



ОНЛАЙН-ОБРАЗОВАНИЕ

Знакомство

С Linux и не только

Александр Румянцев



Краткая история

Первое упоминание - 03.07.91

<https://groups.google.com/d/topic/comp.os.minix/T9SjMGTSpxk/discussion>

Первый релиз - 25.08.91 - считается днём рождения

<https://groups.google.com/d/topic/comp.os.minix/dINtH7RRrGA/discussion>

GNU - набор утилит (userland), Linux - ядро, вместе - GNU/Linux

POSIX (Portable Operating System Interface) - стандарт для совместимости Unix систем

Первый дистрибутив - Softlanding Linux System (1992) → Slackware (1993)

Значимые дистрибутивы

- Debian (1993) → DEB based distros**
- Red Hat (1994) → RPM based distros**
- Yellow Dog Linux (1999) → YUM (yellowdog update manager)**



Многозадачность

Что такое многозадачность?



Многозадачность

Что такое многозадачность?

Свойство операционной системы обеспечивать возможность параллельной или псевдопараллельной обработки нескольких задач

Первая многозадачная операционная система - Multics (Multiplexed Information and Computing Service), прародитель UNIX (UNified Information Computer System) - 1969 год



Кооперативная многозадачность

Переключение контекста инициирует приложение

+ отсутствие необходимости синхронизации

– ненадёжность

Вытесняющая многозадачность

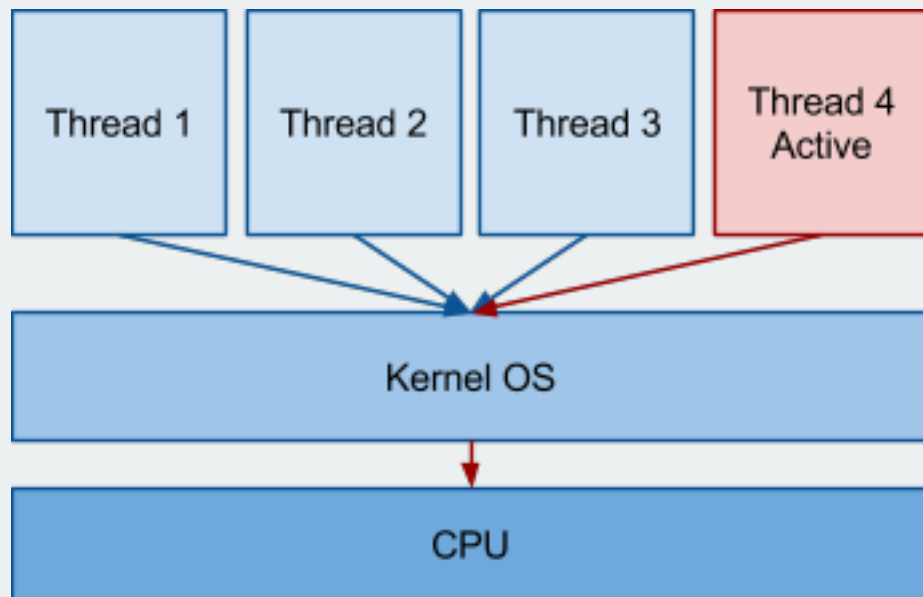
Переключение контекста инициирует ОС по прерыванию таймера

+ надёжность

– необходимость в неэффективных механизмах синхронизации

Linux - вытесняющая многозадачность





Процессор. Прерывания

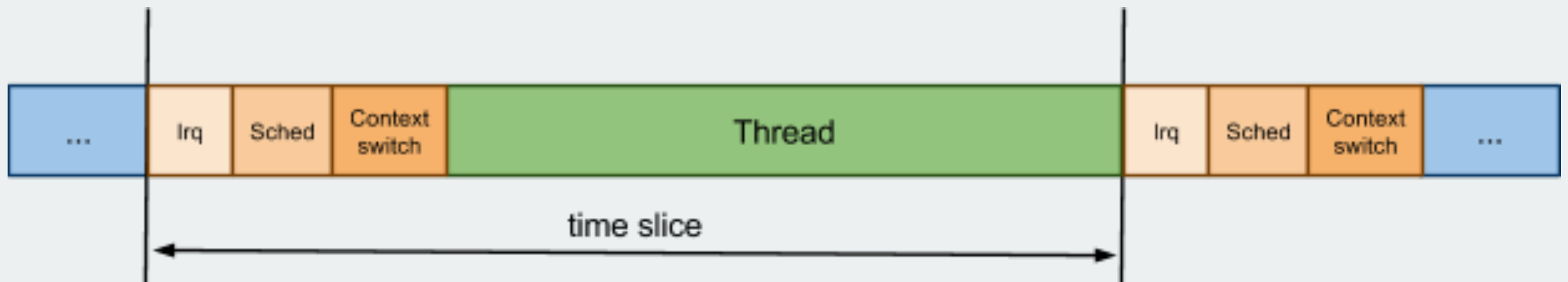
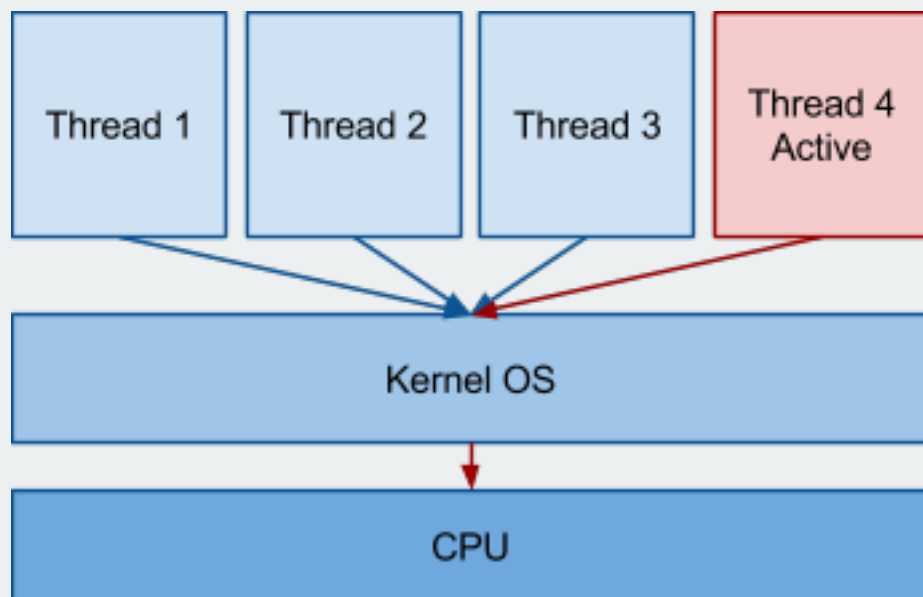
Прерывания - это функции, вызывающиеся по какому-то событию
Адреса этих функций хранятся IDT - Interrupt Description Table
Номер прерывания - номер в таблице IDT

Прерывания бывают:

- Аппаратные (если подано напряжение на ногу процессора)**
- Исключения (если произошла исключительная ситуация в коде)**
- Программные (вызываются в коде инструкцией INT)**

Процессор обрабатывает прерывание как и вызов функции, за исключением того, что в стек кладётся еще и регистр флагов





HZ

```
# grep 'CONFIG_HZ=' /boot/config-$(uname -r)  
CONFIG_HZ=1000
```

LA

Количество задач, стоящих в очереди на исполнение

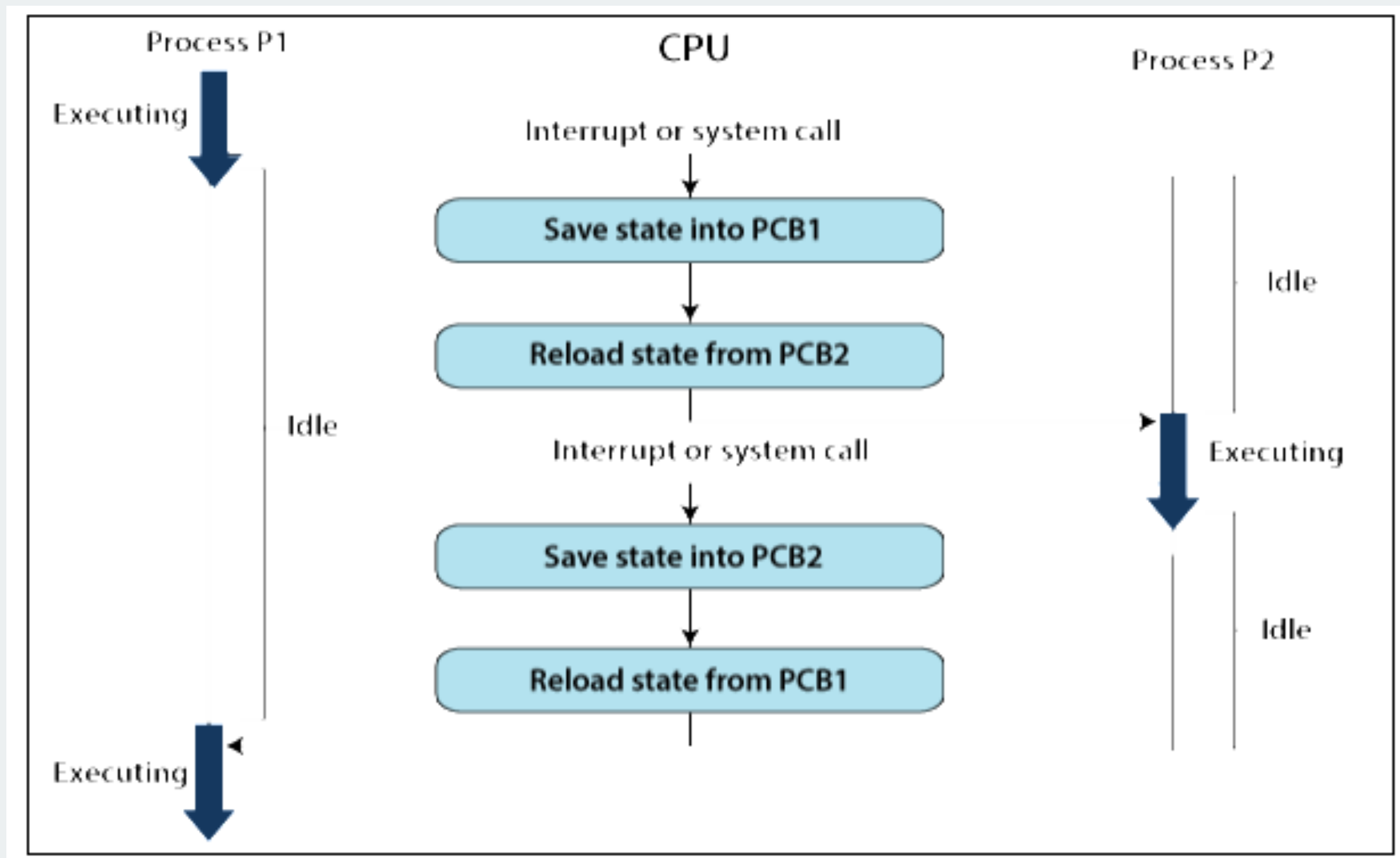


Контекст процессора (CPU context) — это структура данных, которая хранит внутреннее состояние регистров процессора.

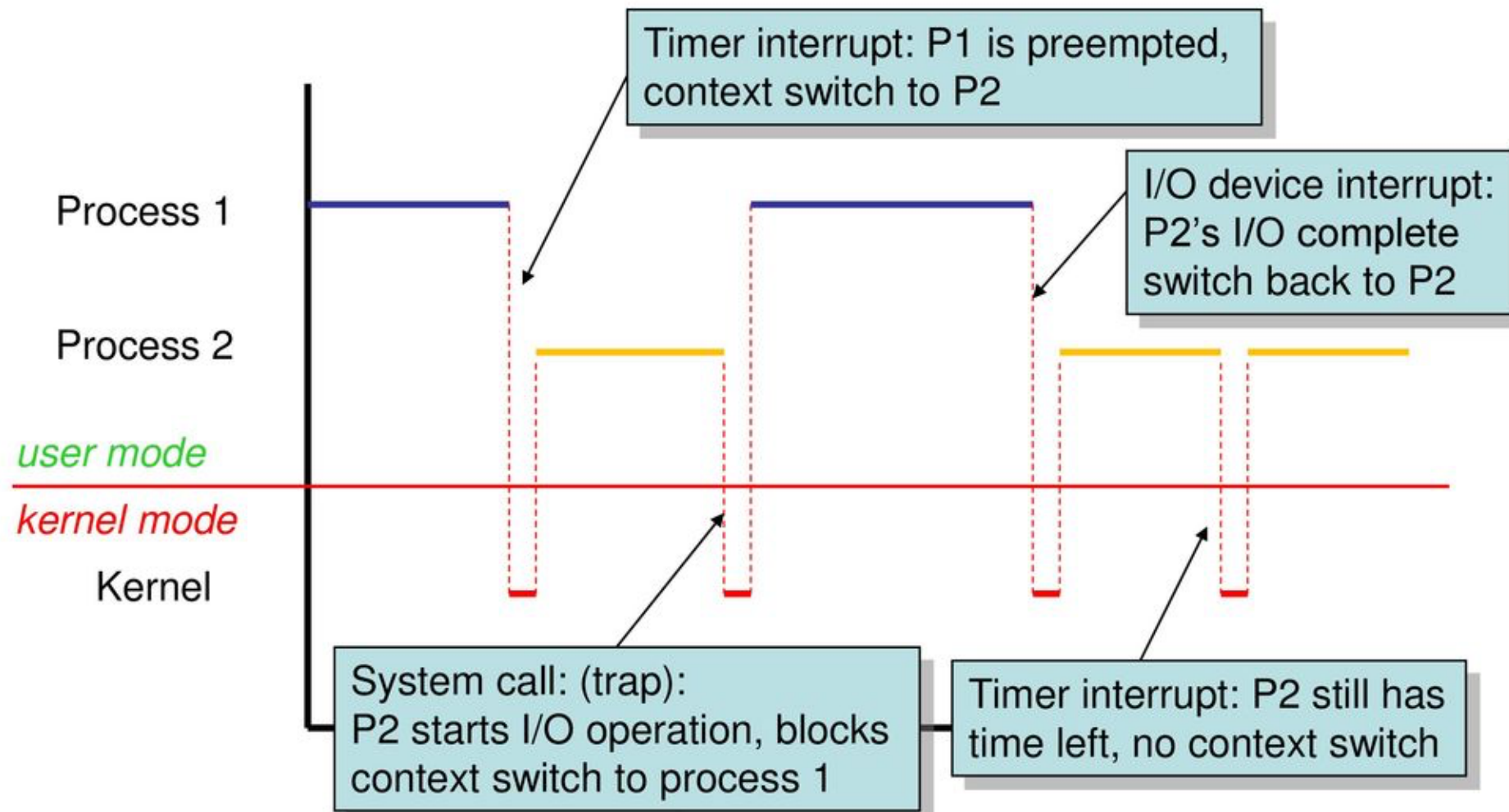
Контекст должен позволять привести процессор в корректное состояние для выполнения вычислительного потока.

Процесс замены одного вычислительного потока другим принято называть переключением контекста (context switch). Текущий контекст перед переключением сохраняется.





A Context Switch Scenario



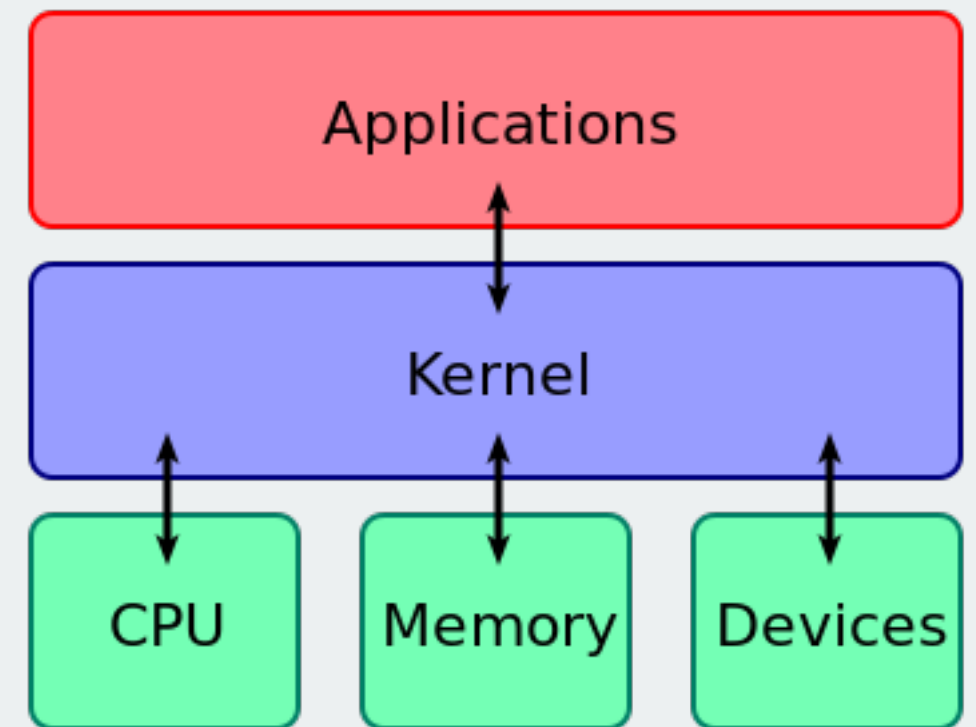
Ядро Linux

- Ядро ОС Linux монолитное, т.е. оно выполняется в одном адресном пространстве, в режиме ядра.
- Ядро Linux поддерживает динамическую загрузку модулей ядра.
- Ядро Linux является преемптивным. Ядро Linux в состоянии вытеснить выполняющееся задание, даже если это задание работает в режиме ядра.
- В ядре Linux используется интересный подход для поддержки многопоточности (threads): потоки ни чем не отличаются от обычных процессов.
Ядро Linux является полностью открытым

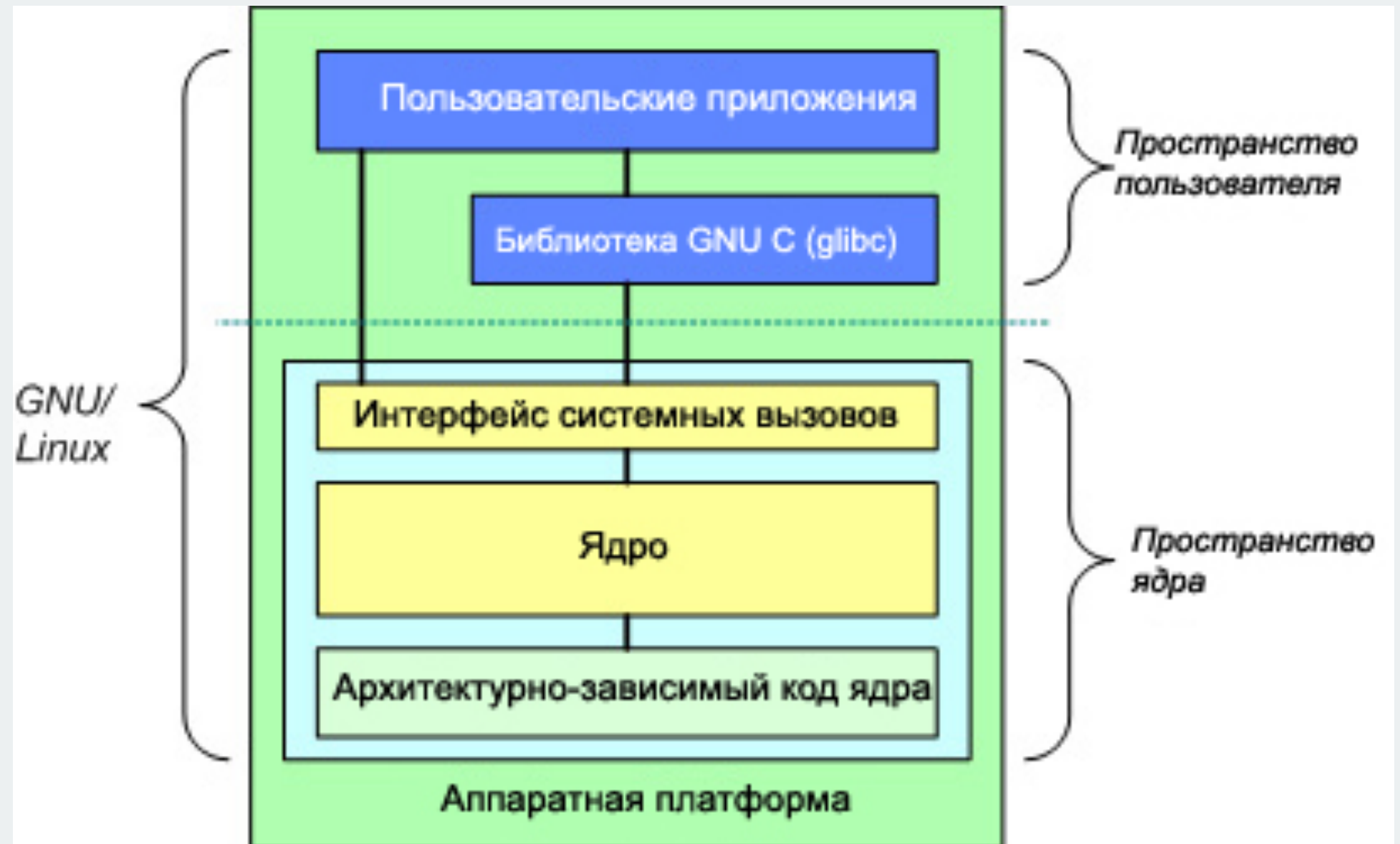


Функции ядра

- Распределение ресурсов (scheduling) между задачами
- Унифицированный доступ к железу (API)

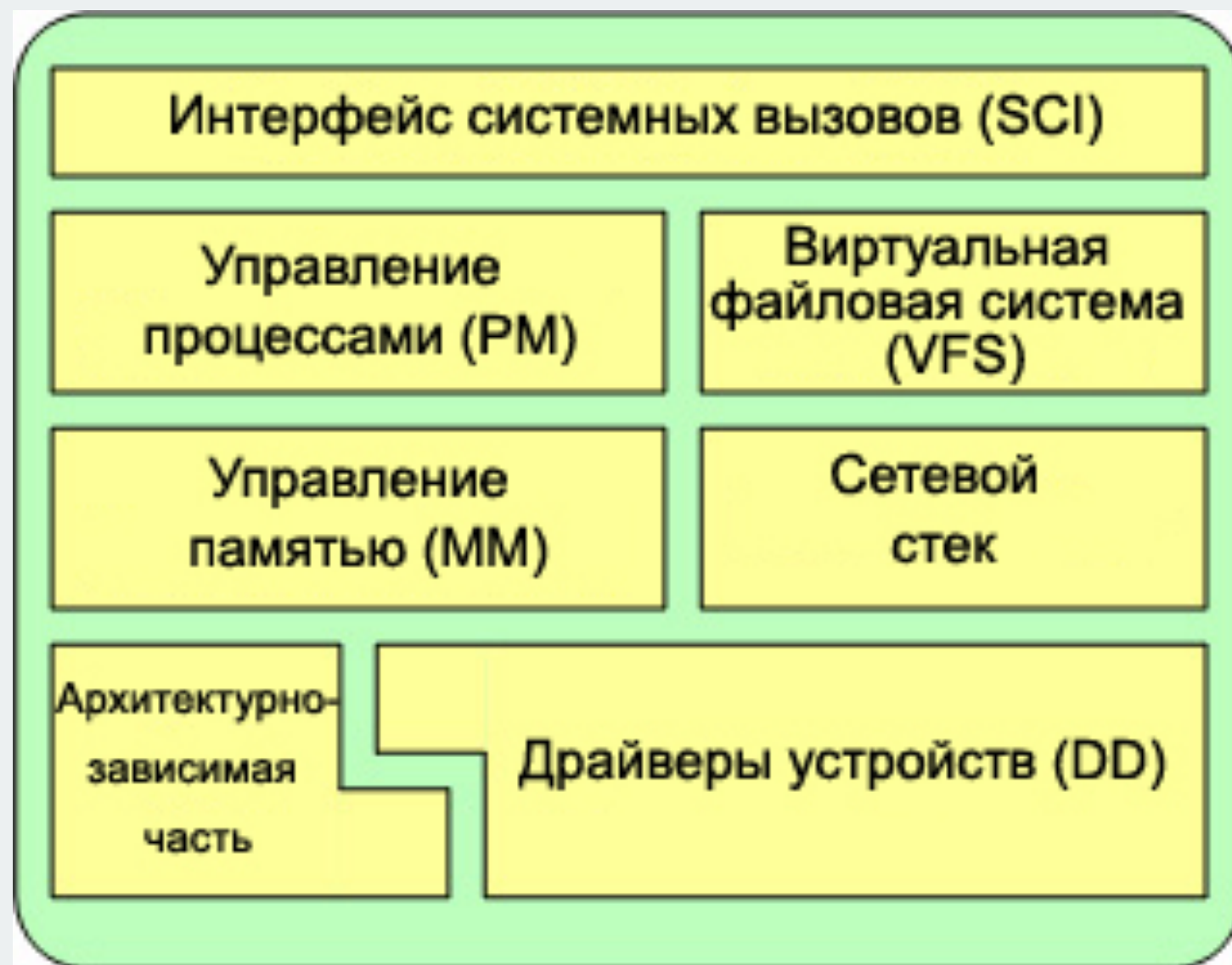


Функции ядра



интерактивная карта ядра Linux

[*https://makelinux.github.io/kernel/map/*](https://makelinux.github.io/kernel/map/)



Функции ядра (API) Linux - syscall

Библиотечные функции, исполняемые в контексте текущей задачи, вызываемые через программные прерывания (int 80) или специальную инструкцию процессора (syscall), выполняющиеся с повышенными привилегиями (ring 0)

При вызове syscall сохраняется состояние задачи (некоторые регистры и флаги) в стек, т.е. вызов syscall - дорогое удовольствие. Как пример - gettimeofday().

`printf("hello, world")` → `write(1, "hello, world\n", 13) = 13`

↑
GLIBC (C Standard Library)

←
syscall

←
STDOUT



Загрузка ядра и процессы ядра

В процессе выполнения кода ядра (загрузка) проходят следующие этапы:

- Инициализация механизмов управления ресурсами CPU (установка шедулера в обработчик прерывания таймера)**
- Инициализация системы управления памятью (установка обработчиков исключений, инициализация таблиц аллокаций)**
- Инициализация системы управления процессами (инициализация очереди сигналов)**
- Инициализация драйверов устройств (настройка устройств и установка обработчиков прерываний устройств)**
- Запуск ядерных процессов**
- Запуск первого процесса**



Сигналы

Сигнал - указание шедулеру, что процесс надо поставить на исполнение не с последней инструкции, на которой он был прерван, а с сохраненной в таблице обработчиков сигналов функции

**Обработчики сигналов устанавливаются программой.
Сигнал с номером 9 не передаётся процессам, шедулер просто инициирует завершение процесса.
Сигнал номер 9 игнорируется для процесса с номером 1**

Сигналы инициируются ядром (в случае исключительных ситуаций) или другими процессами



Трассировка

Утилиты для трассировки

`strace` - трассировка системных вызовов

`ltrace` - трассировка библиотечных вызовов

Посмотрим, что это на примере "hello, world"



Linux версии

Система нумерации до 3.0

2.6.32

минор четный - стабильный релиз
минор нечетный - ветка разработки

Система нумерации после 3.0

Как попало

mainline
stable
longterm (LTS)
linux-next

У разных дистрибутивов возможно разное понимание LTS

<https://www.kernel.org/>



Ядра для RHEL/CentOS

Штатные

```
yum update kernel  
yum remove kernel
```

ELRepo - "ванильные ядра" (<http://elrepo.org/tiki/tiki-index.php>)

```
rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm
```

kernel-lt - longterm

kernel-ml - mainline

OUEK - Oracle Unbreakable Enterprise Kernel

```
curl -o /etc/yum.repos.d/ouek.repo http://yum.oracle.com/public-yum-ol7.repo
```



Варианты конфигурации ядра

- модификация настраиваемых параметров ядра;
- сборка ядра из исходных кодов с внесением нужных изменений и/или дополнений в тексты исходных кодов ядра;
- динамическое подключение новых компонентов (функциональных модулей, драйверов) к существующей сборке ядра;
- передача специальных инструкций ядру во время начальной загрузки и/или используя загрузчик (например GRUB).



Варианты конфигурации ядра

- `/etc/sysctl.conf`
- `/proc/sys/fs`
`/proc/sys/net/ipv4`
`/proc/sys/kernel`
`/proc/sys/dev/cdrom`



Модули и параметры ядра

Унификация образов ОС, модули подгружаются по мере необходимости
Уменьшение размера образа ядра

Утилиты для работы с модулями:

`lsmod`
`modprobe`
`modinfo`
`insmod`
`rmmod`

`ls /lib/modules/`
`find /lib/modules -name *.ko`
`cat /proc/modules`

Конфигурация модулей:

`/sys/modules`
`/etc/modprobe.d`
`/etc/modules-load.d`



Своё ядро

```
cp /boot/config* .config &&  
make oldconfig &&  
make &&  
make install &&  
make modules_install
```



Домашка

1. Результаты складываем в github
2. Ссылку присылаем в "чат с преподавателем"





**Спасибо
за внимание!**

Вопросы?