ФРТК

iSCALARE

intel

Лаборатория суперкомпьютерных технологий для биомедицины, фармакологии и малоразмерных структур

Виртуализация

Григорий Речистов
grigory.rechistov@phystech.edu

12.05.2014

На предыдущих лекциях:

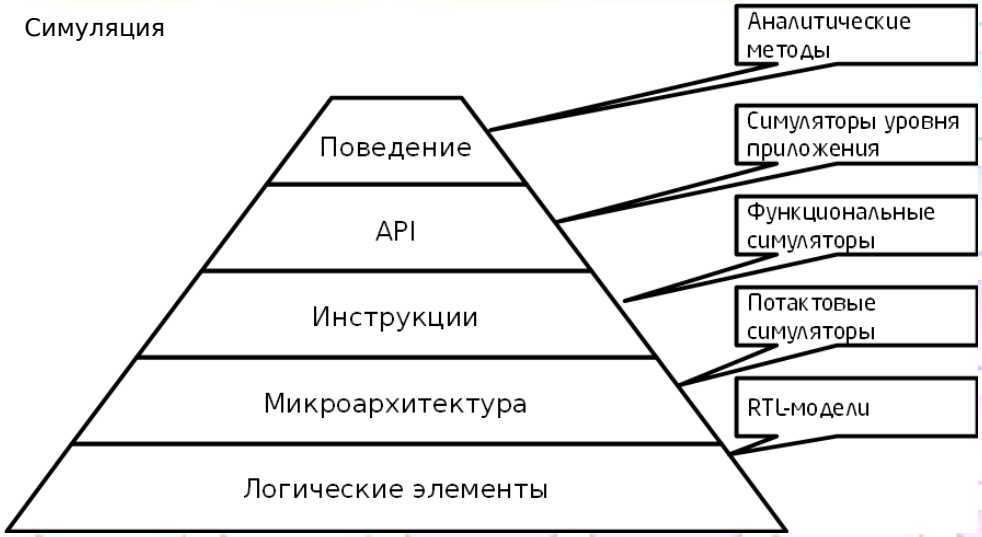
- Симуляция

На этой лекции:

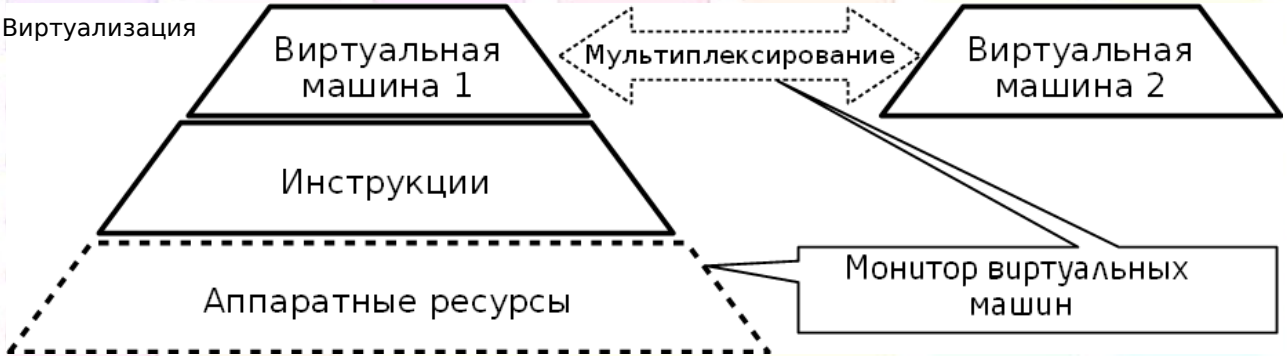
- Виртуализация как частный случай симуляции
- Условия эффективной виртуализации

Симуляция vs виртуализация

Симуляция



Виртуализация



Первое упоминание виртуализации

- IBM System/360 – 1960 гг.
- Popek Gerald J., Goldberg Robert P. **Formal requirements for virtualizable third generation architectures** // Communications of the ACM. V. 17. #7. 1974.

Признаки виртуализации

- **Изоляция** — каждая виртуальная машина должна иметь доступ только к тем ресурсам, которые были ей назначены.
- **Эквивалентность** — любая программа, исполняемая под управлением ВМ, должна демонстрировать поведение, полностью идентичное реальной системе, за исключением эффектов,
- **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)

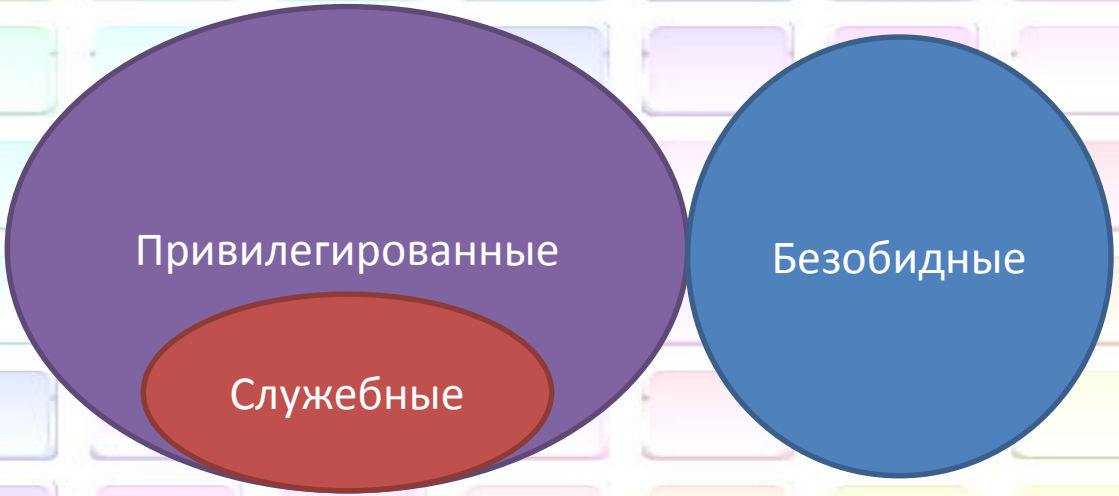
- Вызванные попыткой изменить состояние процессора (потока управления)
- Вызванные механизмом защиты памяти (ловушка з.п.)
- E[0] ← (M1,P1,R1)
- (M2,P2,R2) ← E[1]

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 — не служебные

Достаточное условие

Множество служебных инструкций является подмножеством привилегированных инструкций



Почему это так

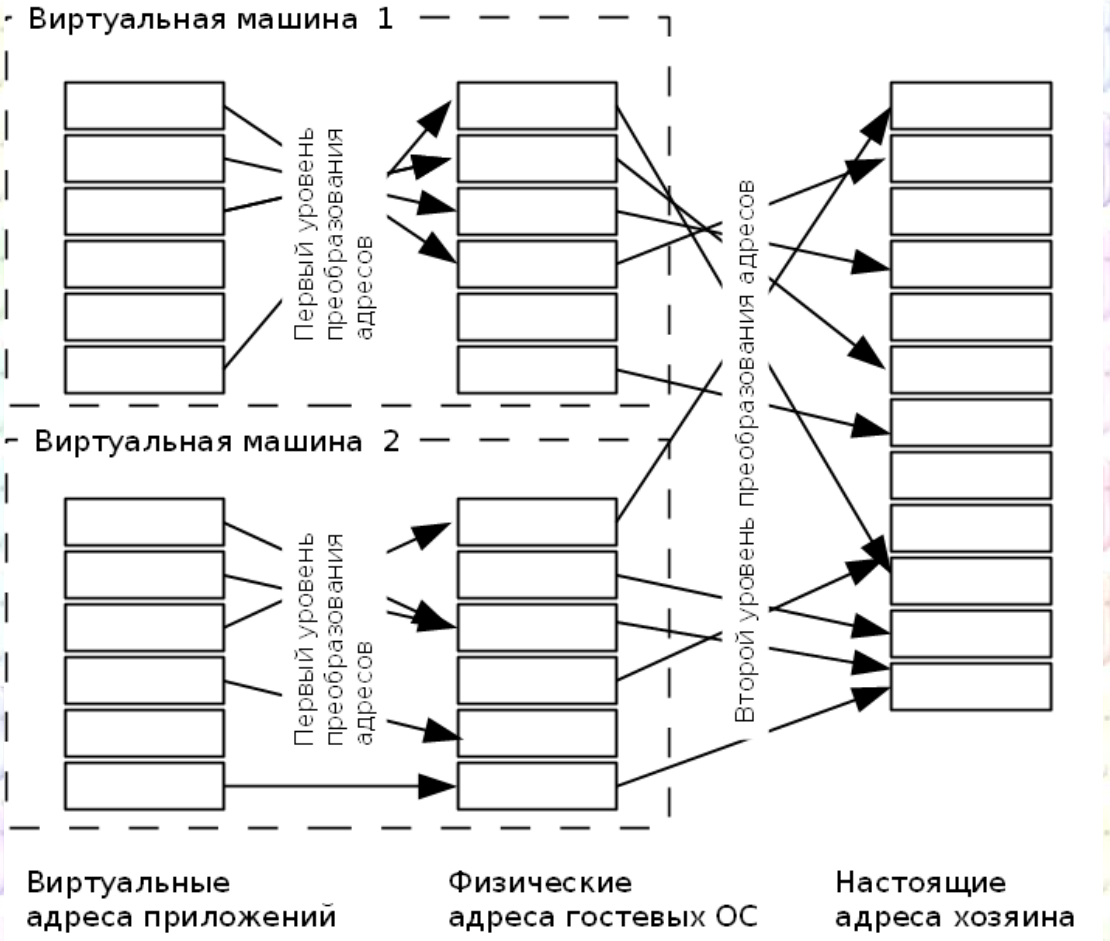
- Программы исполняют безобидные инструкции напрямую
- Служебные инструкции вызывают ловушку → переход в монитор, который их эмулирует
- Привилегированные инструкции (ОС в ВМ) → ловушка

- Изоляция
- Эквивалентность
- Эффективность

Что не упомянуто в условии Г. и П.

- Сложные схемы трансляции адресов
- Периферия
- Многопроцессорные системы

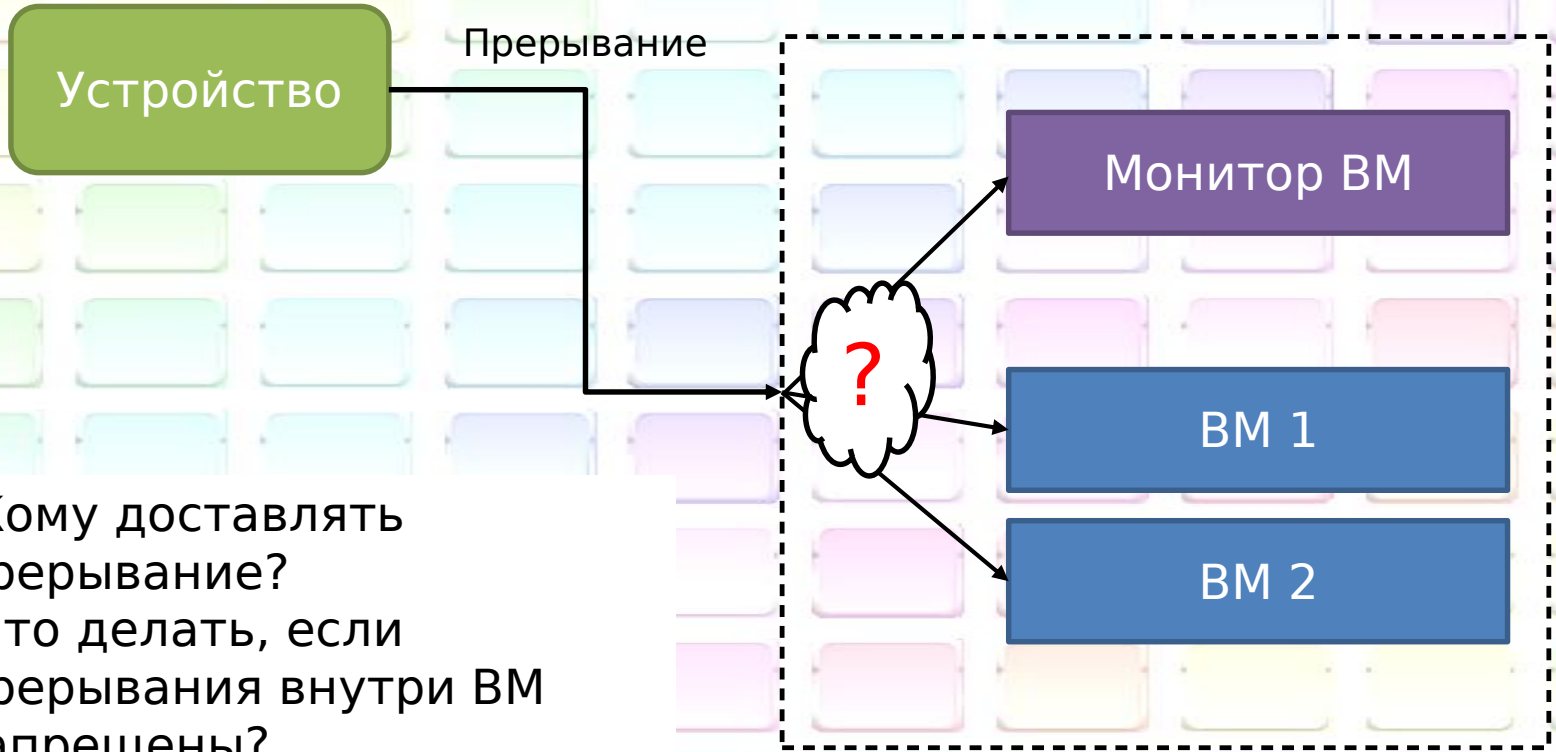
Трансляция адресов



TLB

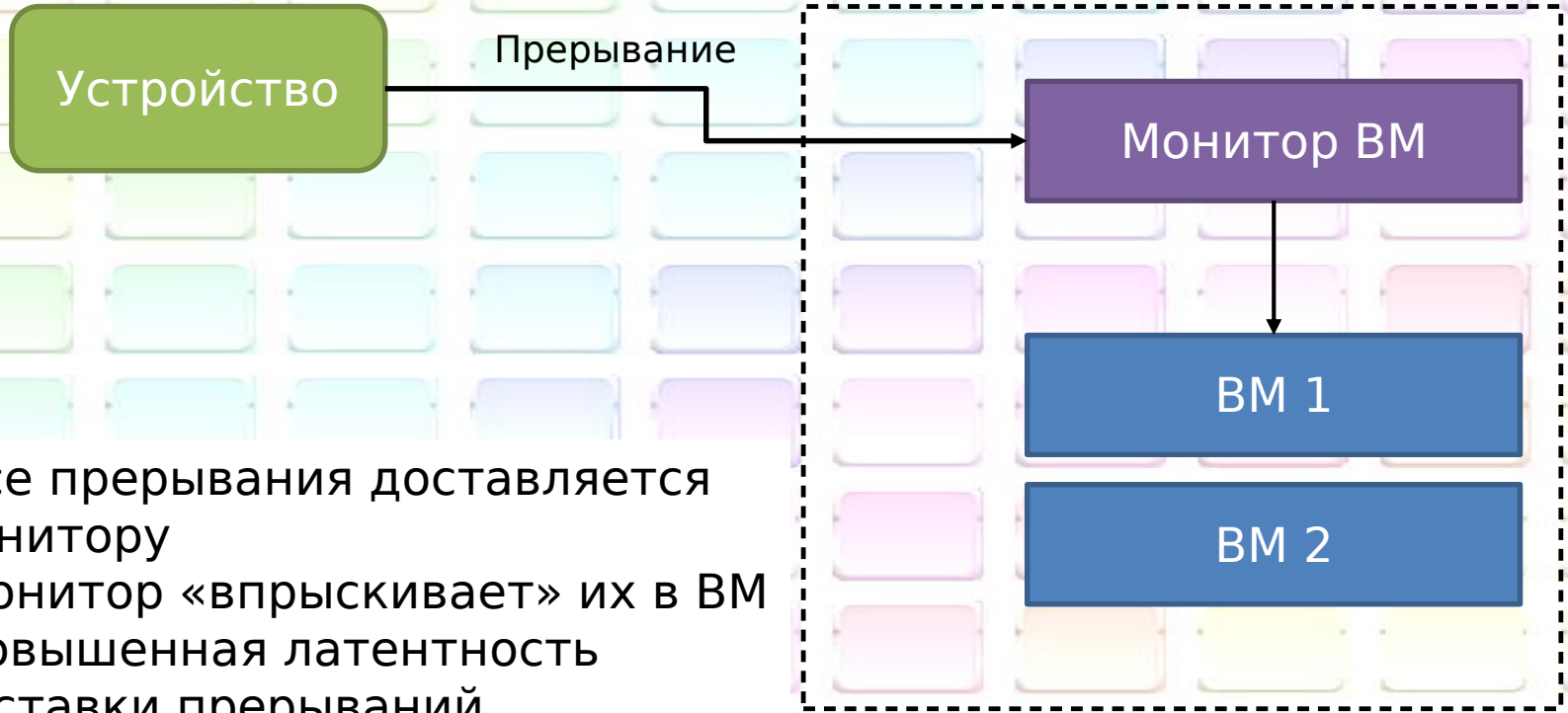
Виртуальный адрес	Физический адрес	Тэг
0x11112222	0x22220000	VM1
0x11112222	0x11110000	VM2
0x44443333	0x55554444	MON0
0xabcd9876	0x00001234	VM1
0xabcd9876	0x11111234	VM3

Периферийные устройства



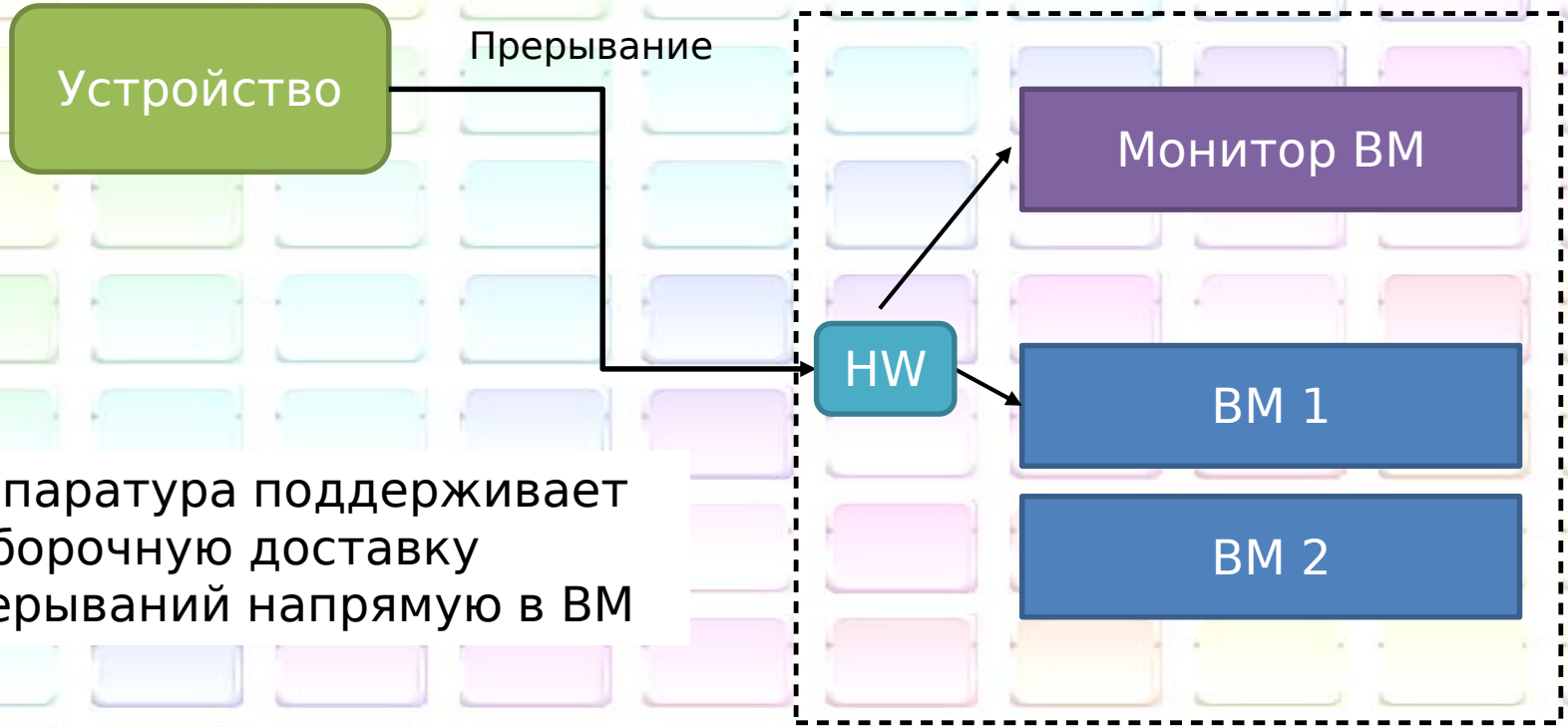
- Кому доставлять прерывание?
- Что делать, если прерывания внутри ВМ запрещены?

Периферийные устройства консервативный подход



- Все прерывания доставляется монитору
- Монитор «впрыскивает» их в ВМ
- Повышенная латентность доставки прерываний

Периферийные устройства аппаратная поддержка



•Аппаратура поддерживает выборочную доставку прерываний напрямую в ВМ

Многопроцессорность

- Планировка исполнения *N* виртуальных процессоров на *M* физических, $N \geq M$
  - Справедливая (fairness)
  - Эффективная — характерные длительности синхронизационных процессов внутри ВМ 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)

Литература

•Harlan McGhan. **The gHost in the Machine: Parts 1,2,3** // Microprocessor Report. 2007. http://mpronline.com

•Matias Zabaljauregui. **Hardware Assisted Virtualization Intel Virtualization Technology**. 2008

Спасибо за внимание!

Все материалы курса выкладываются на сайте лаборатории:
http://iscalare.mipt.ru/material/course_materials/