

Computação em nuvem

Virtualização

Gabriel V C Candido

Instituto Federal do Paraná - Pinhais

Sumário

Contexto

História

Interfaces

Virtualização de interfaces

Tipos de máquinas virtuais

Sumário

Contexto

História

Interfaces

Virtualização de interfaces

Tipos de máquinas virtuais

Camadas da nuvem

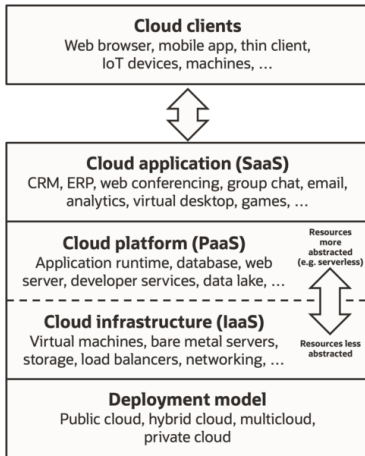


Figura: Modelos da computação em nuvem visualizados em camadas.

Fonte: Wikipedia

Sumário

Contexto

História

Interfaces

Virtualização de interfaces

Tipos de máquinas virtuais

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos
- ▶ década de 1960: IBM e o M44/44X

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos
- ▶ década de 1960: IBM e o M44/44X
- ▶ década de 1970: IBM OS/370

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos
- ▶ década de 1960: IBM e o M44/44X
- ▶ década de 1970: IBM OS/370

Cada usuário tinha um ambiente independente, desvinculado e monousuário

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos
- ▶ década de 1960: IBM e o M44/44X
- ▶ década de 1970: IBM OS/370
- ▶ década de 1980: perde importância com a chegada dos PCs

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos
- ▶ década de 1960: IBM e o M44/44X
- ▶ década de 1970: IBM OS/370
- ▶ década de 1980: perde importância com a chegada dos PCs
- ▶ década de 1990: Java; volta a ganhar tração

História

- ▶ ~ 1960: multiprogramação, compartilhamento de recursos
- ▶ década de 1960: IBM e o M44/44X
- ▶ década de 1970: IBM OS/370
- ▶ década de 1980: perde importância com a chegada dos PCs
- ▶ década de 1990: Java; volta a ganhar tração
- ▶ 2000 ~ -: virtualização se consolida junto com servidores e nuvem
- ▶ Hoje: suporte nativo em processadores; linguagens compiladas para máquinas virtuais

Sumário

Contexto

História

Interfaces

Virtualização de interfaces

Tipos de máquinas virtuais

Arquitetura de von Neumann

- ▶ Unidade lógica e aritmética
- ▶ Unidade de controle
- ▶ Memória
- ▶ Entrada e saída

Camadas de um sistema de computador

► aplicações

Camadas de um sistema de computador

- ▶ aplicações
- ▶ sistema operacional e software de sistema

Camadas de um sistema de computador

- ▶ aplicações
- ▶ sistema operacional e software de sistema
- ▶ hardware

Camadas de um sistema de computador

- ▶ aplicações
- ▶ sistema operacional e software de sistema
- ▶ hardware

Camadas de um sistema de computador

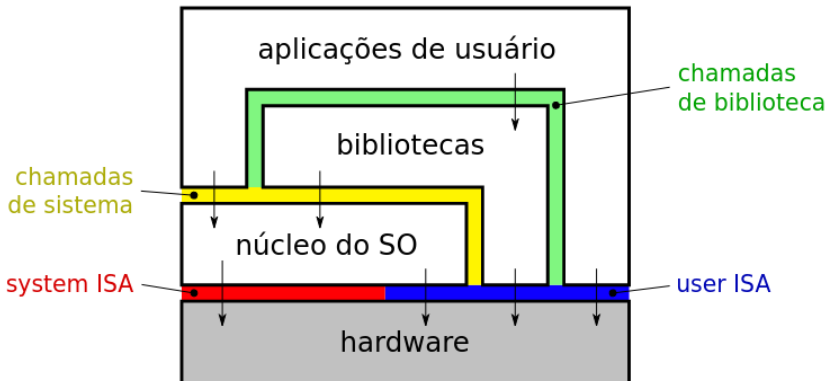


Figura: Componentes e interfaces de um sistema computacional. Fonte: CM 31

Camadas de um sistema de computador

- ▶ ISA: *instruction set architecture*
- ▶ chamadas de sistema: *syscalls*
- ▶ bibliotecas possuem API (*application programming interface*)
- ▶ exemplos de bibliotecas: libc (printf, fopen), GTK+, SDL
- ▶ precisamos compilar programas e bibliotecas respeitando as regras do SO e do processador (*User ISA e syscalls*)

Compatibilidade entre interfaces

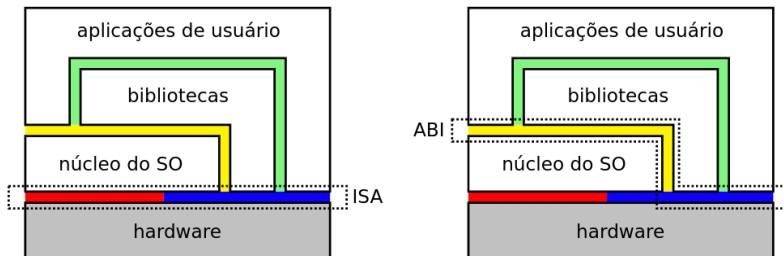


Figura: Interfaces de sistema ISA e ABI. Fonte: CM 31

Compatibilidade entre interfaces

- ▶ ABI: *application binary interface*
- ▶ ISA e ABI são pouco flexíveis: não criamos novas instruções ou *syscalls*
- ▶ Se isso fosse possível: podemos quebrar o que já está funcionando

Compatibilidade entre interfaces

- ▶ sistema operacional é projetado para aproveitar ao máximo os recursos

Compatibilidade entre interfaces

- ▶ sistema operacional é projetado para aproveitar ao máximo os recursos
- ▶ hardware, SO e aplicações são desenvolvidos de forma independente

Compatibilidade entre interfaces

- ▶ sistema operacional é projetado para aproveitar ao máximo os recursos
- ▶ hardware, SO e aplicações são desenvolvidos de forma independente

Plataformas diferentes e incompatíveis!

Compatibilidade entre interfaces

- ▶ interfaces possibilitam desenvolvimento independente

Compatibilidade entre interfaces

- ▶ interfaces possibilitam desenvolvimento independente
- ▶ porém gera dependências pouco flexíveis

Compatibilidade entre interfaces

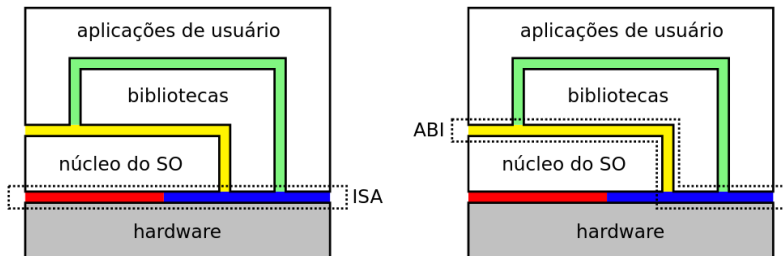


Figura: Interfaces de sistema ISA e ABI. Fonte: CM 31

Compatibilidade entre interfaces

- ▶ um sistema operacional só funciona sobre um ISA específico
- ▶ uma biblioteca só funciona sobre uma ABI específica
- ▶ uma aplicação só funciona sobre as ABI e API específicas

Compatibilidade entre interfaces

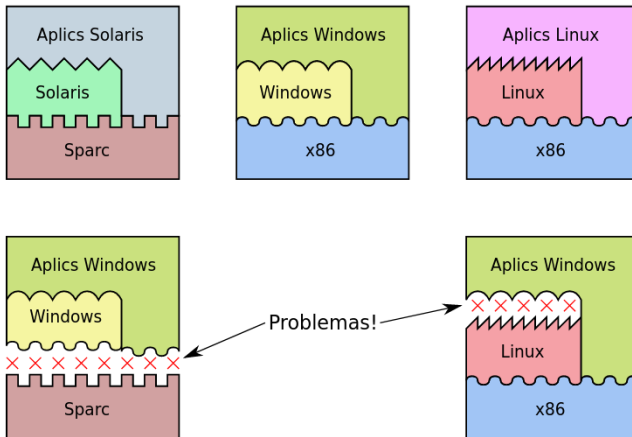


Figura: Problemas de compatibilidade entre interfaces. Fonte: CM 31

Compatibilidade entre interfaces

Questões a serem consideradas:

- ▶ portabilidade
- ▶ inovação
- ▶ otimização intercomponentes

Sumário

Contexto

História

Interfaces

Virtualização de interfaces

Tipos de máquinas virtuais

Camada de virtualização

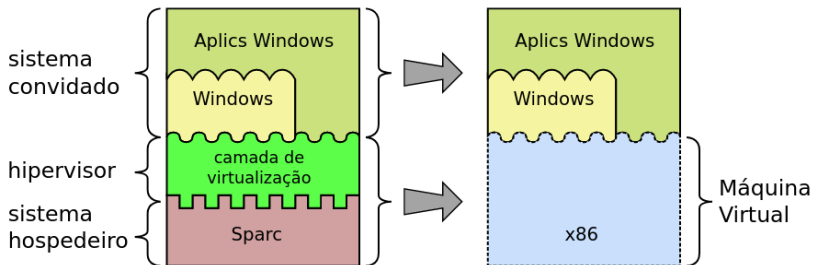


Figura: A camada de virtualização e uma máquina virtual. Fonte: CM 31

Emulação vs virtualização

Emulação é um tipo de virtualização: quando o hipervisor virtualiza uma interface integralmente

Emulação vs virtualização

Emulação é um tipo de virtualização: quando o hipervisor virtualiza uma interface integralmente

Máquina virtual Java e bytecodes

Possibilidades

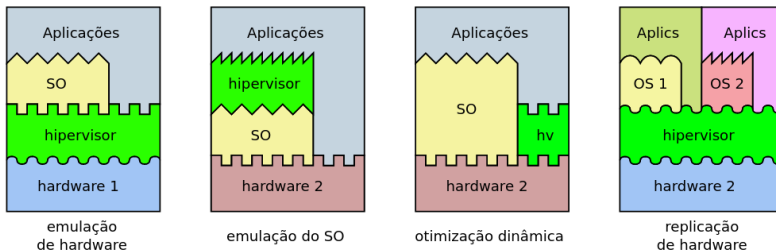


Figura: Possibilidades de virtualização. Fonte: CM 31

Abstração vs virtualização

Abstração consiste em visões de alto nível, esconder detalhes e desacoplar dos detalhes

Abstração vs virtualização

Abstração consiste em visões de alto nível, esconder detalhes e desacoplar dos detalhes

Já na virtualização:

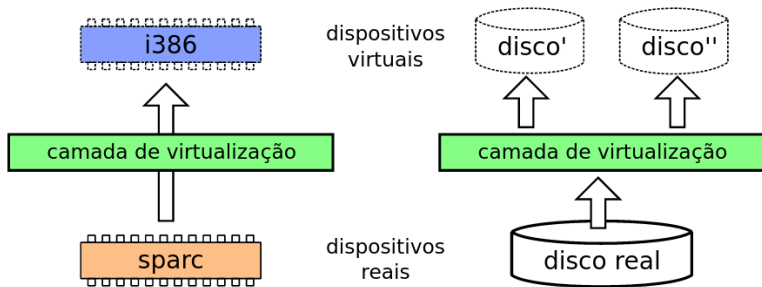


Figura: Virtualização de hardware. Fonte: CM 31

Abstração vs virtualização

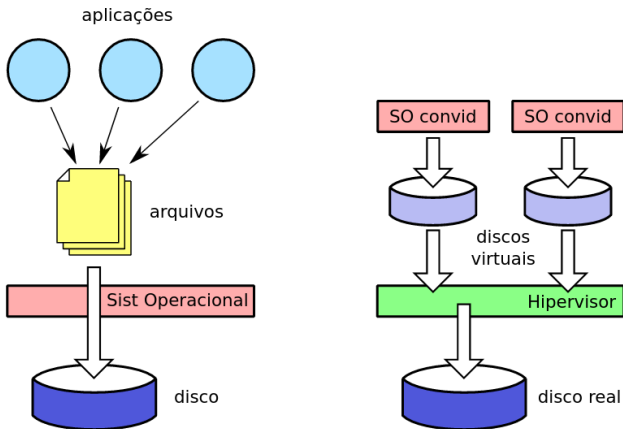


Figura: Abstração vs virtualização de um disco rígido. Fonte: CM 31

Sumário

Contexto

História

Interfaces

Virtualização de interfaces

Tipos de máquinas virtuais

Categorias de máquinas virtuais

- ▶ de sistema: emulação completa de hardware. *KVM, VMware, VirtualBox*

Categorias de máquinas virtuais

- ▶ de sistema: emulação completa de hardware. *KVM, VMware, VirtualBox*
- ▶ de sistema operacional: vários espaços de usuário sobre o mesmo núcleo. *Docker, FreeBSD jails*

Categorias de máquinas virtuais

- ▶ de sistema: emulação completa de hardware. *KVM, VMware, VirtualBox*
- ▶ de sistema operacional: vários espaços de usuário sobre o mesmo núcleo. *Docker, FreeBSD jails*
- ▶ de processo: execução de um processo convidado. *MV Java, Valgrind*

Categorias de máquinas virtuais

- ▶ de sistema: emulação completa de hardware. *KVM, VMware, VirtualBox*
- ▶ de sistema operacional: vários espaços de usuário sobre o mesmo núcleo. *Docker, FreeBSD jails*
- ▶ de processo: execução de um processo convidado. *MV Java, Valgrind*

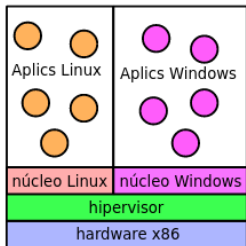
Categorias de máquinas virtuais

- ▶ interfaces equivalentes. *VMware*

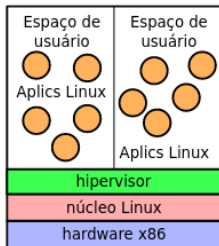
Categorias de máquinas virtuais

- ▶ interfaces equivalentes. *VMware*
- ▶ interfaces distintas. *VM Java, QEMU*

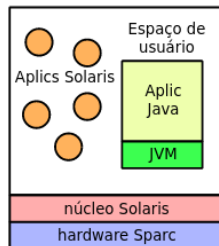
Categorias de máquinas virtuais



VM de sistema
(hardware)



VM de sistema
operacional



VM de processo

Figura: Categorias de máquinas virtuais. Fonte: CM 32

Máquinas virtuais de sistema

- ▶ interface de hardware completa

Máquinas virtuais de sistema

- ▶ interface de hardware completa
- ▶ cada sistema convidado executa de forma isolada e independente

Máquinas virtuais de sistema

- ▶ interface de hardware completa
- ▶ cada sistema convidado executa de forma isolada e independente
- ▶ ilusão de executar sozinho

Máquinas virtuais de sistema

- ▶ interface de hardware completa
- ▶ cada sistema convidado executa de forma isolada e independente
- ▶ ilusão de executar sozinho
- ▶ interface fornecida pelo hipervisor pode ser equivalente ou distinta ao hardware nativo

Máquinas virtuais de sistema

- ▶ interface de hardware completa
- ▶ cada sistema convidado executa de forma isolada e independente
- ▶ ilusão de executar sozinho
- ▶ interface fornecida pelo hipervisor pode ser equivalente ou distinta ao hardware nativo
- ▶ esse foi o primeiro tipo que surgiu (perdeu força e depois voltou)
- ▶ interação só acontece se o sistema de virtualização criar esse mecanismo

Máquinas virtuais de sistema

Hipervisor nativo ou do tipo I:

- ▶ executa diretamente sobre o hardware

Máquinas virtuais de sistema

Hipervisor nativo ou do tipo I:

- ▶ executa diretamente sobre o hardware
- ▶ virtualiza recursos de hardware

Máquinas virtuais de sistema

Hipervisor nativo ou do tipo I:

- ▶ executa diretamente sobre o hardware
- ▶ virtualiza recursos de hardware
- ▶ forma mais antiga de virtualização

Máquinas virtuais de sistema

Hipervisor nativo ou do tipo I:

- ▶ executa diretamente sobre o hardware
- ▶ virtualiza recursos de hardware
- ▶ forma mais antiga de virtualização
- ▶ *OBM OS/370, VMware ESX Server, Xen*



Máquinas virtuais de sistema

Hipervisor convidado ou do tipo II:

- ▶ executa como um processo sobre um sistema operacional

Máquinas virtuais de sistema

Hipervisor convidado ou do tipo II:

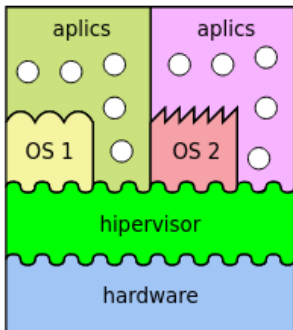
- ▶ executa como um processo sobre um sistema operacional
- ▶ usa os recursos do sistema operacional nativo para oferecer recursos virtuais

Máquinas virtuais de sistema

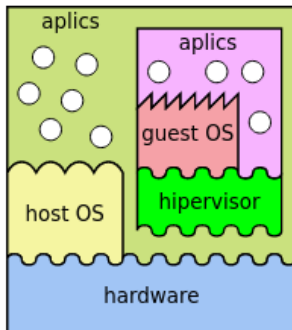
Hipervisor convidado ou do tipo II:

- ▶ executa como um processo sobre um sistema operacional
- ▶ usa os recursos do sistema operacional nativo para oferecer recursos virtuais
- ▶ *VMware Workstation, KVM, VirtualBox*

Máquinas virtuais de sistema



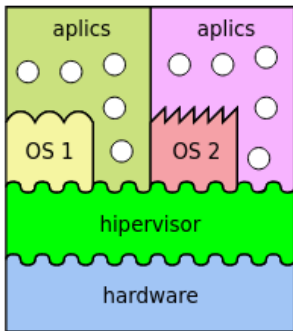
hipervisor nativo



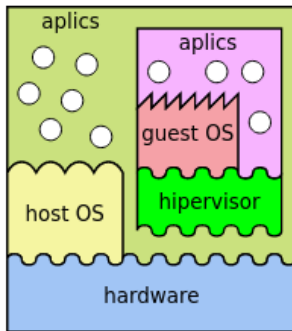
hipervisor convidado

Figura: Arquiteturas de máquinas virtuais de sistema. Fonte: CM 32

Máquinas virtuais de sistema



hipervisor nativo



hipervisor convidado

Figura: Arquiteturas de máquinas virtuais de sistema. Fonte: CM 32

Flexibilidade e desempenho

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

- ▶ virtualização do espaço de usuário (*userspace*)

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

- ▶ virtualização do espaço de usuário (*userspace*)
- ▶ mesmo núcleo do sistema operacional

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

- ▶ virtualização do espaço de usuário (*userspace*)
- ▶ mesmo núcleo do sistema operacional
- ▶ domínios ou contêineres: cada domínio tem seus próprios PID, UID, diretórios, semáforos, ...

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

- ▶ virtualização do espaço de usuário (*userspace*)
- ▶ mesmo núcleo do sistema operacional
- ▶ domínios ou contêineres: cada domínio tem seus próprios PID, UID, diretórios, semáforos, ...
- ▶ aloca uma parcela dos recursos do SO nativo para cada domínio: processamento, memória, espaço em disco, ...

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

- ▶ virtualização do espaço de usuário (*userspace*)
- ▶ mesmo núcleo do sistema operacional
- ▶ domínios ou contêineres: cada domínio tem seus próprios PID, UID, diretórios, semáforos, ...
- ▶ aloca uma parcela dos recursos do SO nativo para cada domínio: processamento, memória, espaço em disco, ...
- ▶ alguns recursos são virtualizados para garantir isolamento: interface de rede

Máquinas virtuais de sistema operacional

Queremos o isolamento, mas sem o custo sobre o desempenho

Vários servidores (DNS, arquivos, email, web) em uma mesma máquina física

- ▶ virtualização do espaço de usuário (*userspace*)
- ▶ mesmo núcleo do sistema operacional
- ▶ domínios ou contêineres: cada domínio tem seus próprios PID, UID, diretórios, semáforos, ...
- ▶ aloca uma parcela dos recursos do SO nativo para cada domínio: processamento, memória, espaço em disco, ...
- ▶ alguns recursos são virtualizados para garantir isolamento: interface de rede
- ▶ interação entre processos de um mesmo domínio, mas não entre domínios

Máquinas virtuais de sistema operacional

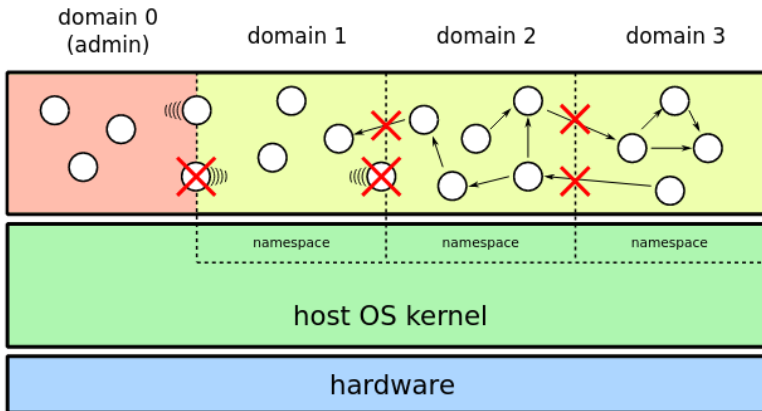


Figura: Máquinas virtuais de sistema operacional. Fonte: CM 32

Máquinas virtuais de sistema operacional

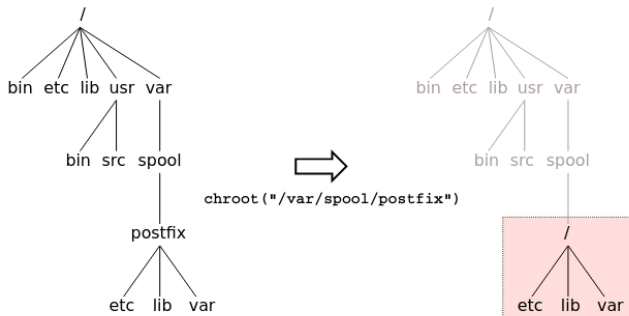


Figura: Funcionamento do `chroot`. Fonte: CM 32

Máquinas virtuais de sistema operacional

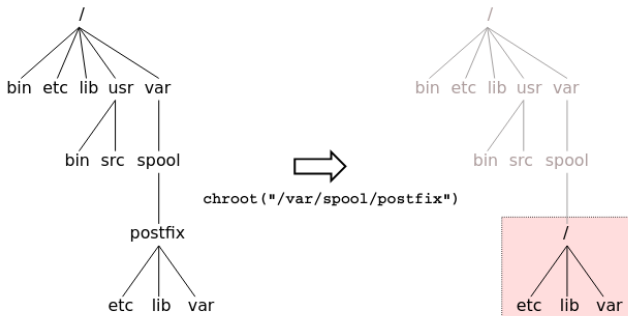


Figura: Funcionamento do `chroot`. Fonte: CM 32

Isolar serviços de rede (DNS, email, ...) para proteger o sistema nativo

Máquinas virtuais de sistema operacional

FreeBSD Jails:

Máquinas virtuais de sistema operacional

FreeBSD Jails:

- ▶ só vê processos e recursos dentro da própria cela

Máquinas virtuais de sistema operacional

FreeBSD Jails:

- ▶ só vê processos e recursos dentro da própria cela
- ▶ só interage com processos dentro da mesma cela



Máquinas virtuais de sistema operacional

FreeBSD Jails:

- ▶ só vê processos e recursos dentro da própria cela
- ▶ só interage com processos dentro da mesma cela
- ▶ árvore de diretórios própria; não pode montar sistemas de arquivos

Máquinas virtuais de sistema operacional

FreeBSD Jails:

- ▶ só vê processos e recursos dentro da própria cela
- ▶ só interage com processos dentro da mesma cela
- ▶ árvore de diretórios própria; não pode montar sistemas de arquivos
- ▶ possui um endereço de rede associado; configuração da rede não pode ser alterada

Máquinas virtuais de sistema operacional

FreeBSD Jails:

- ▶ só vê processos e recursos dentro da própria cela
- ▶ só interage com processos dentro da mesma cela
- ▶ árvore de diretórios própria; não pode montar sistemas de arquivos
- ▶ possui um endereço de rede associado; configuração da rede não pode ser alterada
- ▶ não pode alterar o núcleo do sistema operacional

Máquinas virtuais de sistema operacional

FreeBSD Jails:

- ▶ só vê processos e recursos dentro da própria cela
- ▶ só interage com processos dentro da mesma cela
- ▶ árvore de diretórios própria; não pode montar sistemas de arquivos
- ▶ possui um endereço de rede associado; configuração da rede não pode ser alterada
- ▶ não pode alterar o núcleo do sistema operacional

LXC, Docker

Máquinas virtuais de processo

- ▶ executa um processo individual, é criada sob demanda

Máquinas virtuais de processo

- ▶ executa um processo individual, é criada sob demanda
- ▶ hipervisor + aplicação = um único processo no sistema hospedeiro



Máquinas virtuais de processo

- ▶ executa um processo individual, é criada sob demanda
- ▶ hipervisor + aplicação = um único processo no sistema hospedeiro
- ▶ comunicação e coordenação pode ocorrer: semáforos, *pipes*, mensagens, sistema de arquivos, . . .



Máquinas virtuais de processo

- ▶ executa um processo individual, é criada sob demanda
- ▶ hipervisor + aplicação = um único processo no sistema hospedeiro
- ▶ comunicação e coordenação pode ocorrer: semáforos, *pipes*, mensagens, sistema de arquivos, . . .
- ▶ não possui isolamento!



Máquinas virtuais de processo

- ▶ executa um processo individual, é criada sob demanda
- ▶ hipervisor + aplicação = um único processo no sistema hospedeiro
- ▶ comunicação e coordenação pode ocorrer: semáforos, *pipes*, mensagens, sistema de arquivos, . . .
- ▶ não possui isolamento!
- ▶ se a interface de hardware for equivalente: boa parte do processo executa diretamente

Máquinas virtuais de processo

- ▶ sistemas operacionais multitarefas

Máquinas virtuais de processo

- ▶ sistemas operacionais multitarefas(!): processador virtual, memória virtual, recursos físicos

Máquinas virtuais de processo

- ▶ sistemas operacionais multitarefas(!): processador virtual, memória virtual, recursos físicos
- ▶ tradutores dinâmicos: otimização



Máquinas virtuais de processo

- ▶ sistemas operacionais multitarefas(!): processador virtual, memória virtual, recursos físicos
- ▶ tradutores dinâmicos: otimização
- ▶ depuradores de memória: *Valgrind*
- ▶ Desempenho pior

Máquinas virtuais de processo

- ▶ sistemas operacionais multitarefas(!): processador virtual, memória virtual, recursos físicos
- ▶ tradutores dinâmicos: otimização
- ▶ depuradores de memória: *Valgrind*
- ▶ Desempenho pior: uma solução é compilação *Just-in-time* (JIT) (cache de instruções frequentes)

Máquinas virtuais de processo

- ▶ sistemas operacionais multitarefas(!): processador virtual, memória virtual, recursos físicos
- ▶ tradutores dinâmicos: otimização
- ▶ depuradores de memória: *Valgrind*
- ▶ Desempenho pior: uma solução é compilação *Just-in-time* (JIT) (cache de instruções frequentes): Java, Lua, Python, C#

Máquinas virtuais de processo

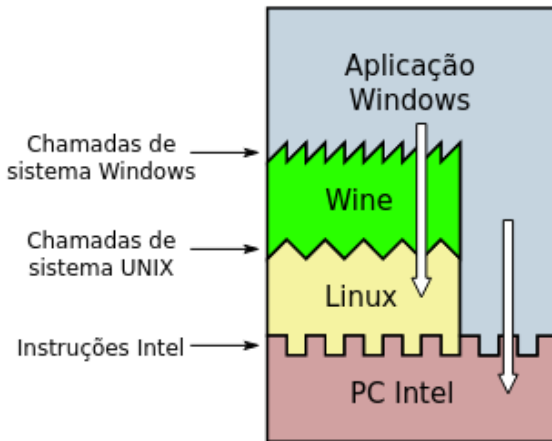


Figura: Funcionamento do Wine. Fonte: CM 32