Виртуализация на процесора

проф. д-р инж. Христо Вълчанов

http://cs.tu-varna.bg

Виртуализация на процесора

- Трябва да има много малка или никаква разлика в производителността между виртуална машина и физическият ѝ еквивалент.

Popek, Golberg

Процеси

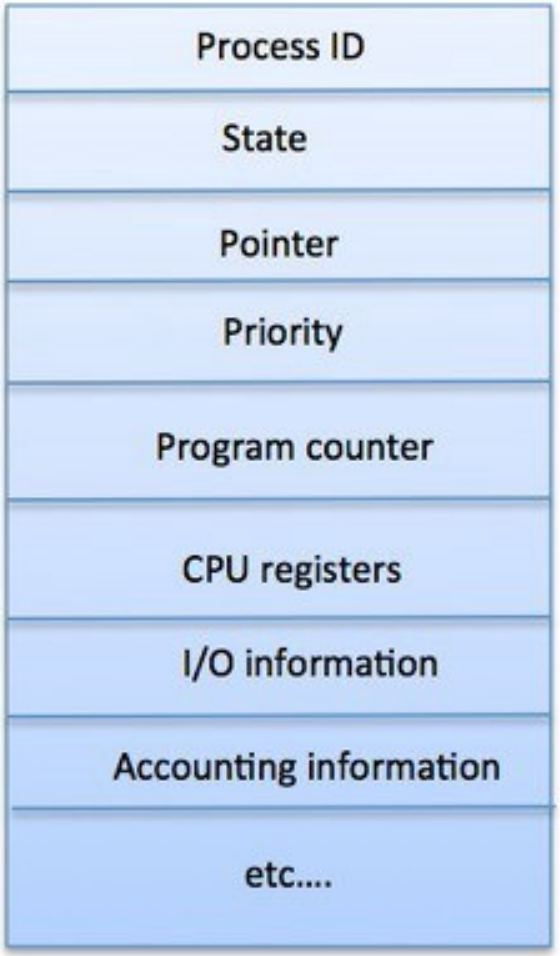
- **Процес** – програма по време на нейното изпълнение.
- Включва множество компонента:
  - Програмен код (text section).
  - Текущата му активност, включваща регистрите на процесора.
  - Стек.
  - Даннова част с глобални променливи.
  - Динамично заемана памет (Heap).

Програма и процес

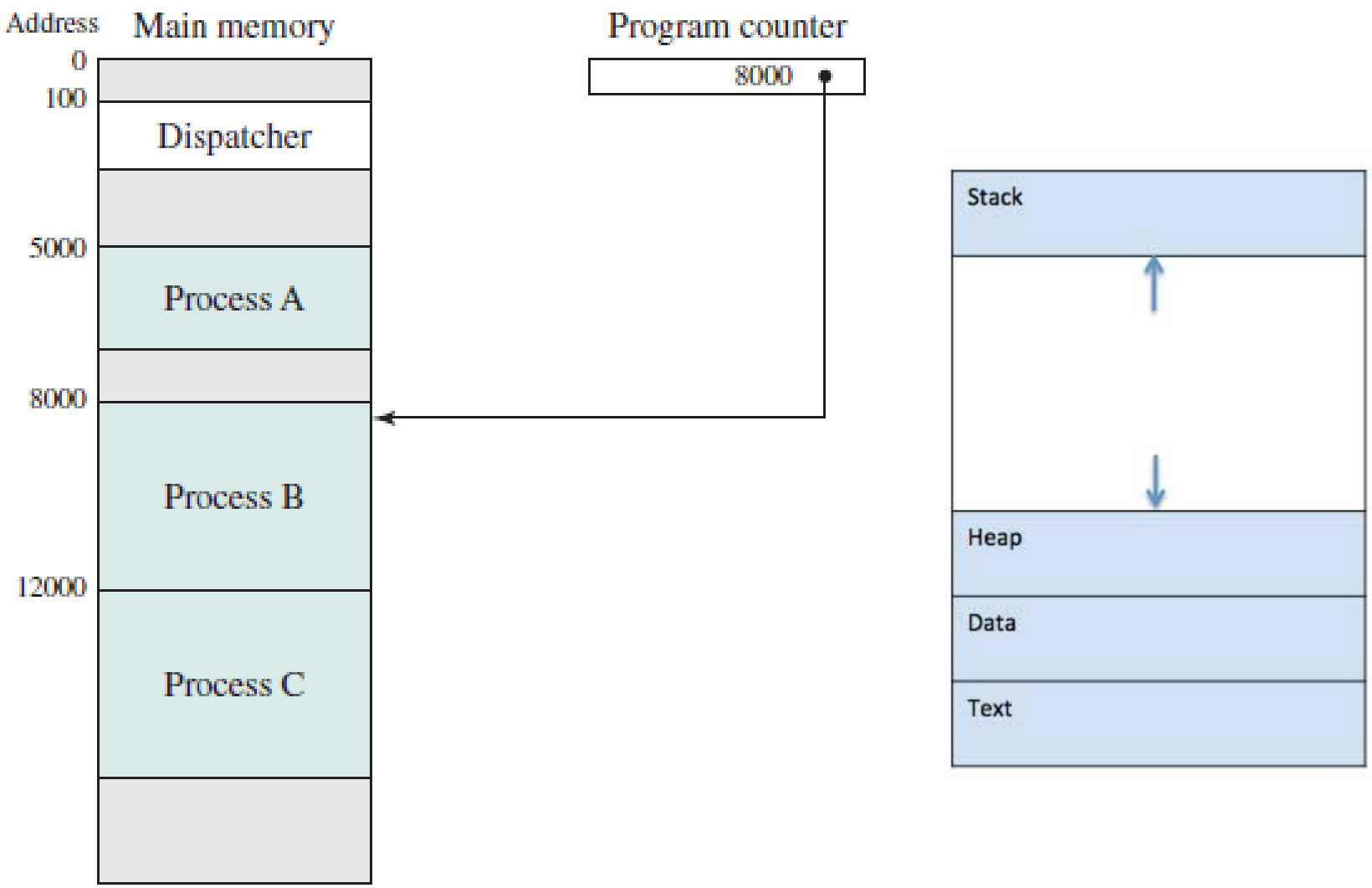
- **Програма** – пасивен компонент, съхранен на диска (изпълним файл).
- **Процес** – активен компонент.
- Програмата става процес, когато изпълнимият файл се зареди в паметта.
- Една програма може да се изпълнява като множество процеси.

Блок за управление на процеса

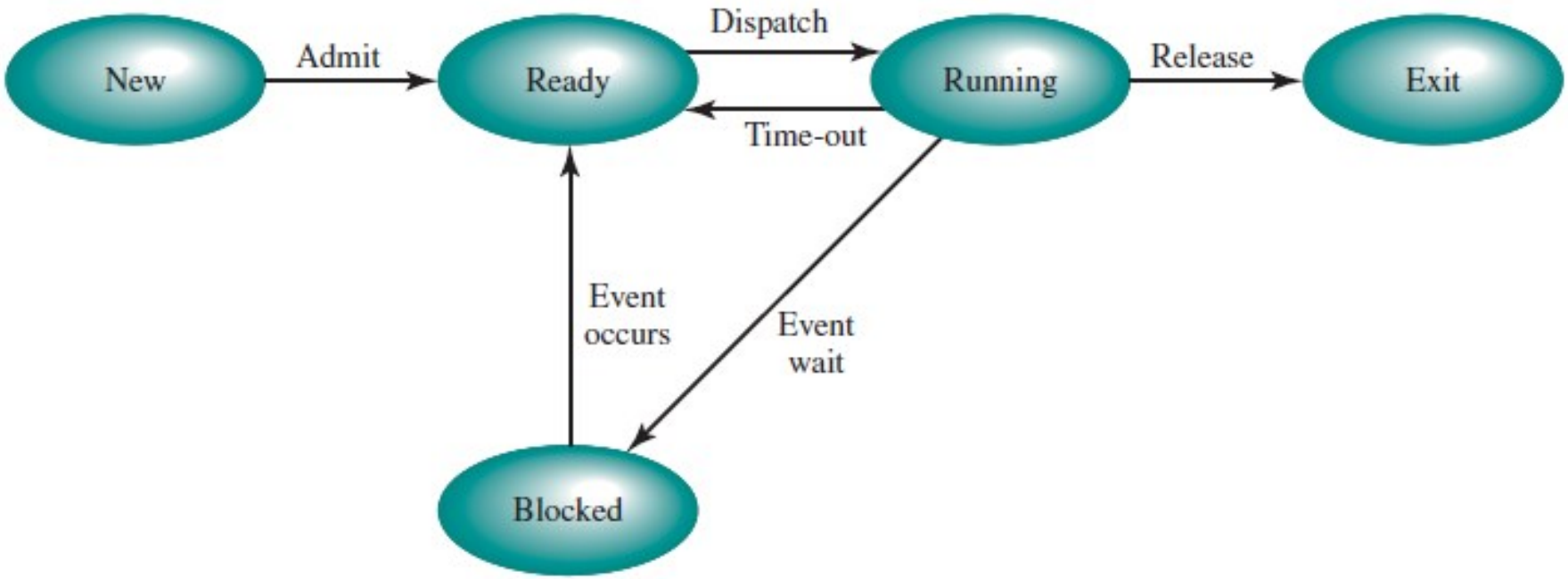
Информация, асоциирана с всеки процес (Process Control Block – PCB).



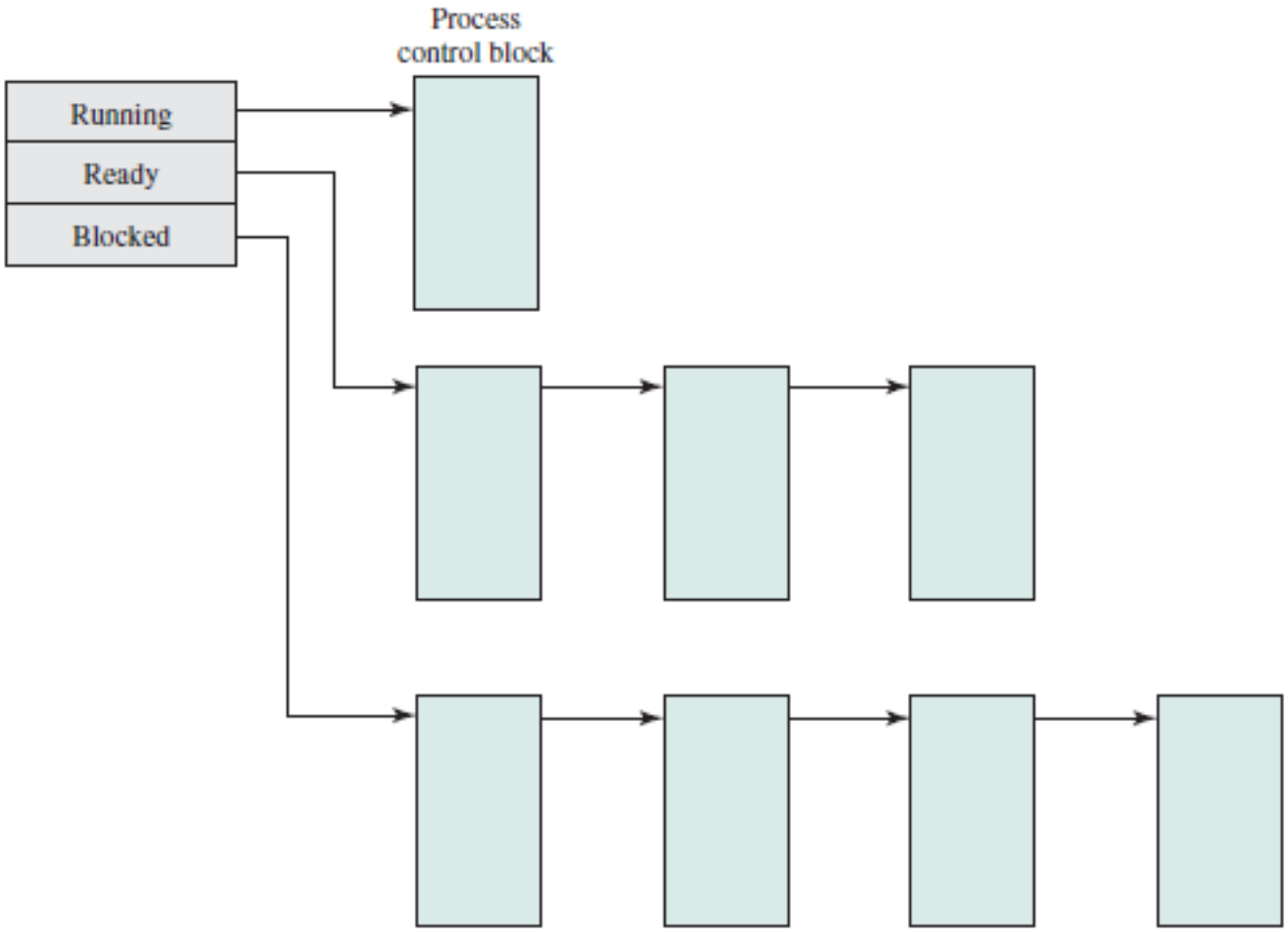
Процеси в паметта



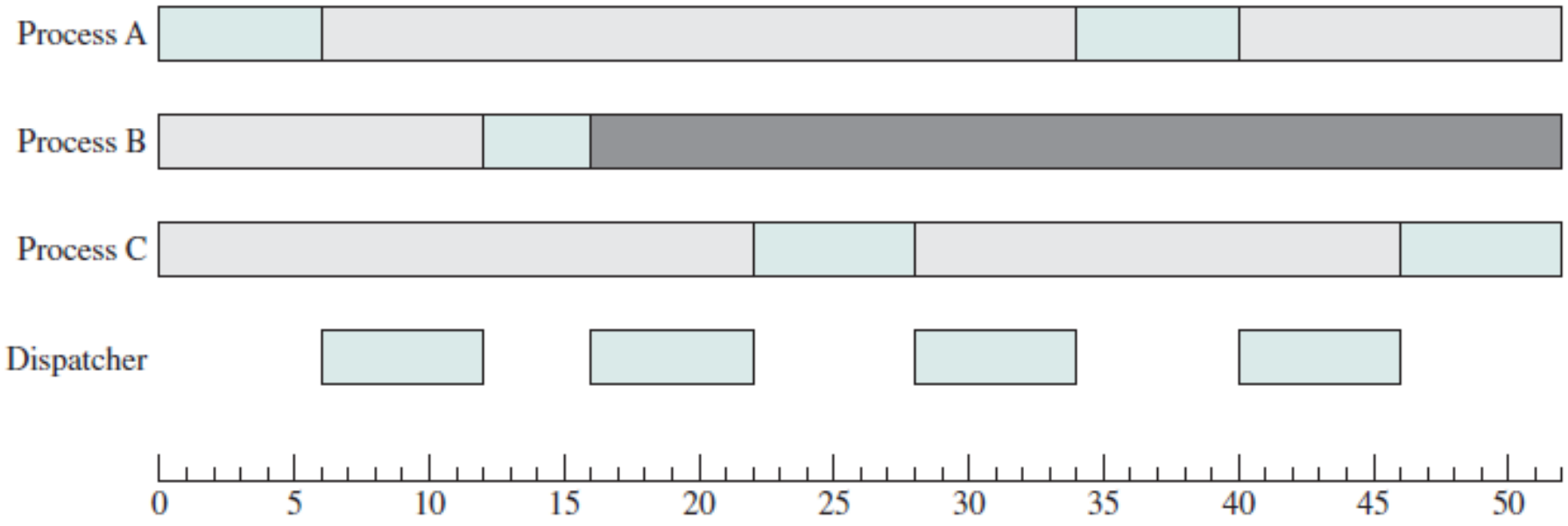
Състояния на процесите



Организация на състоянията на процесите



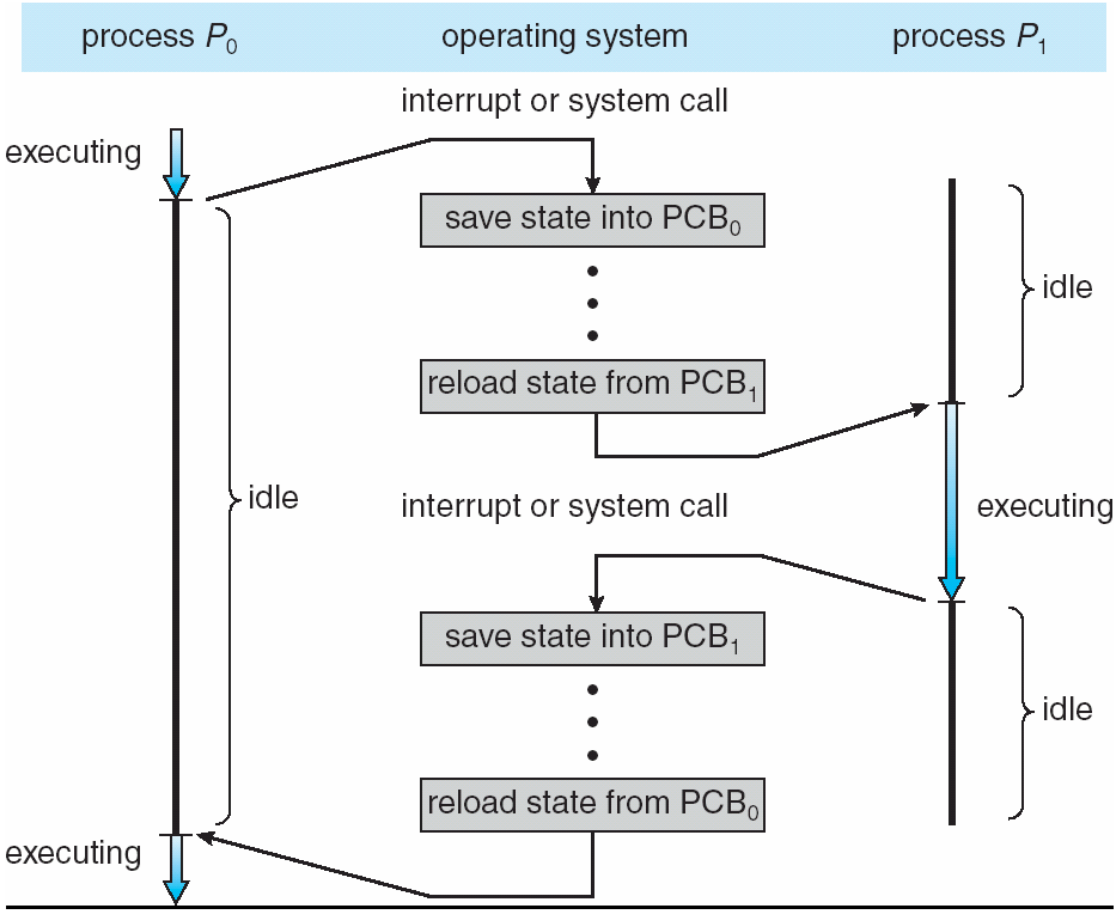
Изпълнение на процесите



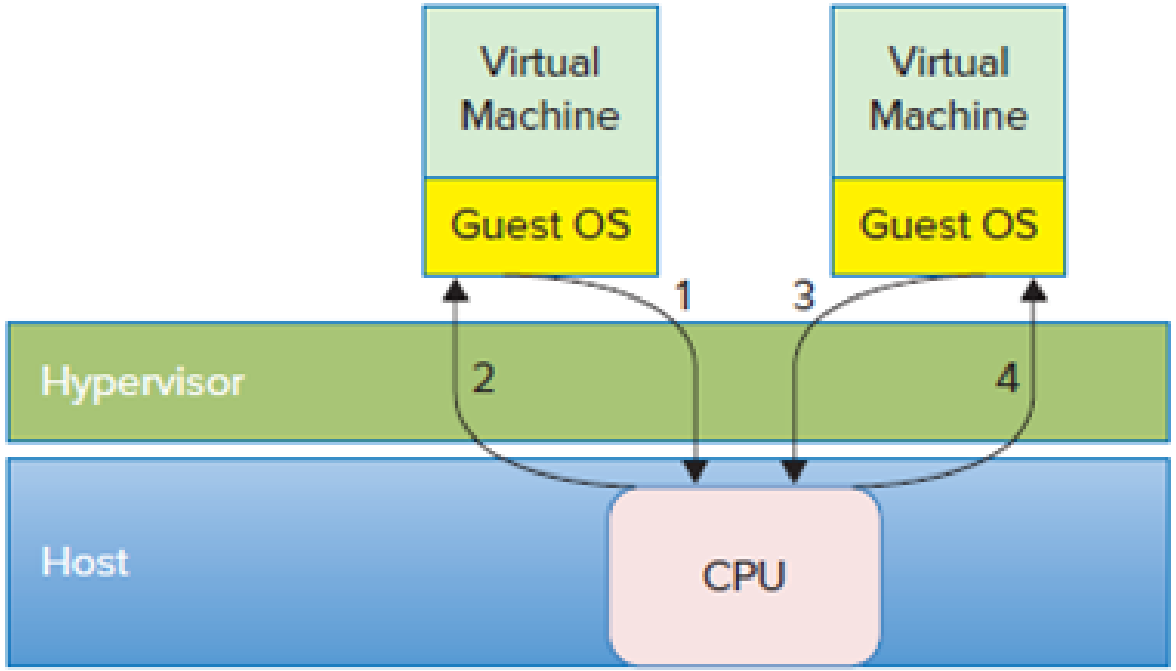
Превключване контекста на процесите

- При превключване към друг процес системата трябва да съхрани **състоянието** на стария процес и да зареди състоянието на новия чрез превключване на контекста.
- Контекстът на процес се съхранява в PCB.
- Превключването на контекста води до системни разходи (**overhead**).

Превключване на процесора между процесите

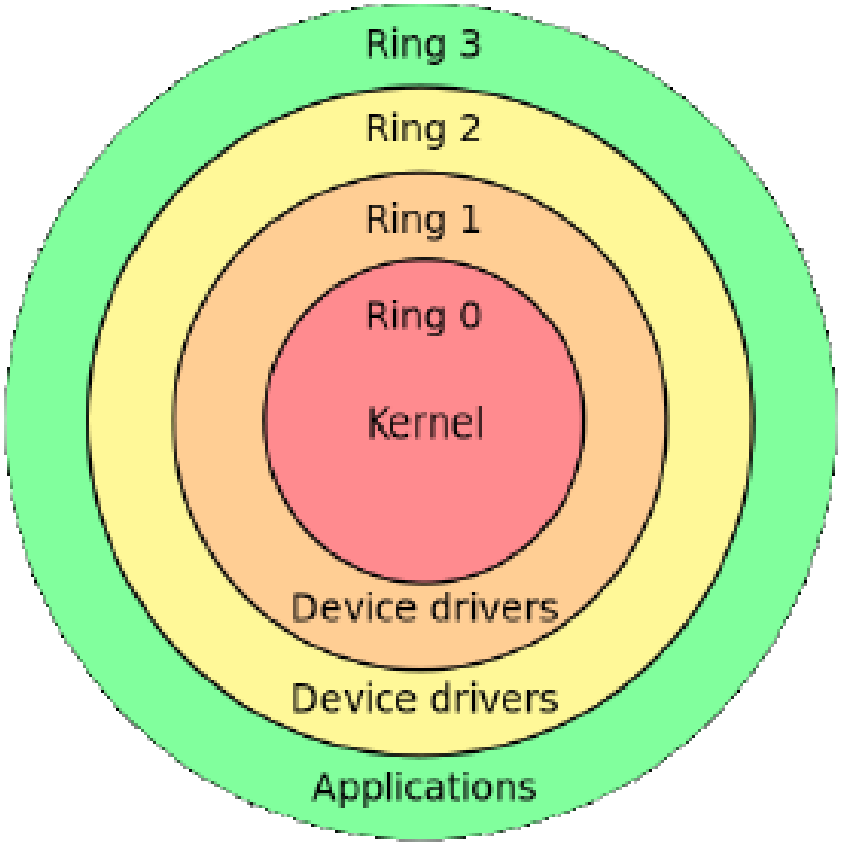


Виртуализация на процесора

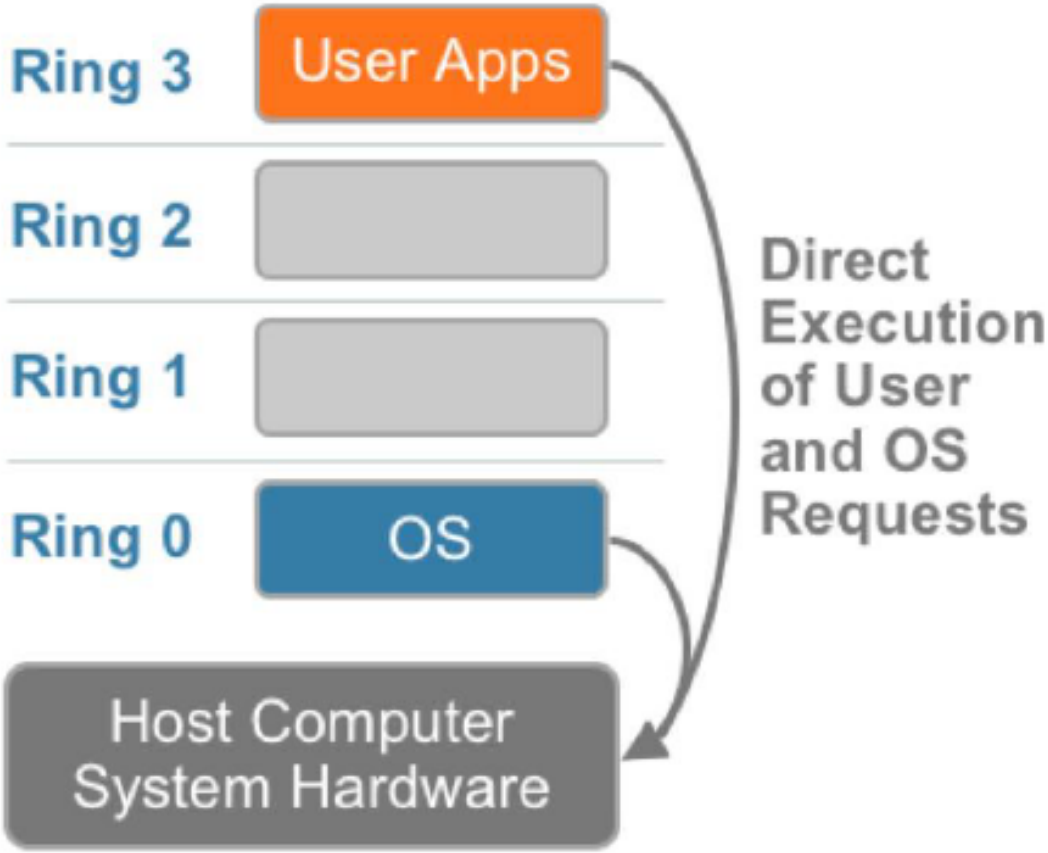


Нива на привилегии на процесора

- X86 архитектура на привилегии: 4 нива
- В повечето съвременни системи нива 1 и 2 не се използват

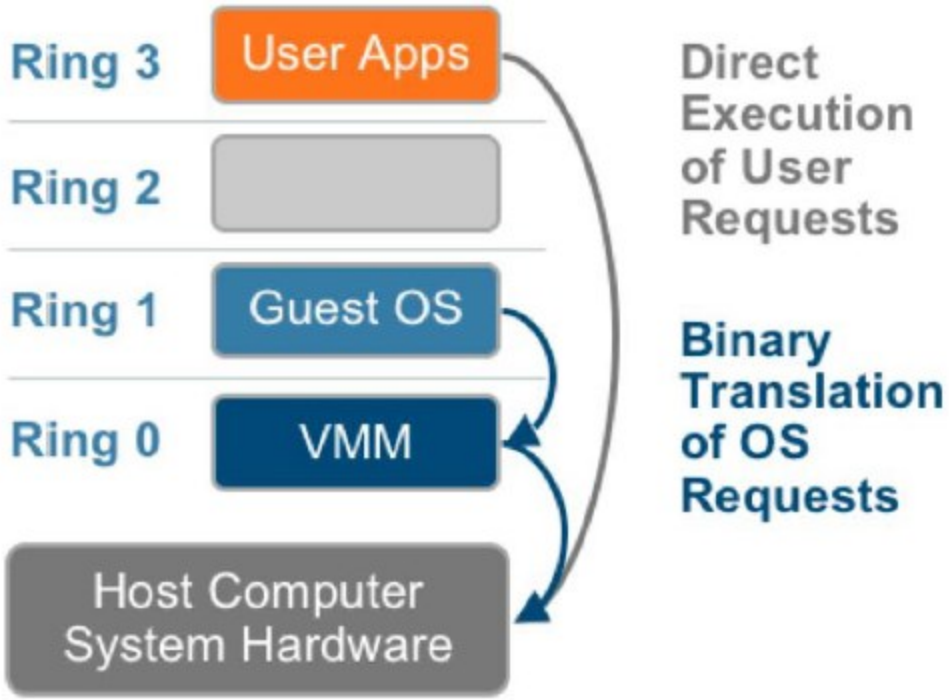


Изпълнение при не-виртуализирана система



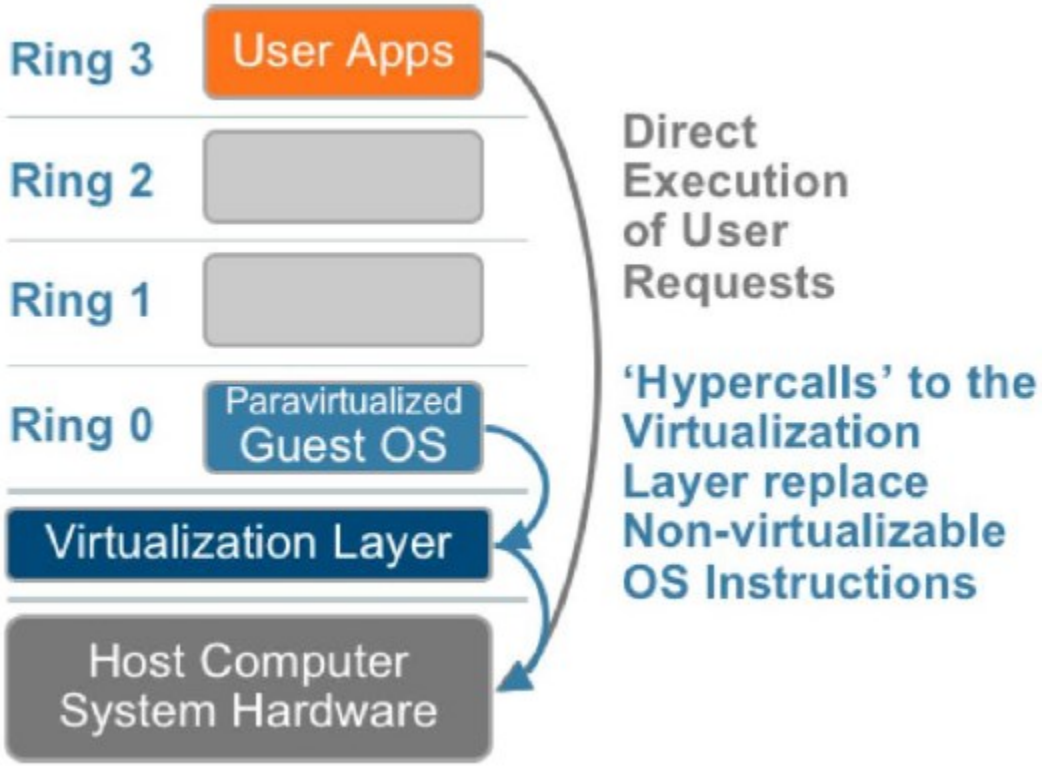
Бинарно транслиране

- Компресия на ринга
- Бинарно транслиране
- Директно изпълнение

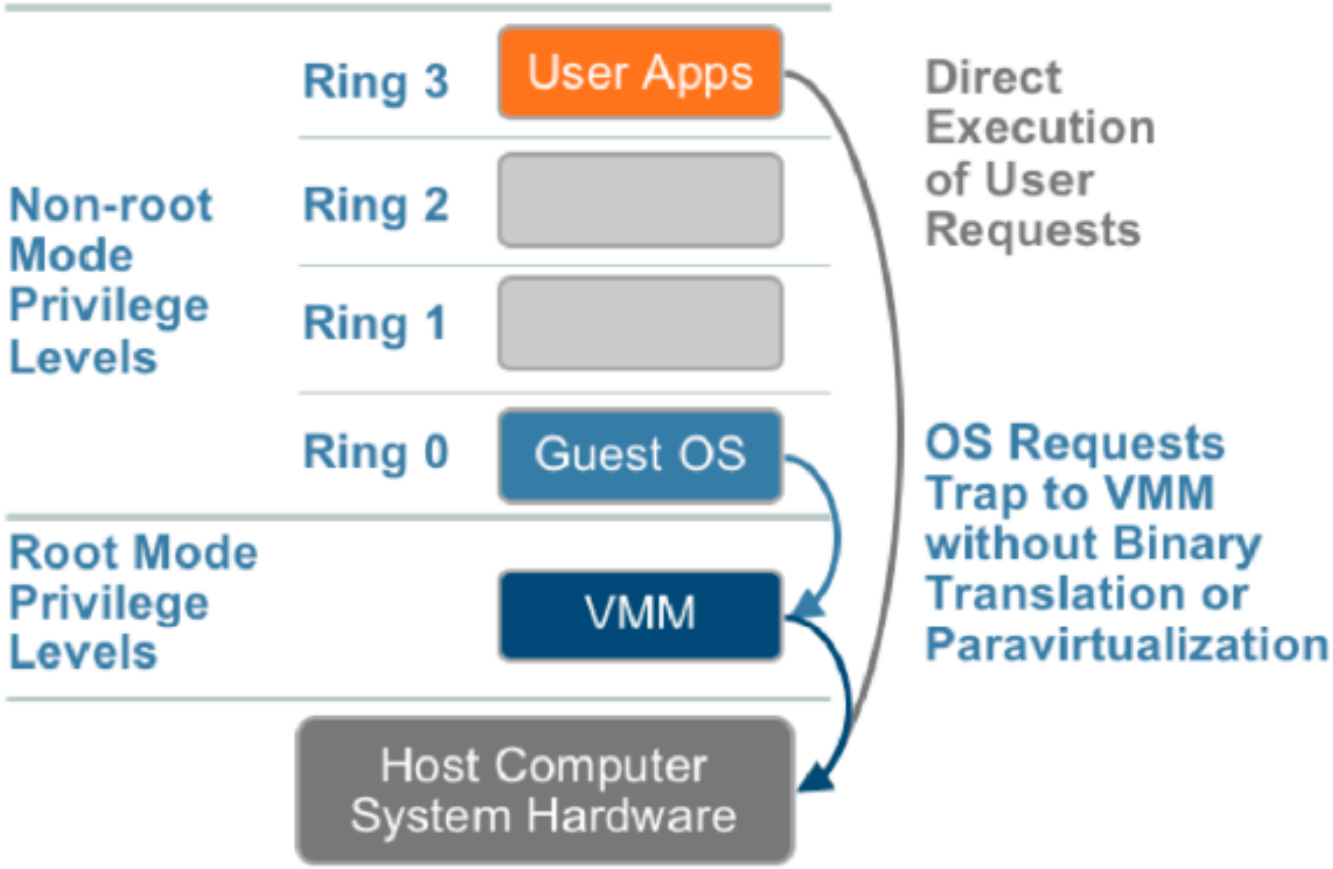


Паравиртуализация

- Изисква модификация на гост ОС

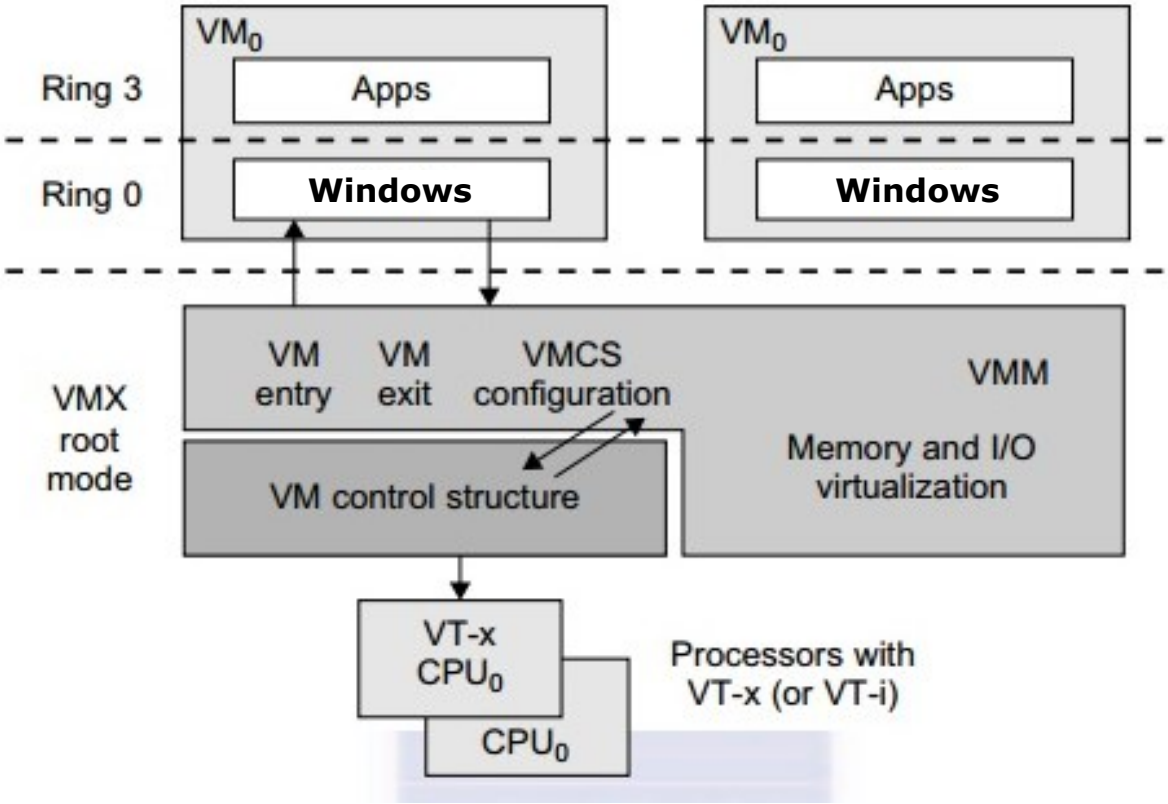


Хардуерно подпомогната виртуализация



Хардуерно подпомогната виртуализация

- Intel: VT-x (x86), VT-i (Itanium)
- AMD: AMD-V (x86)



VMCS

GUEST STATE AREA

CR0	CR3			CR4
DR7				
RSP	RIP			RFLAGS
CS	Selector	Base Address	Segment Limit	Access Right
SS	Selector	Base Address	Segment Limit	Access Right
DS	Selector	Base Address	Segment Limit	Access Right
ES	Selector	Base Address	Segment Limit	Access Right
FS	Selector	Base Address	Segment Limit	Access Right
GS	Selector	Base Address	Segment Limit	Access Right
LDTR	Selector	Base Address	Segment Limit	Access Right
TR	Selector	Base Address	Segment Limit	Access Right
GDTR	Selector	Base Address	Segment Limit	Access Right
IDTR	Selector	Base Address	Segment Limit	Access Right
IA32_DEBUGCTL	IA32_SYSENTER_CS	IA32_SYSENTER_ESP		IA32_SYSENTER_EIP
IA32_PERF_GLOBAL_CTRL	IA32_PAT	IA32_EFER		IA32_BNDCFGS
SMBASE				
Activity state	Interruptibility state			
Pending debug exceptions				
VMCS link pointer				
VMX-preemption timer value				
Page-directory-pointer-table entries	PDPTE0	PDPTE1	PDPTE2	PDPTE3
Guest interrupt status				
PML index				

HOST STATE AREA

CR0	CR3	CR4
RSP	RIP	
CS	Selector	
SS	Selector	
DS	Selector	
ES	Selector	
FS	Selector	Base Address
GS	Selector	Base Address
TR	Selector	Base Address
GDTR	Base Address	
IDTR	Base Address	
IA32_SYSENTER_CS	IA32_SYSENTER_ESP	IA32_SYSENTER_EIP
IA32_PERF_GLOBAL_CTRL	IA32_PAT	IA32_EFER

CONTROL FIELDS

Pin-Based VM-Execution Controls	External-interrupt exiting	NMI exiting	Virtual NMIs	
	Activate VMX-preemption timer	Process posted interrupts		
Primary processor-based VM-execution controls	Interrupt-window exiting	Use TSC offsetting		
	HLT exiting	INVLPG exiting	MWAIT exiting	RDPMC exiting
	RDTSC exiting	CR3-load exiting	CR3-store exiting	CR8-load exiting
	CR8-store exiting	Use TPR shadow	NMI-window exiting	MOV-DR exiting
	Unconditional I/O exiting	Use I/O bitmaps	Monitor trap flag	Use MSR bitmaps
	MONITOR exiting	PAUSE exiting	Activate secondary controls	
Secondary processor-based VM-execution controls	Virtualize APIC accesses	Enable EPT	Descriptor-table exiting	Enable RDTSCP
	Virtualize x2APIC mode	Enable VPID	WBINVD exiting	Unrestricted guest
	APIC-register virtualization	Virtual-interrupt delivery	PAUSE-loop exiting	
	RDRAND exiting	Enable INVPCID	Enable VM functions	VMCS shadowing
	Enable ENCLS exiting	RDSEED exiting	Enable PML	EPT-violation #VE
	Conceal VMX non-root operation from Intel PT	Enable XSAVES/XRSTORS		
	Mode-based execute control for EPT	Use TSC scaling		
Exception Bitmap	I/O-Bitmap Addresses	TSC-offset		
Guest/Host Masks for CR0	Guest/Host Masks for CR4	Read Shadows for CR0	Read Shadows for CR4	
CR3-target value 0	CR3-target value 1	CR3-target value 2	CR3-target value 3	CR3-target count
APIC Virtualization	APIC-access address	Virtual-APIC address	TPR threshold	
	EOI-exit bitmap 0	EOI-exit bitmap 1	EOI-exit bitmap 2	EOI-exit bitmap 3
	Posted-interrupt notification vector	Posted-interrupt descriptor address		
Read bitmap for low MSRs	Read bitmap for high MSRs	Write bitmap for low MSRs	Write bitmap for low MSRs	
Executive-VMCS Pointer	Extended-Page-Table Pointer	Virtual-Processor Identifier		
PLE_Gap	PLE_Window	VM-function controls	VMREAD bitmap	VMWRITE bitmap
ENCLS-exiting bitmap	PML address			
Virtualization-exception information address	EPTP index	XSS-exiting bitmap		

VM-EXIT CONTROL FIELDS

VM-Exit Controls	Save debug controls	Host address space size	Load IA32_PERF_GLOBAL_CTRL		
	Acknowledge interrupt on exit	Save IA32_PAT	Load IA32_PAT	Save IA32_EFER	Load IA32_EFER
	Save VMX preemption timer value	Clear IA32_BNDCFGS	Conceal VM exits from Intel PT		
VM-Exit Controls for MSRs	VM-exit MSR-store count	VM-exit MSR-store address			
	VM-exit MSR-load count	VM-exit MSR-load address			

VM-EXIT INFORMATION FIELDS

Basic VM-Exit Information	Exit reason	Exit qualification		
	Guest-linear address	Guest-physical address		
VM Exits Due to Vectored Events	VM-exit interruption information	VM-exit interruption error code		
VM Exits That Occur During Event Delivery	IDT-vectoring information	IDT-vectoring error code		
VM Exits Due to Instruction Execution	VM-exit instruction length	VM-exit instruction information		
	I/O RCX	I/O RSI	I/O RDI	I/O RIP
VM-instruction error field				

- Natural-Width fields.
- 16-bits fields.
- 32-bits fields.
- 64-bits fields.

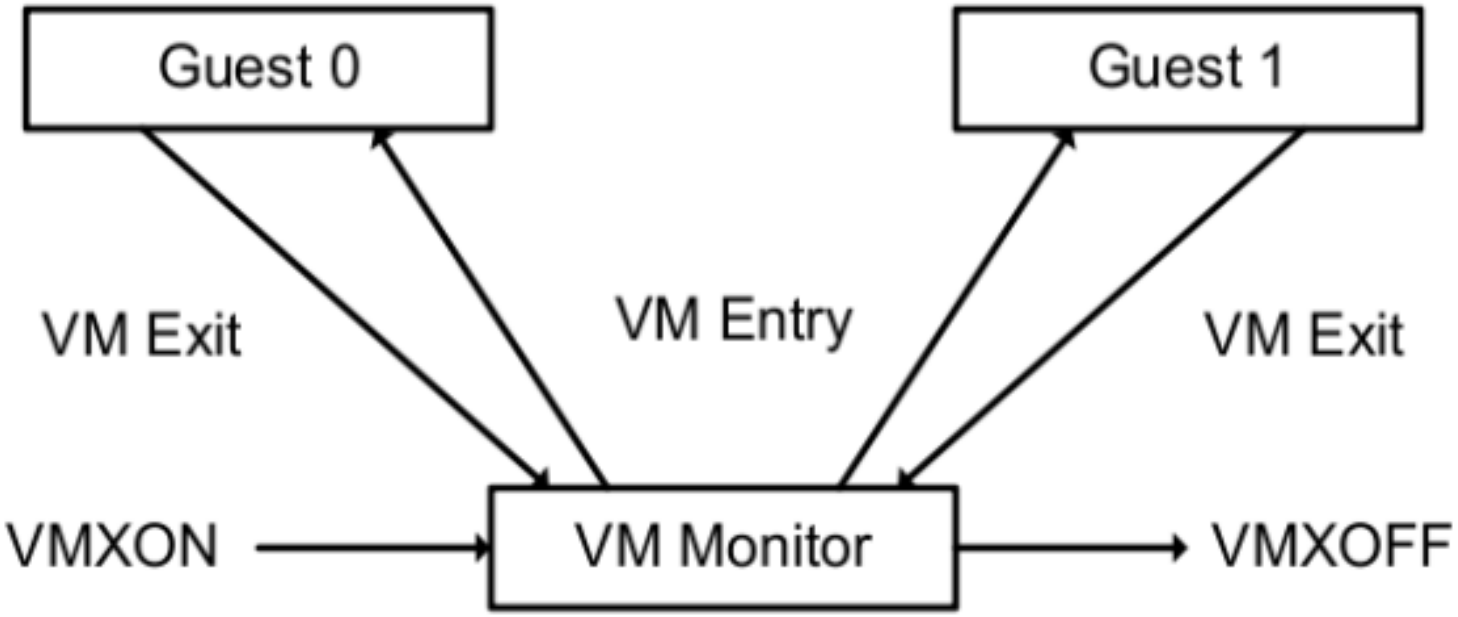
CopyLeft 2017, @Noteworthy (Intel Manuel of July 2017)

Хардуерно подпомогната виртуализация

- Нов режим на работа на CPU:
  - **VT Root операции** (за хипервайзора)
  - **Non root операции** (за гост ОС)
- Елиминира се компресията на ринга
- Нови инструкции на процесора – Virtual Machine Extensions (VMX)

VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXOFF, VMXON

VMX операции



Физически и виртуални процесори

- Процесор на ВМ – Virtual CPU (vCPU)
- Процесор на хоста – Physical CPU (pCPU)
- На vCPU се съпоставя pCPU
- Поддръжка на vCPU за ВМ – в зависимост от версията на хипервайзора

Hyper-V 2012 R2: 320 logical CPU / хост, до 2048 vCPU за ВМ /хост

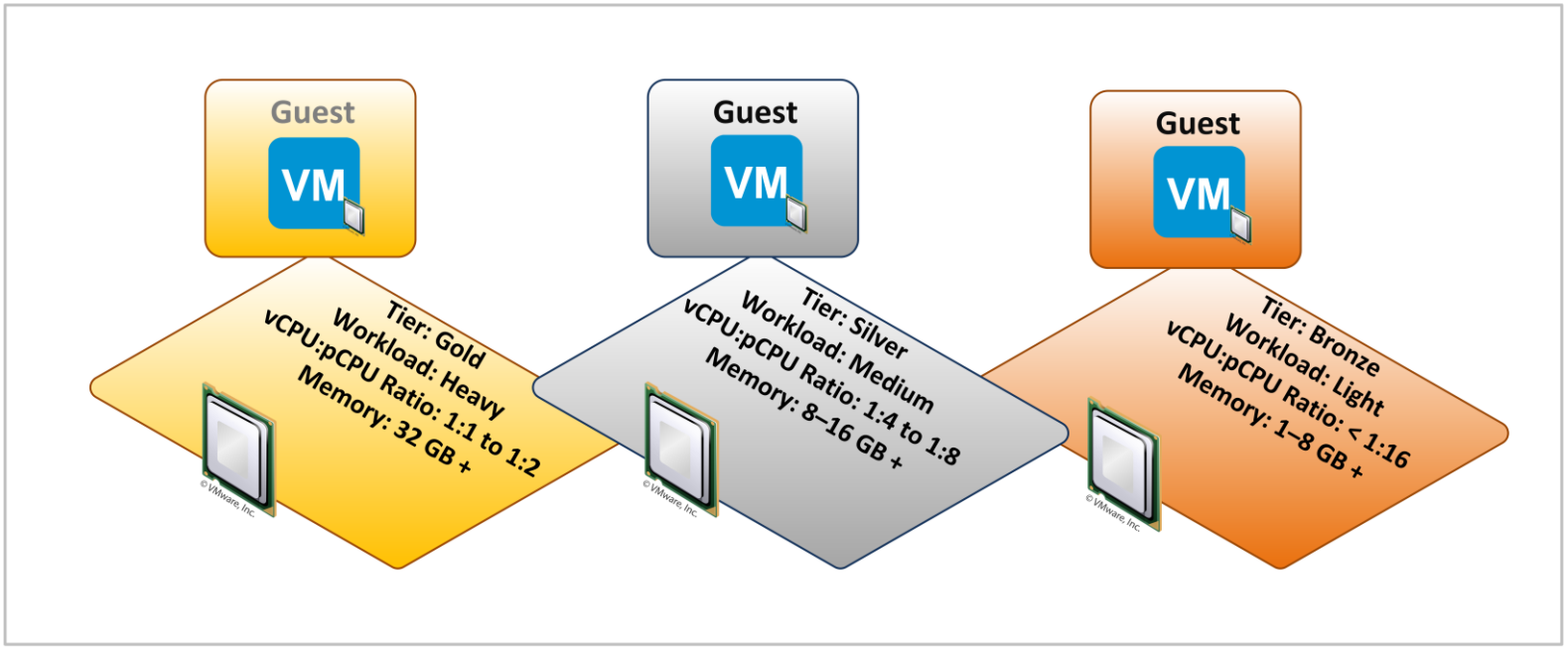
ESXi 6.0: 320 logical CPU / хост, до 4096 vCPU за ВМ / хост

Поддържан брой на CPU от ОС

- Windows 10 Home – 1 CPU, 256 cores
- Windows 10 Professional – 2 CPU, 256 cores
- Windows 10 Enterprise – до 4 CPU, 256 cores
- Windows Server 2016 – до 64 CPU, няма лимит

➢ *Използване на многоядрени процесори*

Отношение vCPU-pCPU



- Хипервайзорът планира всички vCPU на ВМ по едно и също време
- Ако не са налични -> *co-stop* състояние

Изчисляване на броя CPU

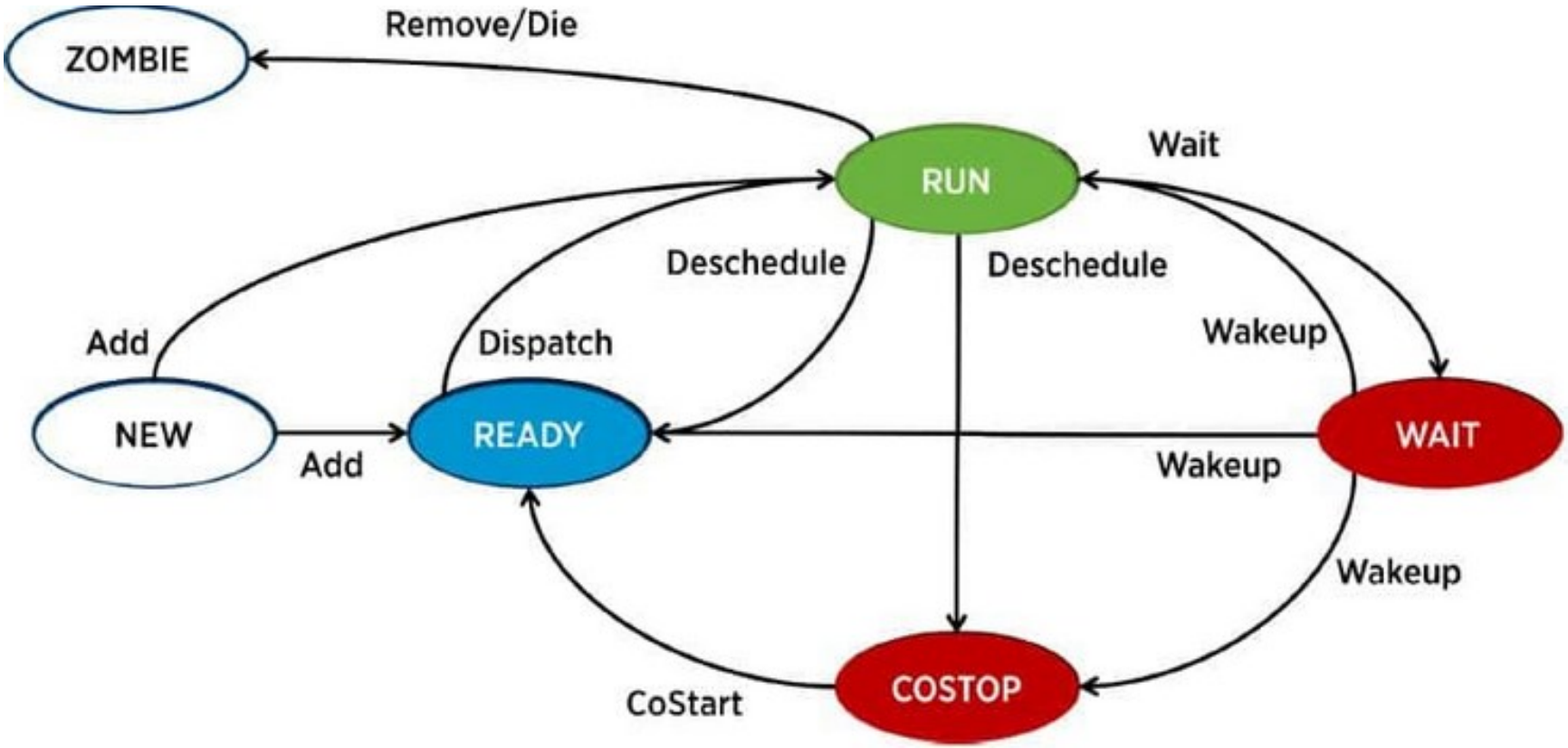
- Брой ядра на хост

(# Processor Sockets) X (# Cores/Processor) = # Physical Processors (pCPU)

- Брой логически ядра (при hyper-threading)

(# pCPU) X (2 threads/physical processor) = # Virtual Processors (vCPU)

Състояния на процесите



Въпроси?