



iSCALARE

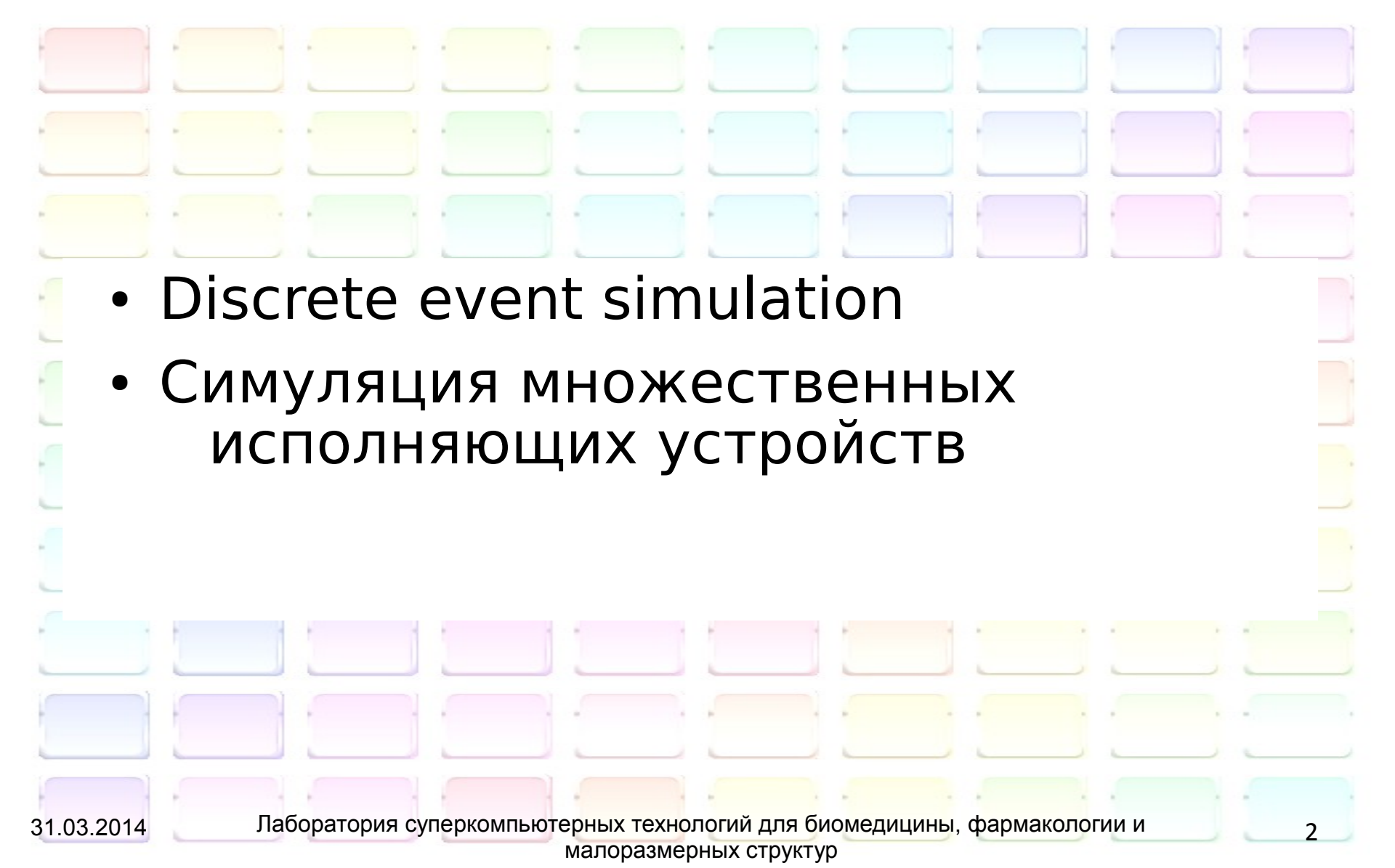


Лаборатория суперкомпьютерных технологий для биомедицины, фармакологии и малоразмерных структур

Полноплатформенная симуляция Многопроцессорные системы

Григорий Речистов

`grigory.rechistov@phystech.edu`

- 
- A decorative border composed of numerous small, colorful sticky notes in various shades of red, orange, yellow, green, blue, and purple, arranged in a grid-like pattern around the central text.
- Discrete event simulation
 - Симуляция множественных исполняющих устройств

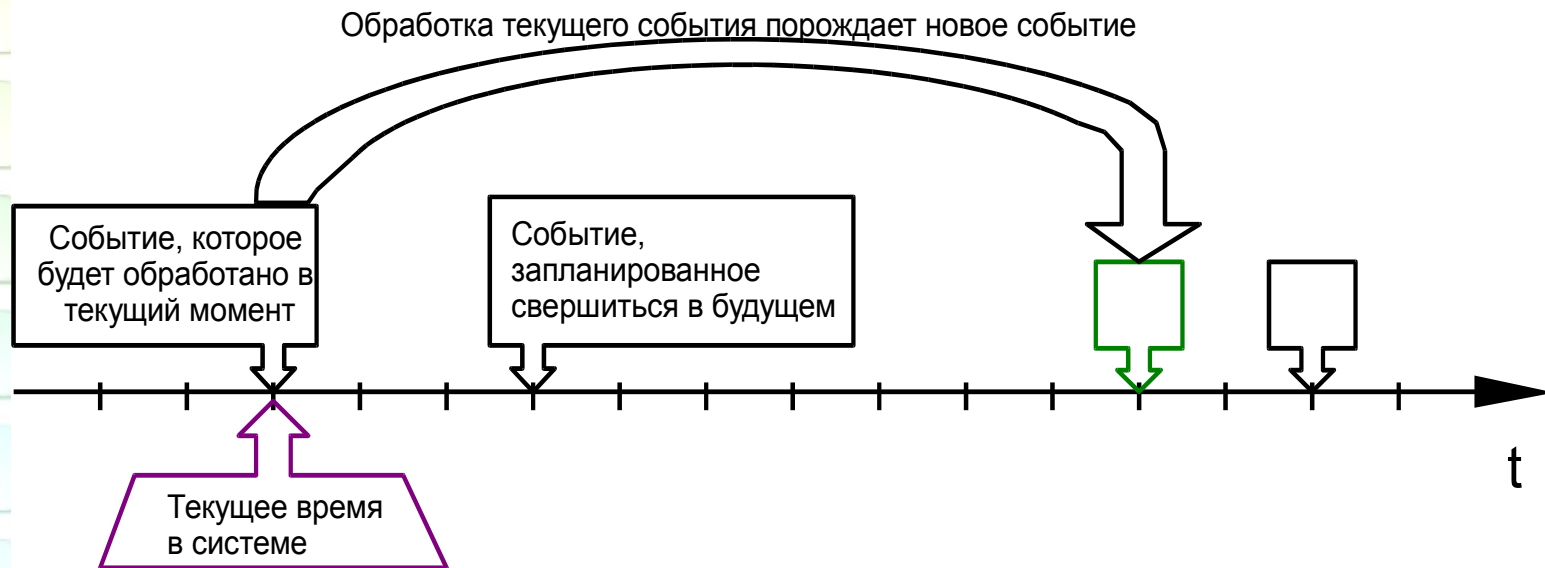
На предыдущей лекции:

- Рассмотрены различные способы моделирования ЦПУ
- Модель работала в «вакууме»
 - Единственное внешнее устройство — ОЗУ
-
- Как поступать, если мы хотим исполнять не только ЦПУ, но и периферийные устройства?

Дискретные события

- Состояние любого устройства в любой момент времени может быть описана полностью
- Изменения состояния происходят мгновенно в известные моменты времени
- Состояние полной системы = сумма состояний её компонентов

Очередь* событий



* Очередь с приоритетами

Что содержится в одном событии

- Функция, которая должна быть вызвана
- Объект, состояние которого изменяется
 - Может состоять из нескольких моделей
-
- Результаты обработки события
 - Изменение состояния системы
 - Добавление/уничтожение событий

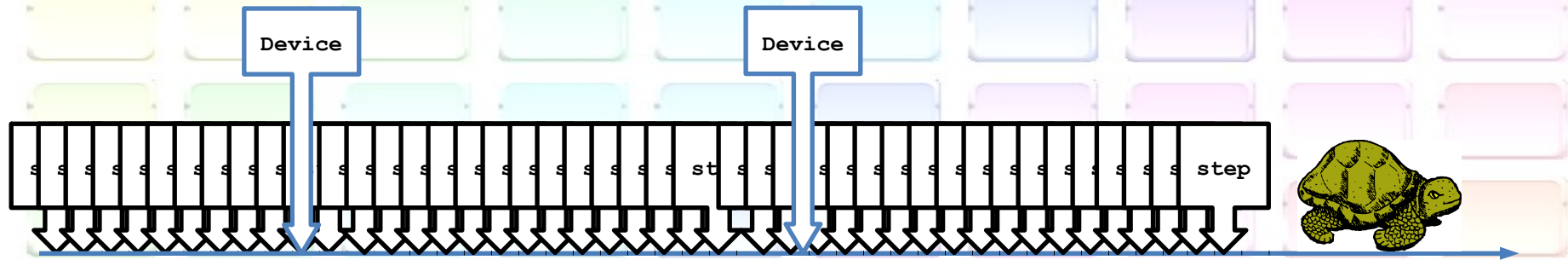
Алгоритм DES

```
struct evt_t { fn(); *obj; }  
uint sim_time = 0;  
while (! empty(queue)) {  
    sim_time += get_delta(queue);  
    evt_t evt = pop(queue);  
    evt.fn(evt.obj, queue);  
}
```

События

- Порождаемые события не могут попасть в прошлое, т.е. иметь метку времени меньше, чем текущее время.
- Обработка событий может не только порождать события в будущем, но и отменять некоторые из них (ещё не обработанные).
- Несколько событий могут иметь одинаковую метку времени.
- В модели может существовать больше одной очереди.
 - Первая очередь измеряется в инструкциях процессора, а другая – в тактах.
 - Многопроцессорные системы, в которых с каждым ЦПУ связана своя очередь.

Моделирование ЦПУ в DES

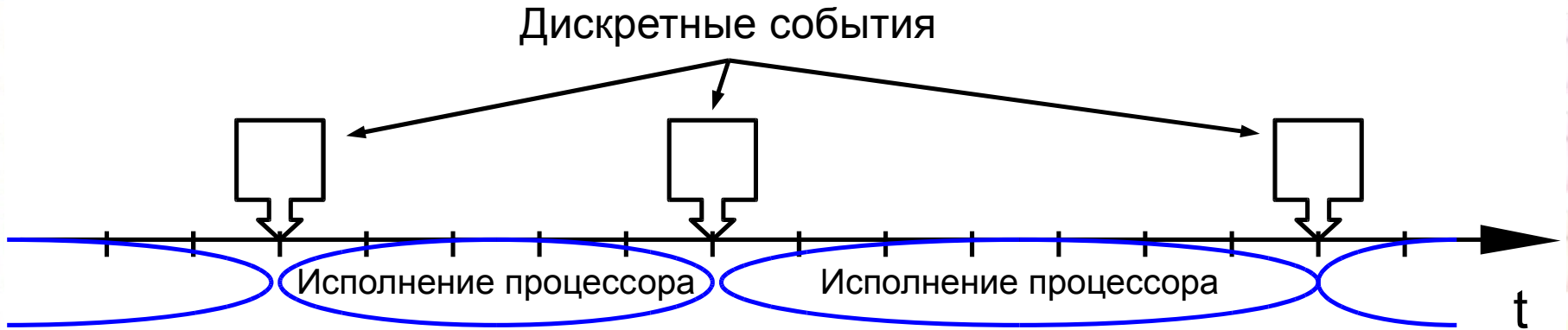


Лучшее, что можно будет использовать – это интерпретация

Два класса моделей

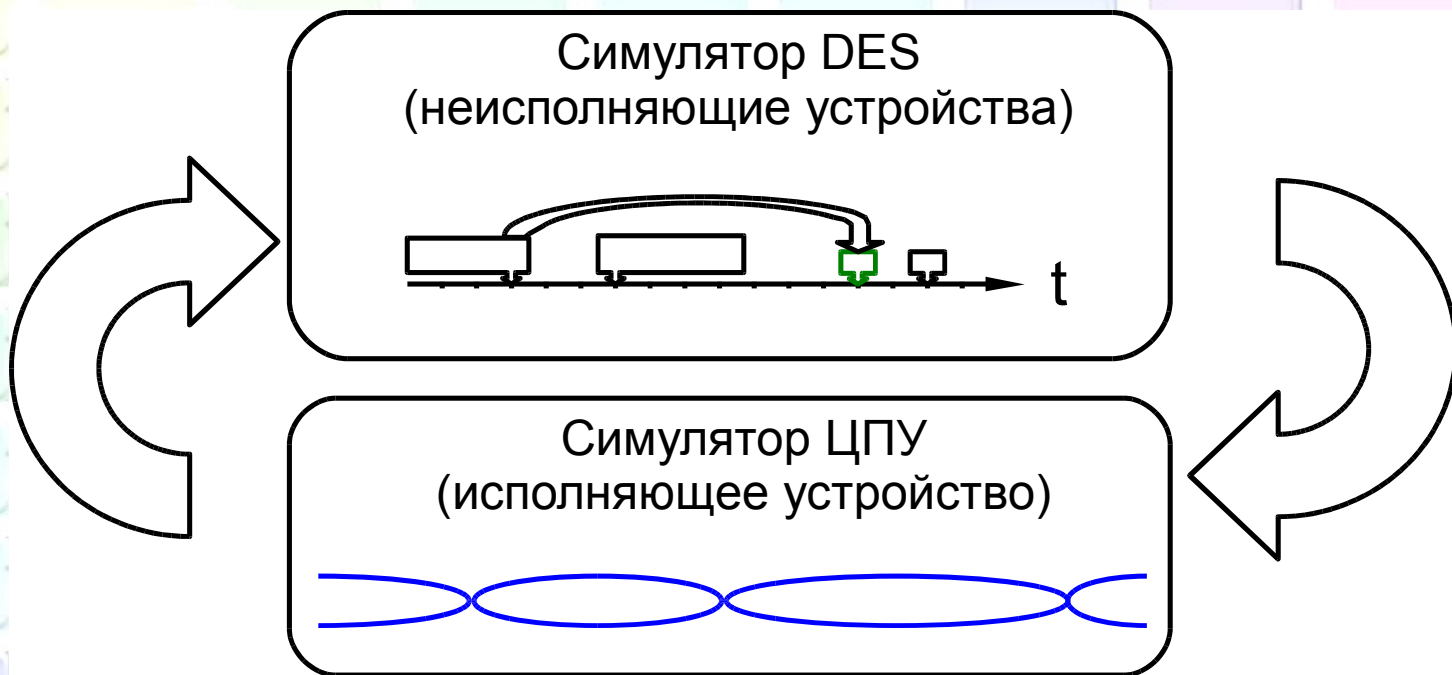
- Исполняющая (executing),
- Неисполняющая (non-executing)

Схема симуляции системы с ЦПУ



1. Определяется длительность интервала, в течение которого в моделируемой системе не произойдёт никаких событий. Эта величина равна времени ближайшего не обработанного события в очереди.
2. Управление передаётся в модель процессора, которая исполняется некоторое время, не превышающее найденное в первом пункте значение. Затем она останавливается и возвращает управление симулятору.
3. Симулируемое время продвигается на число тактов, потраченных процессором. События обрабатываются по модели DES. Затем мы переходим к первому шагу.

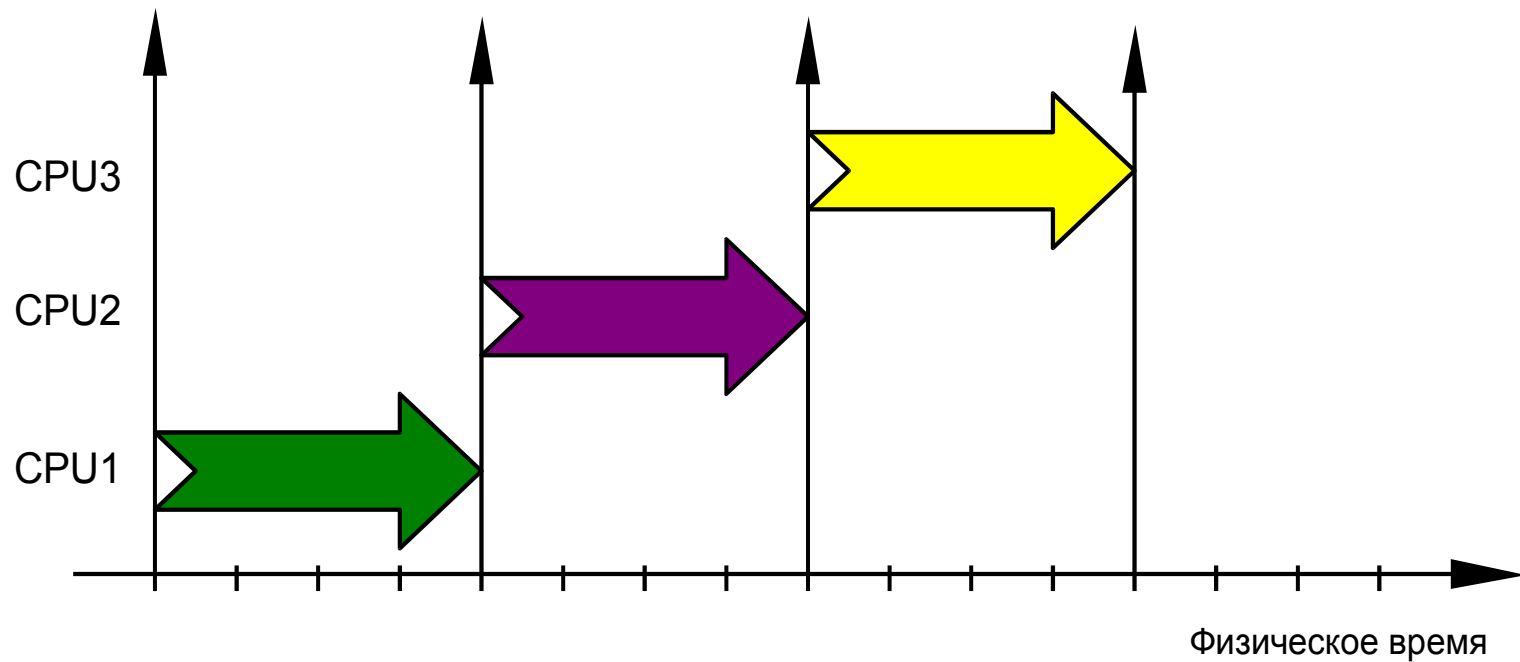
Ко-симуляция

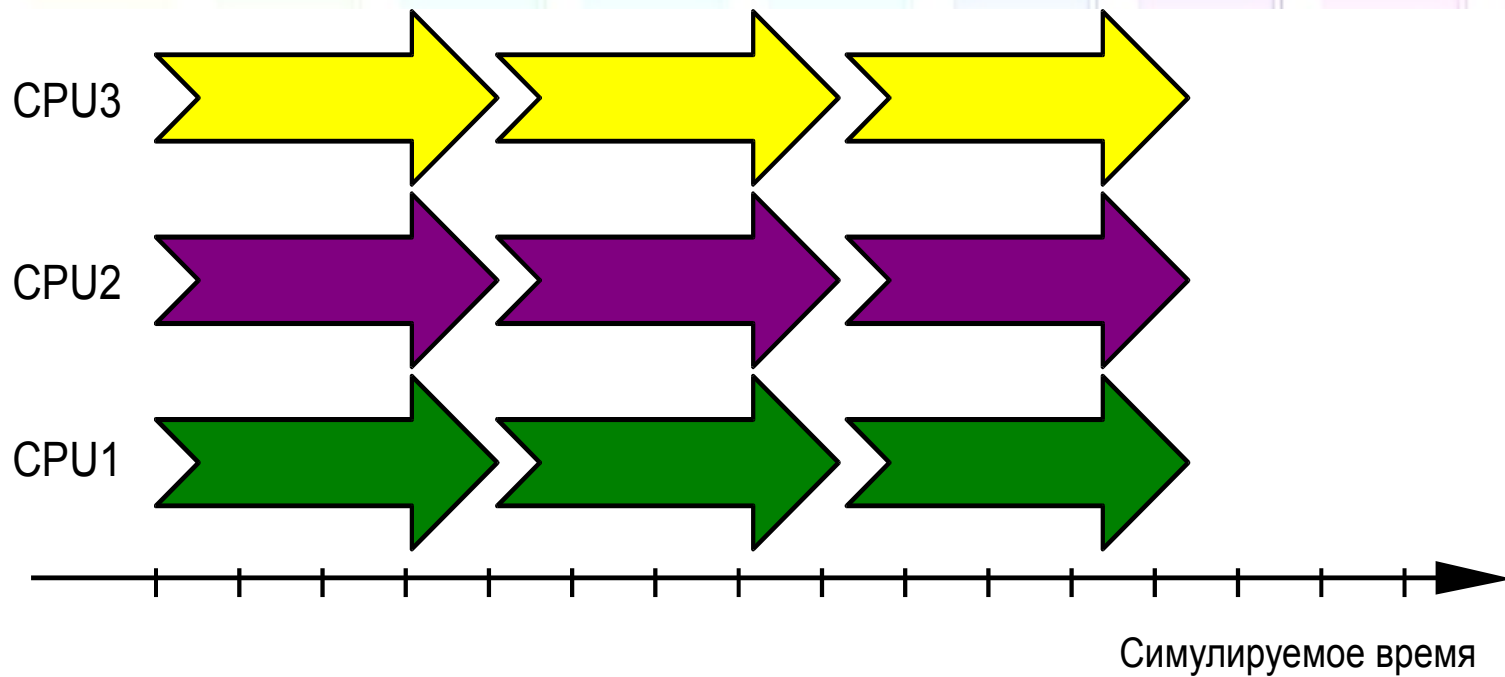


Оптимизация

- В реальности полная синхронность потоков приложению не гарантируется – оно должно быть написано таким образом, чтобы работать на разных системах
- Можно моделировать отдельные процессоры большими кусками симулируемого времени
- Максимальный размер такого интервала – *квота* исполнения
== Quota == quantum == switch time

Переключение текущего устройства





Замечания

- Процессор может исполнить меньше инструкций, чем содержится в выданной ему квоте.
- Не следует увлекаться излишне большими квотами, пытаясь ускорить исполнение – это может негативно повлиять на точность модели
 - Каждое устройство бежит в полной изоляции, остальные заморожены и не могут посылать ему никакие сигналы.
- В симуляторе могут быть реализованы псевдособытия, обработка которых вызывает переключение текущего устройства.

Пример: очереди сообщений

```
mc - /tmp_proj/grechist/iscalare/genx - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

simics> master0.mb.cpu0.core[0][0]->time_queue
[["sim", "Time Quantum End", "master0.mb.cpu0.core[0][0]", "time_quantum", 68631], ["master0.m
b.sb.uhci[0]", "frame_update", 0, "default", 2268631], ["master0.mb.sb.uhci[1]", "frame_update
", 0, "default", 2268631], ["master0.mb.sb.uhci[2]", "frame_update", 0, "default", 2268631], [
"master0.mb.sb.uhci[3]", "frame_update", 0, "default", 2268631], ["master0.mb.sb.uhci[4]", "fr
ame_update", 0, "default", 2268631], ["master0.mb.sb.uhci[5]", "frame_update", 0, "default", 2
268631], ["master0.mb.sb.timer", "handle_irq", 0, "default", 31377806], ["master0.mb.sb.lpc",
"pm1_ovf", 0, "default", 6057665917]]

simics> master0.mb.cpu0.core[0][0]->step_queue
[]

simics> help master0.mb.cpu0.core[0][0]->step_queue
Attribute <x86-nehalem>.step_queue

Optional attribute; read/write access; type: [[o|n,s,a,s,i]*].

((object, evclass, value, slot, step)*). Pending step queue events.

simics> █
```

Пример: квота ЦПУ

```
mc - /tmp_proj/grechist/iscalare/genx - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

simics> cpu-switch-time
Current time quantum: 9.52381e-05 s
 314285.7 master0.mb.cpu0.core[0][0]
 314285.7 master0.mb.cpu0.core[1][0]
 314285.7 master0.mb.cpu1.core[0][0]
 314285.7 master0.mb.cpu1.core[1][0]
Default time quantum: 9.52381e-05 s
simics> master0.mb.cpu0.core[1][0]->freq
master0.mb.cpu0.core[1][0]->freq_mhz master0.mb.cpu0.core[1][0]->frequency
simics> master0.mb.cpu0.core[1][0]->frequency
[3300000000, 1]
simics> master0.mb.cpu0.core[1][0]->freq_mhz
3300
simics> master0.mb.cpu0.core[1][0]->freq_mhz = 1000
simics> cpu-switch-time
Current time quantum: 9.52381e-05 s
 314285.7 master0.mb.cpu0.core[0][0]
 95238.1 master0.mb.cpu0.core[1][0]
 314285.7 master0.mb.cpu1.core[0][0]
 314285.7 master0.mb.cpu1.core[1][0]
Default time quantum: 9.52381e-05 s
simics> █
```

Рекомендуемая литература

Fujimoto R. M. **Parallel and Distributed Simulation Systems**
— New York, NY, USA : John Wiley & Sons, Inc., 2000. —
ISBN: 0471183830.

На следующей лекции:

Потактовая симуляция

Спасибо за внимание!

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

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.