

# Современная виртуализация

## Курс «Программное моделирование вычислительных систем»

Григорий Речистов  
[grigory.rechistov@phystech.edu](mailto:grigory.rechistov@phystech.edu)

18 мая 2015 г.

- 1** Классические условия виртуализации
  - Определения
  - Достаточное условие
  
- 2** Современные корректировки
  - Трансляция адресов
  - Периферийные устройства
  - Многопроцессорность

## На прошлых лекциях

- Симуляция — моделирование системы через имитацию внешне видимых эффектов, возникающих при взаимодействии с ней
- Эмуляция — моделирование системы через имитацию внутренней структуры и процессов, происходящих внутри её подсистем
- Виртуализация — обеспечение эффективной изоляции нескольких систем друг от друга при одновременном и прозрачном доступе к ресурсам нижележащей системы

# Связь виртуализации и симуляции



# История

- IBM VM — 1960 гг. [1]
- VMware Workstation — 1998 г.
- Intel VT-x, VT-i — 2005 г.

## Необходимые свойства

- 1** Изоляция — каждая виртуальная машина должна иметь доступ только к тем ресурсам, которые были ей назначены
- 2** Эквивалентность — любая программа, исполняемая под управлением ВМ, должна демонстрировать поведение, полностью идентичное реальной системе, за исключением эффектов, связанных с объёмами ресурсов
- 3** Эффективность — «статистически преобладающее подмножество инструкций виртуального процессора должно исполняться напрямую хозяйским процессором, без вмешательства монитора ВМ»



# Модель

- Один процессор, исполняющий инструкции
- Состояние:  $(M, P, R)$
- Два режима  $M$ :  $u$  и  $s$
- Указатель текущей инструкции  $P$
- Границы сегмента памяти  $R$  ( $l, b$ )
- Оперативная память: линейная  $E$  с ячейками  $E[n]$

## События ловушки (trap)

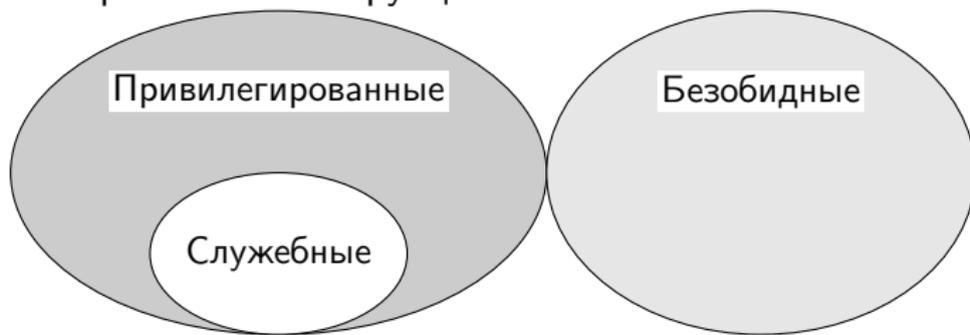
- Вызванные попыткой изменить состояние процессора (потока управления)
- Вызванные механизмом защиты памяти (ловушка з.п.)
- $E[0] \leftarrow (M1, P1, R1)$
- $(M2, P2, R2) \leftarrow E[1]$

# Инструкции

- Привилегированные (privileged). Исполнение с  $M=u$  всегда вызывает ловушку потока управления
- Служебные (sensitive)
  - Инструкции, исполнение которых закончилось без ловушки защиты памяти и вызвало изменение  $M$  и/или  $R$
  - Инструкции, поведение которых в случаях, когда они не вызывают ловушку защиты памяти, зависит или от режима  $M$ , или от значения  $R$
- Безвредные (innocuous) — не служебные

# Достаточное условие

Множество служебных инструкций является подмножеством привилегированных инструкций





# Построение

- Программы VM исполняют безобидные инструкции напрямую
- Служебные инструкции вызывают ловушку → переход в монитор, который их эмулирует
- Привилегированные инструкции (ОС внутри VM) → ловушка

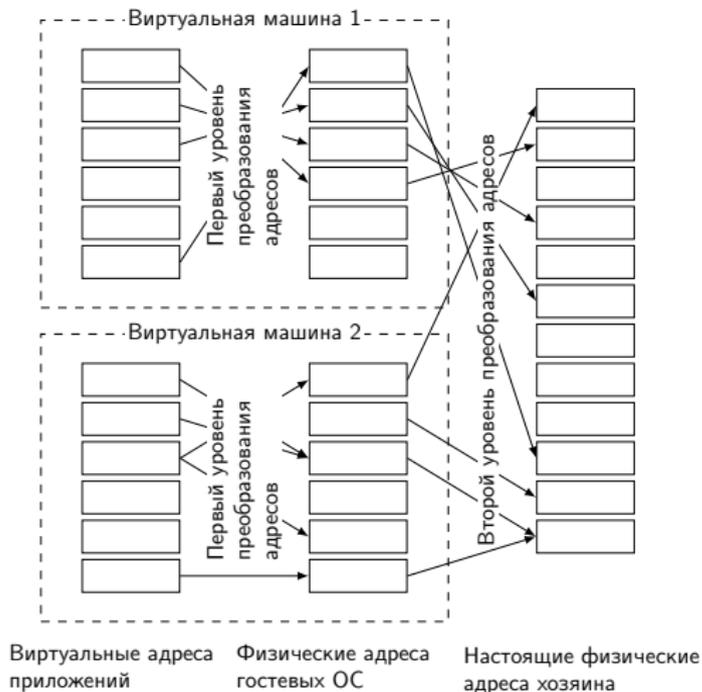


- 1 Изоляция
- 2 Эквивалентность
- 3 Эффективность

## Что не упомянуто в условии Г. и П.

- Сложные схемы трансляции адресов
- Периферия
- Многопроцессорные системы

# Трансляция адресов

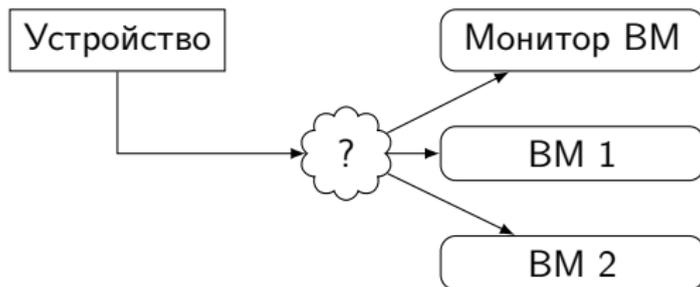


# Виртуализация TLB

Виртуальный адрес	Физический адрес	Тэг
0x11112222000	0x22220000	VM1
0x11112222000	0x11110000	VM2
0x44443333000	0x55554000	MON
0xabcd9876000	0x00001000	VM1
0xabcd9876000	0x11111000	VM3

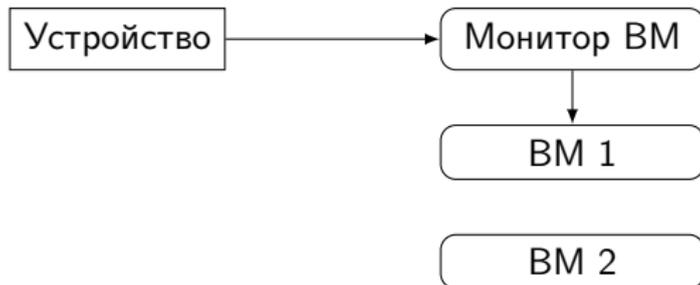
# Периферийные устройства

- Кому доставлять прерывание?
- Что делать, если прерывания внутри VM запрещены?



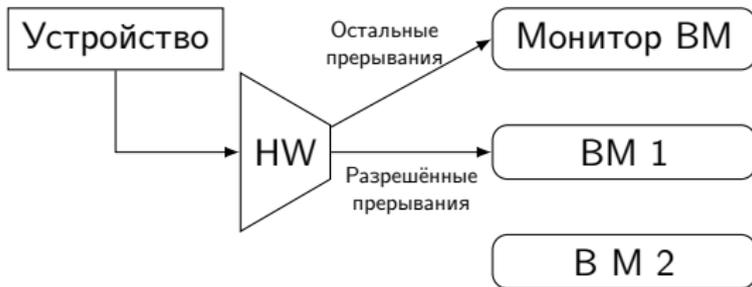
## Консервативный подход

- Все прерывания доставляются монитору
- Монитор «впрыскивает» их в VM
- Повышенная задержка доставки прерываний



# Аппаратная поддержка

Аппаратура обеспечивает доставку выбранных прерываний в текущую VM, остальные — в монитор



# Многопроцессорность

- Планировка исполнения  $N$  виртуальных процессоров на  $M$  физических,  $N \geq M$ 
  - Справедливая (fair)
  - Эффективная — характерные длительности синхронизационных процессов внутри VM должны быть близки к наблюдаемым на реальной аппаратуре
- Проблема вытеснения потоков, заблокировавших ресурсы (lock holder preemption)
- Монитору необходимо детектировать новый класс гостевых инструкций — синхронизационные примитивы (атомарные инструкции)

# Рекомендуемая литература I

-  Popek Gerald J., Goldberg Robert P. Formal requirements for virtualizable third generation architectures // Communications of the ACM. V. 17. #7. 1974.
-  Leung, F. [et al.] Intel® Virtualization Technology // ITJ Vol 10 Issue 3 — 2006.
-  Harlan McGhan. The gHost in the Machine: Parts 1,2,3 // Microprocessor Report. 2007. <http://mpronline.com>
-  Jiannan Ouyang and John R. Lange. Preemptable ticket spinlocks: improving consolidated performance in the cloud — 2013 <https://labs.vmware.com/vee2013/docs/p191.pdf>

## Рекомендуемая литература II



Аппаратная виртуализация. Теория, реальность и поддержка в архитектурах процессоров

<http://habrahabr.ru/company/intel/blog/196444/>



# На следующей лекции

# Спасибо за внимание!

Слайды и материалы курса доступны по адресу  
<http://is.gd/ivuboc>

*Замечание:* все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.