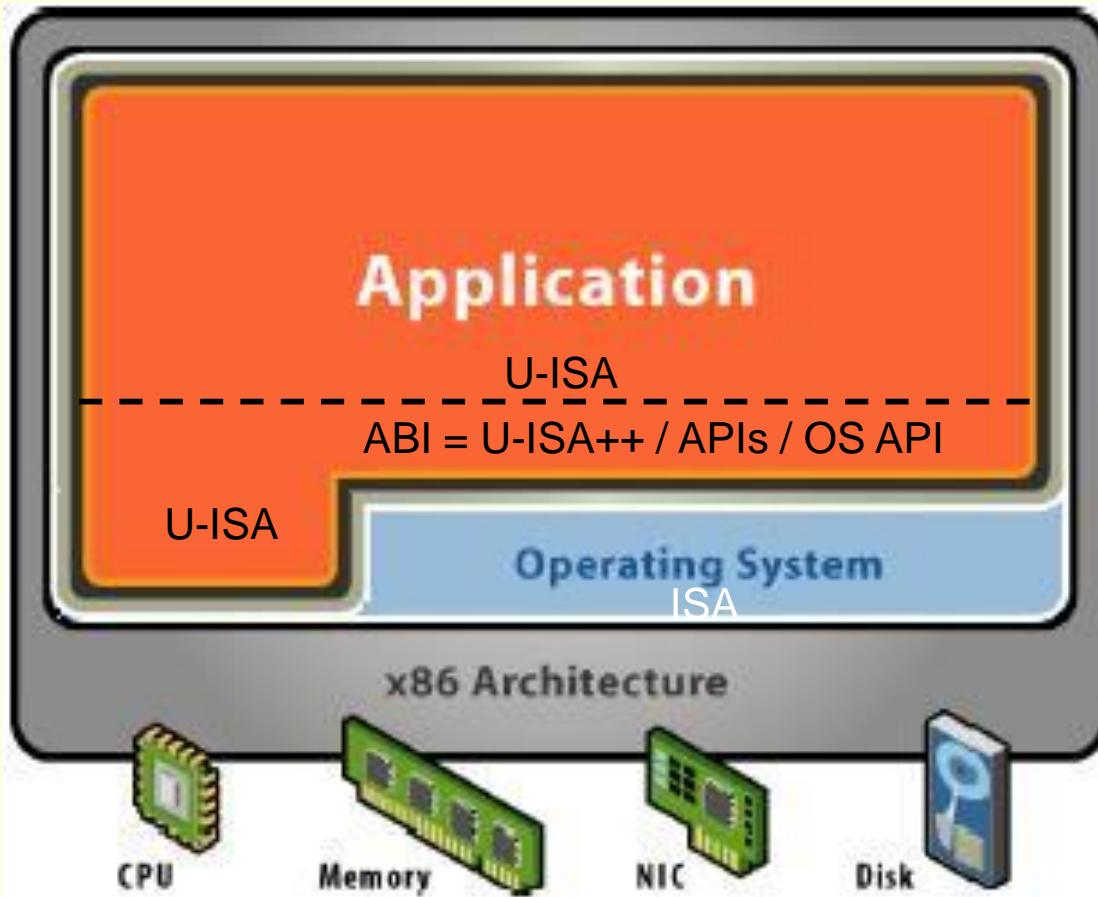


Fundamentos de Sistemas de Operação

Unix *Windows NT* *Netware* *MacOS* *DOS/VIS* *Vax/VMS*
Linux Solaris *HP/UX* *AIX* *Mach*
Chorus

A idade do ágil:
2. *Virtualização e VMs: Parte II*

Aplicação, SO e Máquina (1)



ISA: Instruction Set Architecture

U-ISA: ISA disponível p/o utilizador

U-ISA++: U-ISA mais a “syscall”/trap

APIs: Linguagens, bibliotecas e APIs do SO

ABI: Application Binary Interface

Aplicação, SO e Máquina (2)

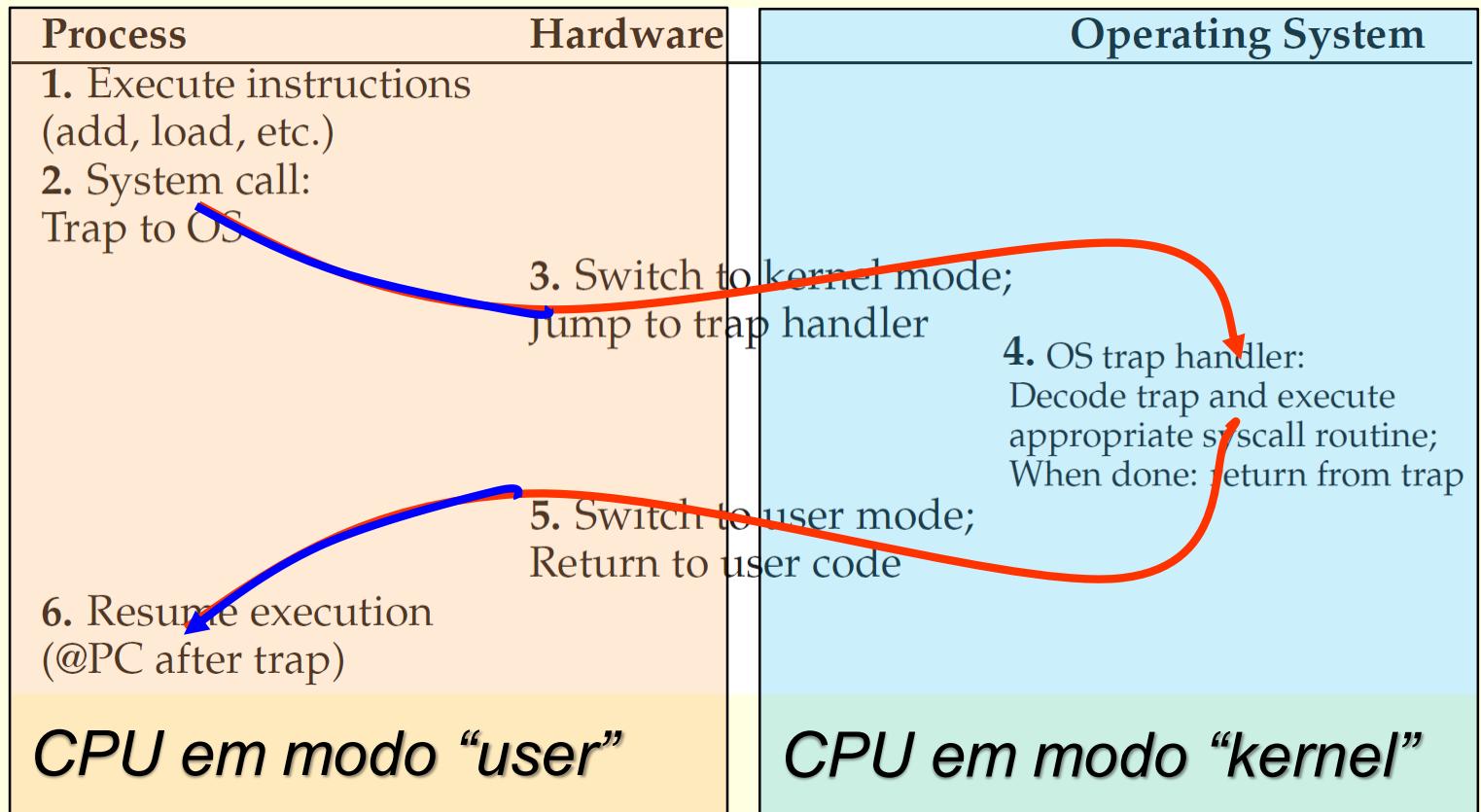
□ A aplicação,

- Executa instruções U-ISA (`mov`, `push`, `call`, `jmp`, `add`, etc.) que não requerem intervenção do SO para se completarem (desde que, naturalmente, não “façam” nada de ilegal)
- Executa a instrução `trap` / `int` para solicitar serviços ao SO
- Usa (em runtime) as APIs oferecidas pelas linguagens bibliotecas adicionais (e.g., sockets), e a API do próprio SO
- Se a aplicação obedece ao standard ABI, pode até ser copiada (em formato binário/executável) para outro SO distinto e executada lá

□ O SO

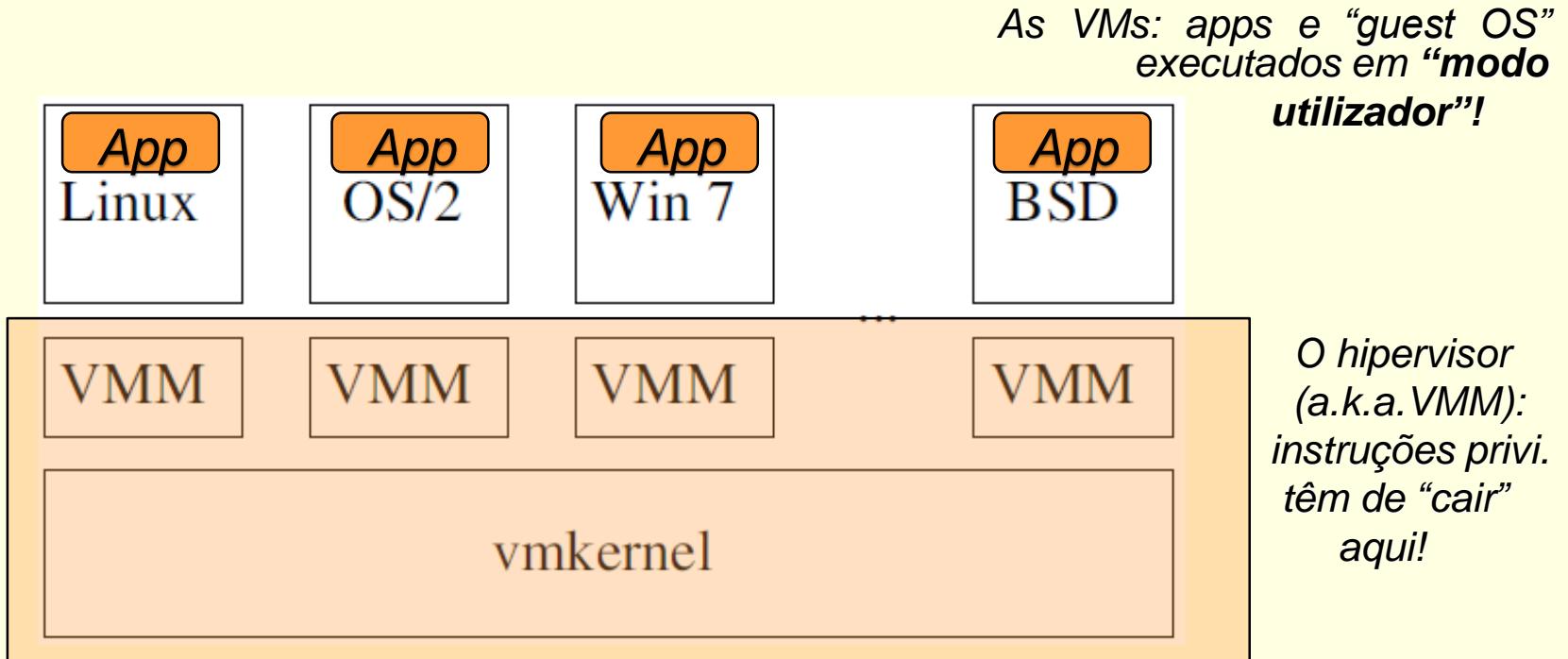
- Pode usar o ISA completo para executar operações sobre o hardware

Aplicação, SO e Máquina (3)

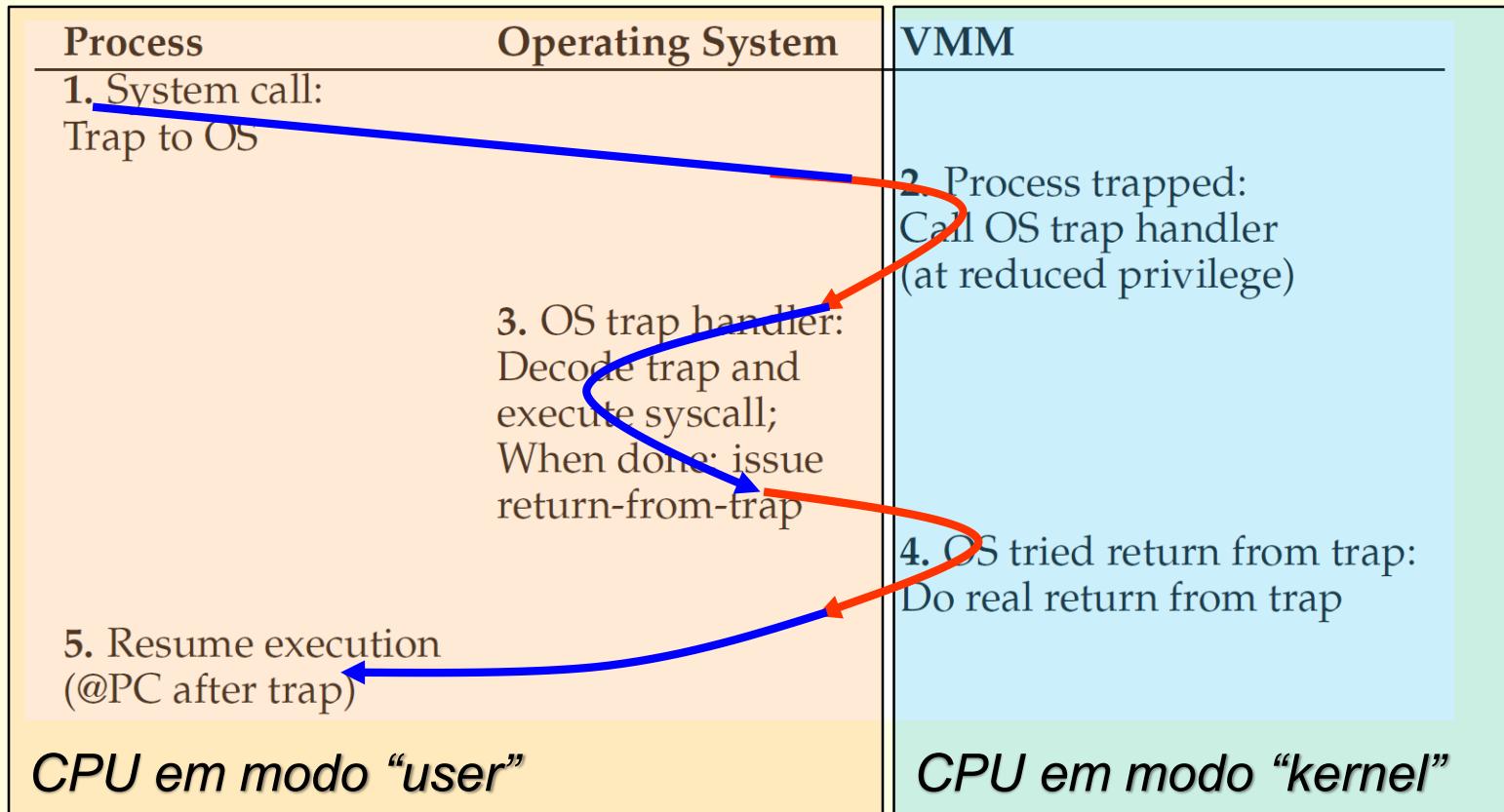


Aplicação, SO, VMM e Máquina (1)

Unix Windows NT Netware Macos DOS/VX Vax/VMS
Linux Solaris HP/UX AIX Mach
Chorus



Aplicação, SO, VMM e Máquina (2)



Afinal, o que é um vCPU?

- Um vCPU (CPU virtual),

- É um “parente próximo” do “CPU real”; podemos *imaginar* que é representado no VMM por uma estrutura de dados com “registos” (genéricos, de endereçamento – *sp, *bp, PTBR, ... – “flags”, etc.).
- Cada VMM é representado no hipervisor por estruturas de dados (“funciona” como um processo que corre no hipervisor). Portanto, a representação do CPU na VMM é uma “imagem” do CPU físico, tal qual como numa máquina real cada processo tem uma representação do “seu” CPU (“registos” (genéricos, de endereçamento – *sp, *bp, PTBR, ... – “flags”, etc.)...)
- Só que... **no 1º caso**, para um processo que corre no “guest OS” o PTBR aponta a Tabela de Páginas (PT) de tradução de endereços virtuais do processo em... endereços virtuais da VMM!
- E **no 2º caso**, aponta a PT de tradução de endereços virtuais da VMM em endereços virtuais da máquina física!

Paginação: guest→VMM e VMM→real

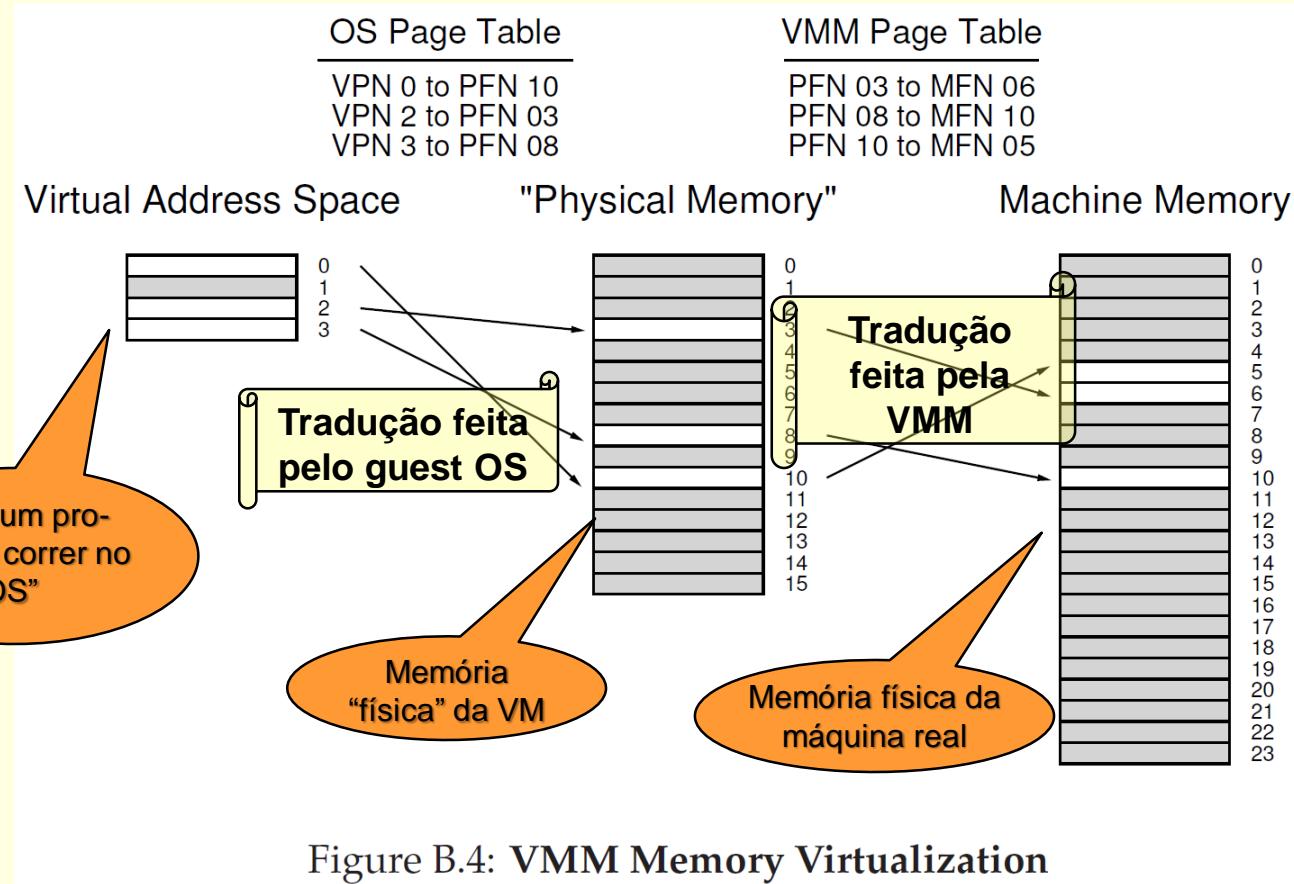


Figure B.4: VMM Memory Virtualization

Paginação: processo → SO → real

Process	Operating System
1. Load from memory: TLB miss: Trap	2. OS TLB miss handler: Extract VPN from VA; Do page table lookup; If present and valid: get PFN, update TLB; Return from trap
3. Resume execution (@PC of trapping instruction); Instruction is retried; Results in TLB hit	

Figure B.5: **TLB Miss Flow without Virtualization**

Paginação: processo → gSO → VMM → real



Process	Operating System	Virtual Machine Monitor
1. Load from mem TLB miss: Trap		2. VMM TLB miss handler: Call into OS TLB handler (reducing privilege)
	3. OS TLB miss handler: Extract VPN from VA; Do page table lookup; If present and valid, get PFN, update TLB	
		4. Trap handler: Unprivileged code trying to update the TLB; OS is trying to install VPN-to-PFN mapping; Update TLB instead with VPN-to-MFN (privileged); Jump back to OS (reducing privilege)
	5. Return from trap	6. Trap handler: Unprivileged code trying to return from a trap; Return from trap
7. Resume execution (@PC of instruction); Instruction is retried; Results in TLB hit		

Figure B.6: TLB Miss Flow with Virtualization

Hardware de suporte à virtualização

□ Desenvolvimentos “recentes” (ideias **muito** sumárias)

- A partir de 2007 a Intel e a AMD, em conjunto, trabalharam em “ajuda” baseada em hardware para suportar virtualização eficiente.
- 1^a fase (2007): Intel VT-x/AMD-V, melhoramentos no CPU
 - Introdução de um novo estado de execução nos CPUs: “guest mode” (podemos renomear o “supervisor ou kernel mode” como “host ou root mode”). Estes modos são acompanhados de duas novas instruções máquina: vmrun (host → guest) e vmexit (guest → host)
 - O “guest mode” é subdividido em “user guest mode” (aplicações que correm no guest) e “kernel guest mode” (SO guest)
- 2^a fase (2010): Intel EPT/AMD-RVI, melhoramentos na MV
 - Introdução dum conjunto adicional de page tables que podem ser manipuladas pelo SO guest para tradução “directa” de endereços sem necessidade de intervenção do VMM