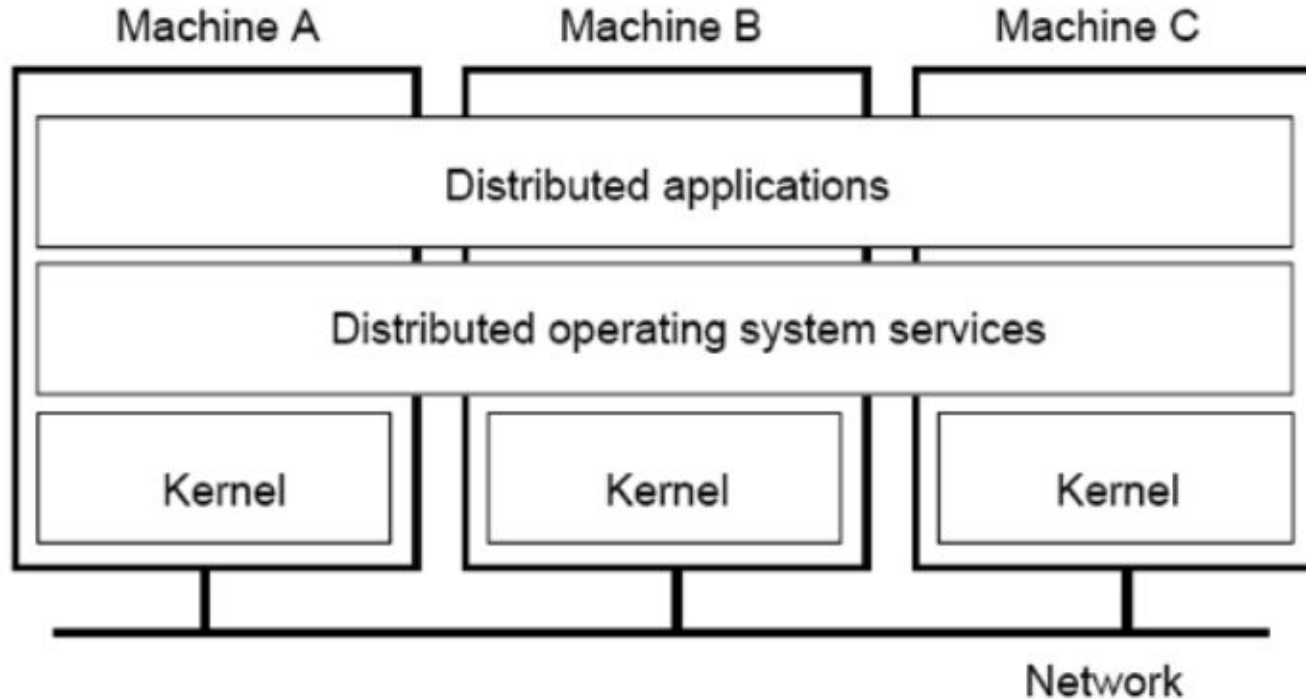
Sistema Operacional Distribuído

- Os sistemas operacionais rodando em diferentes computadores funcionam como um único SO.
- O processo não sabe (nem precisa saber) que outros recursos/processos estão em outros computadores.
 - O processo obtém entrada/saída do hardware X, que pode estar em qualquer computador.
 - O processo A se comunica com o processo B da mesma forma, independentemente de estarem no mesmo computador ou não.
 - O SO cuida de usar a rede, se necessário.
- Um processo pode estar rodando em um computador diferente daquele em que foi iniciado. Processos podem ser movidos entre diferentes computadores.
- A natureza “distribuída” do sistema é ocultada dos processos.
- O SO gerencia todos os aspectos "distribuídos".

Sistema Operacional Distribuído



Sistema Operacional Distribuído



Problemas com SO distribuído

- O que acontece se parte da rede falhar e os processos forem separados em dois conjuntos?
 - Agora temos que informar aos processos que a rede falhou, e o processo precisa tomar uma ação.
 - E se alguns processos do SO tiverem sido movidos para outro lugar?
- Suponha que iniciemos os processos A e B no mesmo computador.
 - O SO os move para computadores diferentes.
 - Mas A e B se comunicam muito, então teria sido mais eficiente mantê-los no mesmo computador!

Problemas com SO distribuído

- Acesso a recursos externos
 - Deve ser feito através de uma conexão de rede explícita.
 - Todos os computadores do mundo não podem estar no mesmo sistema!
- Adicionar novos nós à computação distribuída
 - Pode fazer parte de uma instância diferente do SO.
 - Ainda precisaremos de conexões explícitas.

Problemas com SO distribuído

- Uma falha de rede ou de computador significa que parte do sistema operacional falhou.
 - É difícil projetar um sistema operacional com tolerância a esse tipo de falha.
- O sistema operacional distribuído precisa lidar com várias possibilidades diferentes de computação distribuída.
 - Isso torna o design mais complexo.
- Na verdade, não é possível prever e lidar com todas as diferentes possibilidades.
- "Computação distribuída" significa coisas diferentes em diferentes contextos.
- É melhor deixar que o programador da aplicação decida como será feita a distribuição, como lidar com a comunicação, falhas, etc.
- O sistema operacional fornece apenas a infraestrutura básica para isso.

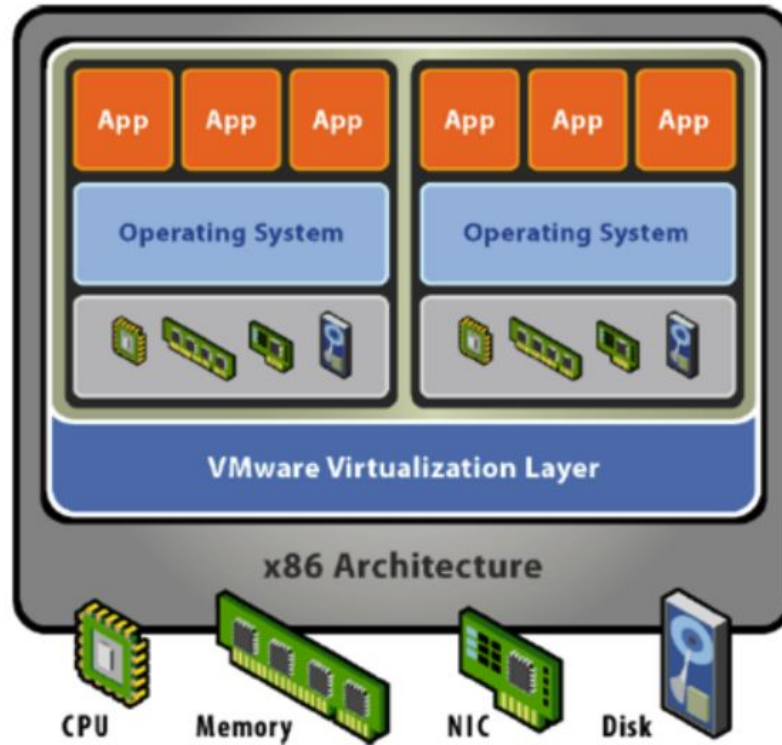
SO em rede vs SO distribuído

- Como resultado, não temos nenhum sistema operacional distribuído em uso regular.
- Sistemas operacionais em rede são populares.
- Eles fornecem facilidades de comunicação.
- Deixam que o software decida como executar a computação distribuída.
 - Mais flexibilidade.
 - Falhas, entre outros, são responsabilidade da aplicação.
 - O sistema operacional continua realizando apenas as tarefas básicas.

Computação distribuída e SO em rede

- Use algoritmos distribuídos na camada de aplicação para:
 - Sincronização
 - Ordenação consistente
 - Exclusão mútua
 - Eleição de líder
 - Detecção de falhas
 - Multicast
 - Etc.
- Além disso, projete aplicações de computação distribuída.
- Diferentes aplicações precisarão de conjuntos diferentes de funcionalidades.

Virtualização



Virtualização

- Várias instâncias de sistemas operacionais podem ser executadas simultaneamente dentro de máquinas virtuais em um único computador, particionando e compartilhando dinamicamente os recursos físicos disponíveis, como CPU, armazenamento, memória e dispositivos de E/S.
- Arquitetura hospedada ou hypervisor:
 - A arquitetura hospedada instala e executa a camada de virtualização como um aplicativo em cima de um sistema operacional.
 - A arquitetura hypervisor (bare-metal) instala a camada de virtualização diretamente em um sistema baseado em x86 limpo.
 - Acesso direto ao hardware:
 - Mais eficiente, com maior escalabilidade, robustez e desempenho.

Virtualização

- Considere uma fazenda de servidores:
 - Muitos servidores diferentes estão em operação.
 - Em vez de fornecer um servidor físico para cada um, muitas fazendas de servidores consistem em servidores reais executando máquinas virtuais.
 - Por exemplo, ao alugar um servidor para hospedar um site, é provável que você receba um servidor baseado em máquina virtual (VM).

Virtualização

- Vantagens: mais flexibilidade
 - Múltiplas VMs no mesmo computador.
 - Necessita de menos máquinas físicas.
 - Mais fácil de ligar/desligar.
 - Mais fácil de fazer backup.
 - As VMs podem ser movidas de um computador para outro, preservando seu estado.
 - Útil quando a carga de trabalho muda; alguns servidores precisam de mais computação, enquanto outros precisam de menos.

Virtualização e sistemas distribuídos

- Essa não é uma boa estratégia para computação intensiva em CPU, como mineração de dados em larga escala.
- Isso ocorre porque executar uma grande computação em uma máquina virtual é ineficiente.
- No entanto, muitos sistemas precisam de computação funcionando o tempo todo, mas não de forma tão intensiva.
- A virtualização é mais útil quando a flexibilidade é crítica.