



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/dlNtH7RRrGA/discussion>**

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

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

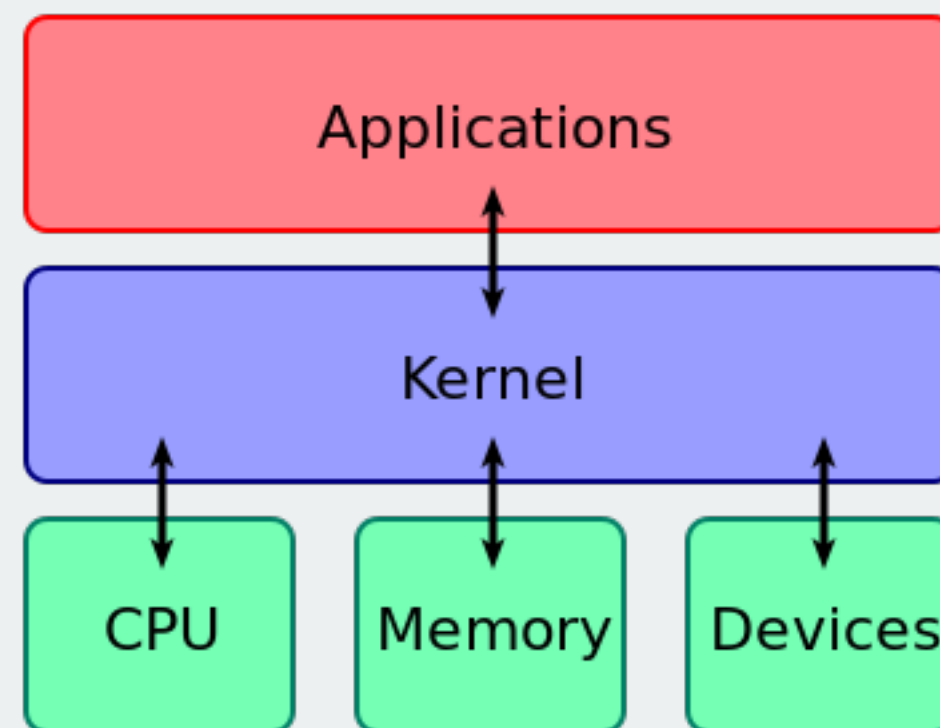
**Значимые дистрибутивы**

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



# Функции ядра

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



# 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**



## Виды многозадачности

1. Кооперативная

2. Вытесняющая

## Подвиды многозадачности

- Системы реального времени
- Пользовательская кооперативная

# Linux - вытесняющая многозадачность



## **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**





# HZ

```
# zgrep CONFIG_HZ= /proc/config.gz  
CONFIG_HZ=100
```

**Мало - потому что дорого**

**Переключение приложения происходит по прерыванию таймера**

# LA

**Количество задач, стоящих в очереди на исполнение**



# Виды ядер

**Монолитное (Linux)**

**Микроядро (GNU Mach, QNX)**

**Гибрид (Windows)**

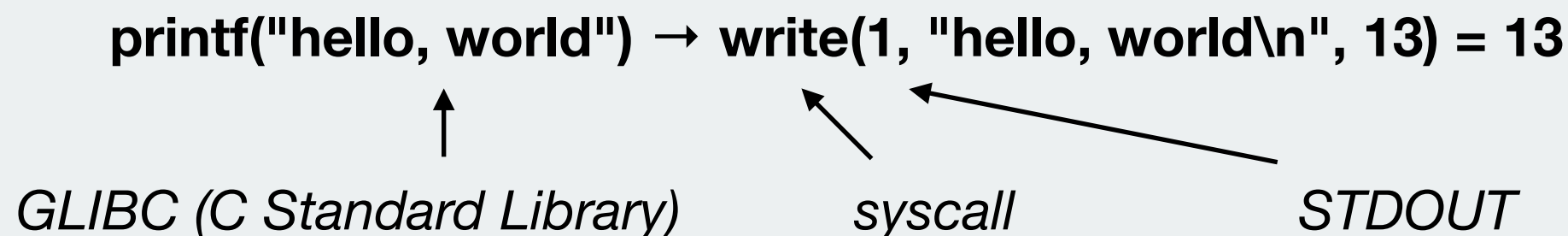
**Почти монолит (Mac, Xen)**



# 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().



# 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
```



# Модули и параметры ядра

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

**Утилиты для работы с модулями:**

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

**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"
3. При необходимости вы можете запросить виртуалку





**Спасибо  
за внимание!**

**Вопросы?**