СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ



к.т.н.
Папулин Сергей Юрьевич

papulin_bmstu@mail.ru

Программа курса





Модули

- Модуль 1
 - Основы теории операционных систем
 - Язык программирования Python
- Модуль 2
 - Распределенная файловая система
 - Контейнеризация вычислительных программ



Лекции. Модуль 1

- Основы теории операционных систем (Unix подобные Linux)
 - Архитектура ядра Linux
 - Управление памятью
 - Управление процессами
 - Межпроцессное взаимодействие
 - Файловая система
 - Сетевой стек и пр.
- Язык программирования Python
 - Архитектура
 - Интерпретатор
 - Глобальная блокировка интерпретатора (GIL)
 - Основные структуры данных и реализация принципов ООП и пр.



Семинары. Модуль 1

- Основы теории операционных систем (Unix подобные Linux)
 - Командная оболочка
 - Пользователи и группы
 - Права доступа
 - Взаимодействие с файловой системой и управление файлами
 - Запуск bash-скриптов
 - Управление демонами и пр.
- Язык программирования Python
 - Среда разработки и особенности кода на Python
 - Взаимодействие с файловой системой
 - Библиотеки для работы с процессами и потоками
 - Межпроцессное взаимодействие
 - Создание python-скриптов и пр.



Лекции. Модуль 2

- Распределенная файловая система
 - Распределенная файловая система HDFS
 - Архитектура и основные компоненты
 - Чтение и запись данных
 - Обеспечение отказоустойчивости и пр.

Контейнеризация вычислительных программ

- Контейнеризация
- Архитектура и объекты Docker
- Сетевые режимы
- Хранение данных
- Кластерный режим (swarm) и пр.



Семинары. Модуль 2

- Распределенная файловая система
 - Установка и настройка HDFS
 - Администрирование HDFS кластера.
 - Основные команды файловой системы
 - Java API и Python оболочка пр.

Контейнеризация вычислительных программ

- Основные команды по управлению контейнерами
- Создание образов с использованием Dockerfile
- Построение образов и развертывание контейнеров с docker-compose
- Развертывание HDFS в Docker среде и пр.



Модуль 1

Д31:

- Часть 1 (10 баллов)
- Часть 2 (10 баллов)

$$PK1 = 35/20 \cdot (K1.1 \cdot Д31. Часть 1 + K1.2 \cdot Д31. Часть 2)$$

$$PK1 = K1.1 \cdot Д31.$$
 Часть $1 + K1.2 \cdot Д31.$ Часть $2 + Вопросы (15 баллов)$

K – коэффициент (1; 0.85; 0.7)

Модуль 2

Д32:

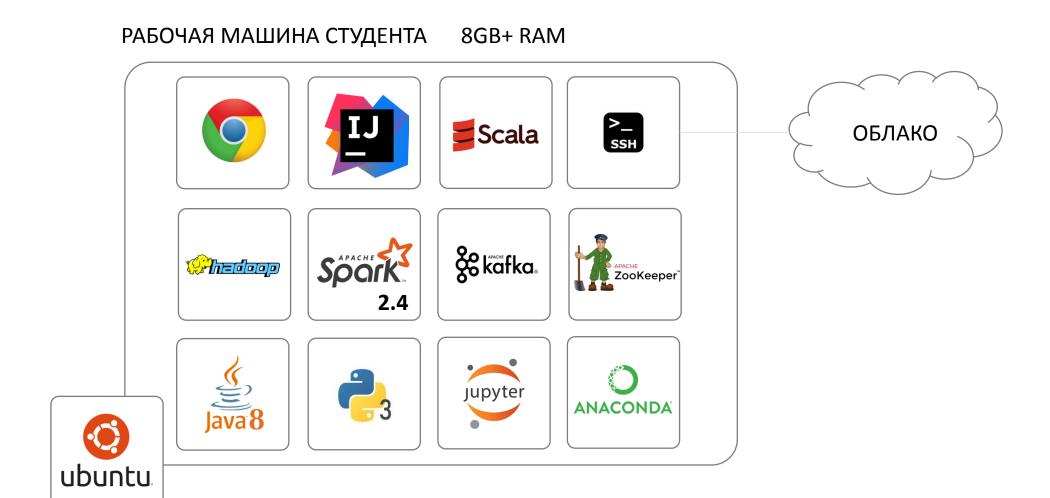
- Часть 1 (10 баллов)
- Часть 2 (10 баллов)

$$PK2 = 35/20 \cdot (K2.1 \cdot Д32. Часть 1 + K2.2 \cdot Д32. Часть 2)$$

РК2 =
$$K2.1 \cdot Д32$$
. Часть 1 + $K2.2 \cdot Д32$. Часть 2 + Вопросы (15 баллов)



Образ с предустановленным ПО



https://github.com/bigdataprocsystems

Ядро Linux





Основные темы

- Особенности Linux и версии
- Основные функции и архитектура ядра
- Компоненты ядра
- Системный вызов
- Процессы и потоки



Исполняемый файл

gcc -o hello hello.c

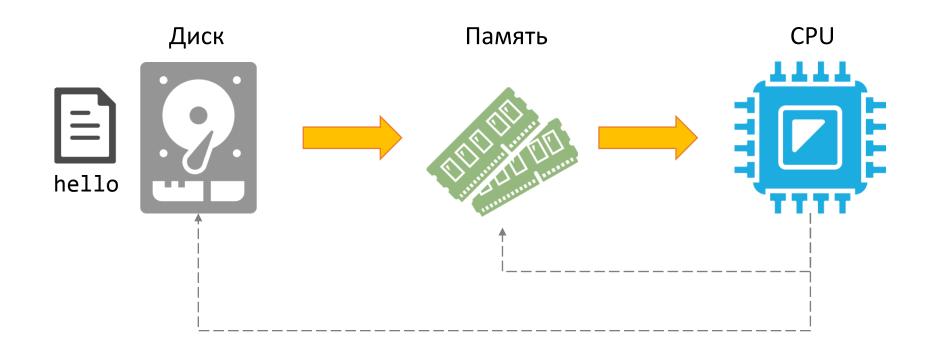


- preprocessing
- compilation
- assembly
- linking



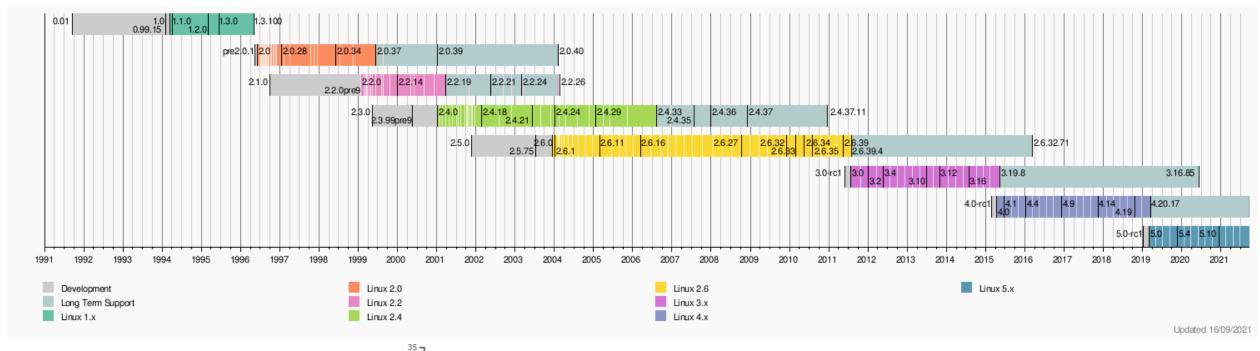
Запуск исполняемого файла

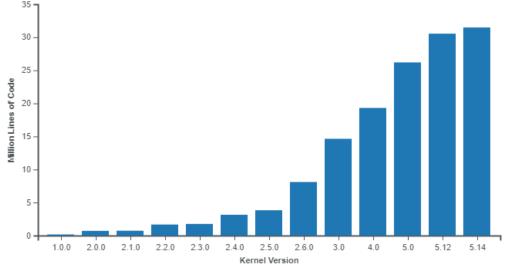
Ядро Linux управляет системными ресурсами и обеспечивает коммуникацию между программными и аппаратными компонентами.





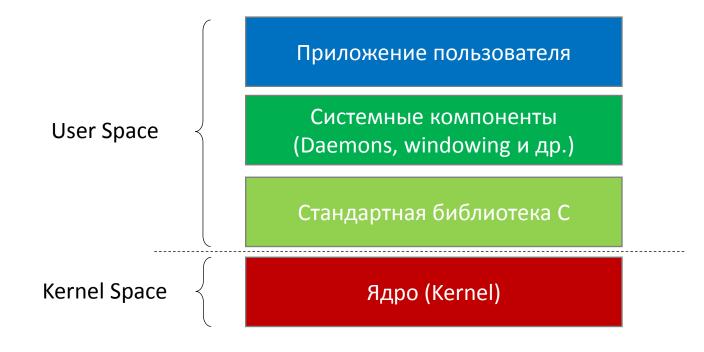
Версии Linux





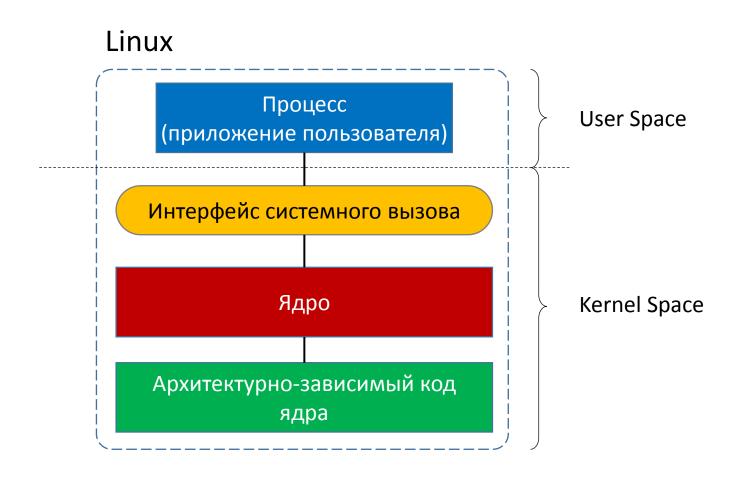


Уровни ОС Linux





Архитектура OC Linux





Основные функции ядра ОС

ОС обеспечивает управление

- процессами (Process management)
- памятью (Memory management)
- > файловыми системами (Filesystems)
- устройствами (Device control)
- > сетью (Networking)



Основные функции ядра ОС

Интерфейс системного вызова

Process management

Memory management

Filesystems

Device Networking

Подсистемы ядра

Параллельность, многозадачность

Виртуальная память Файлы и папки: виртуальная ФС

Доступ к устройствам Связь

Architecturedependent code

Memory manager File system types

Block devices

Character devices

Network subsystem

Interface driver

Программная поддержка







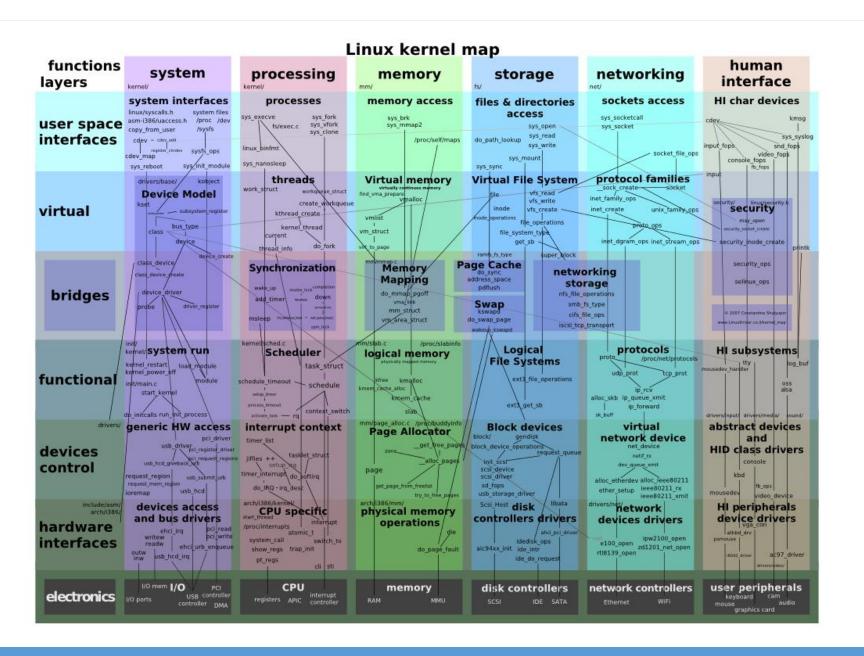




Аппаратные средства



Более сложная схема



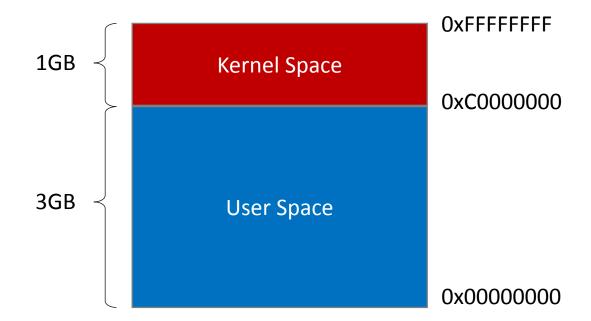


Образ процесса

 Информация о процессе	Идентификатор, состояние, информация об управлении процессом
Раздел стеков	Стек пользователя, стек ядра
Раздел данных	Инициализированные глобальные переменные, внешние переменные, статические переменные
Раздел кода	Код программы

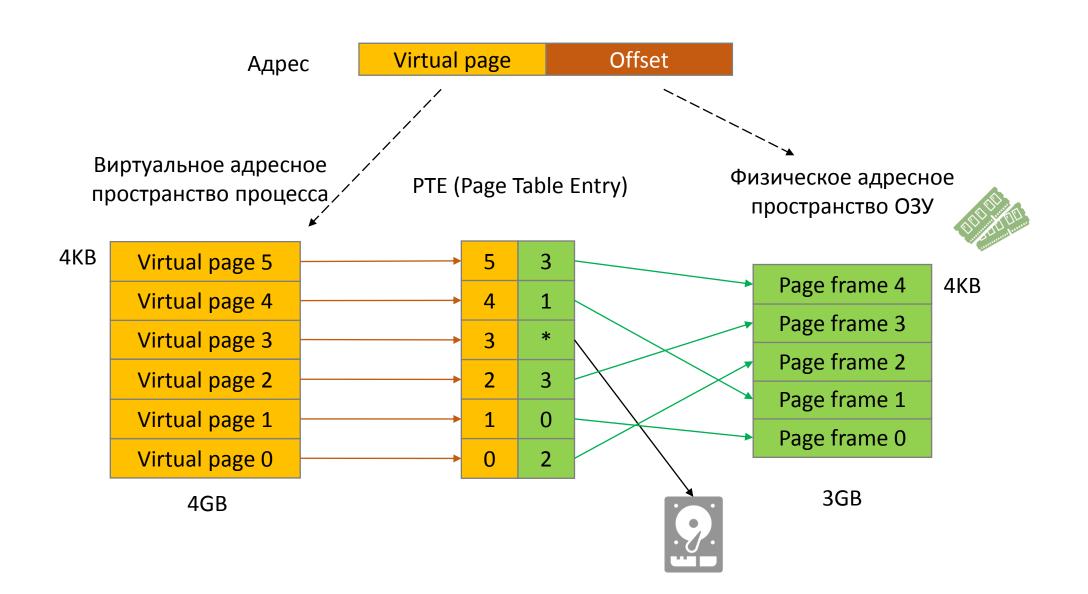


Пространства процесса (32-разрядное)



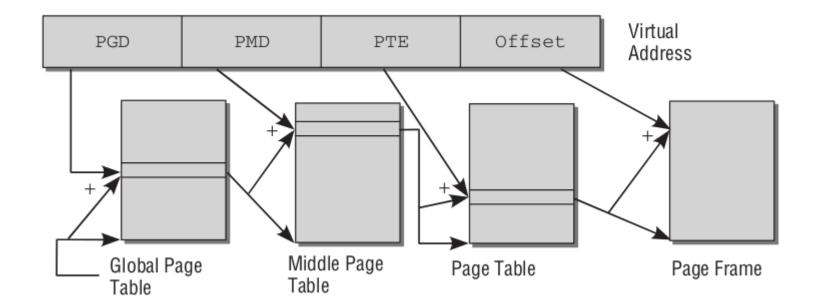


Виртуальные страницы vs физические страницы



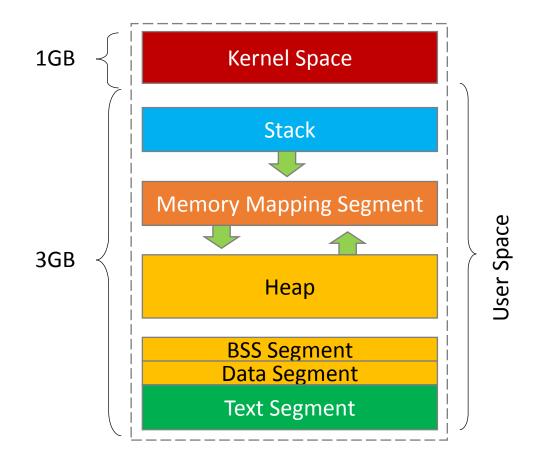


Виртуальные страницы vs физические страницы



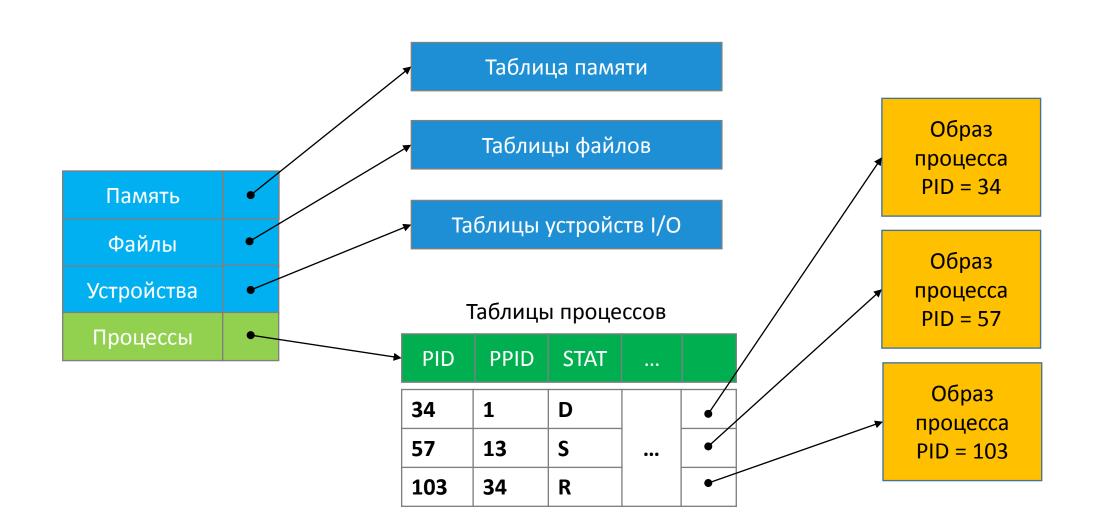


Виртуальное адресное пространство процесса (32-разрядное)



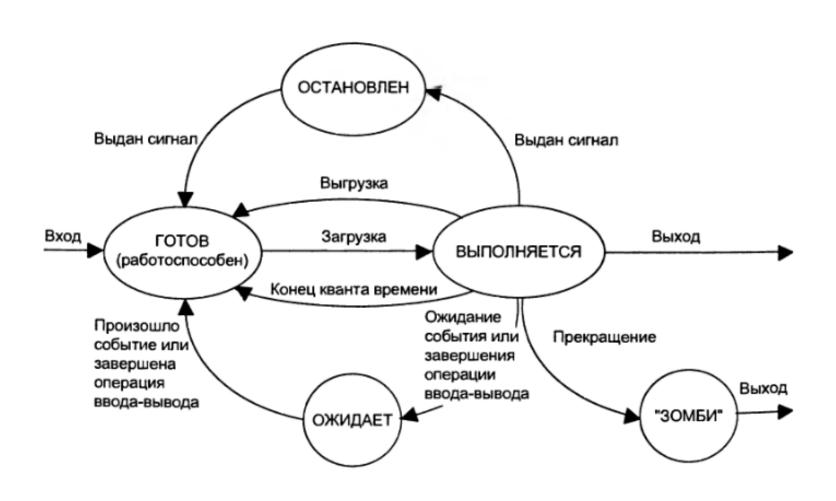


Таблицы ОС





Состояния процесса



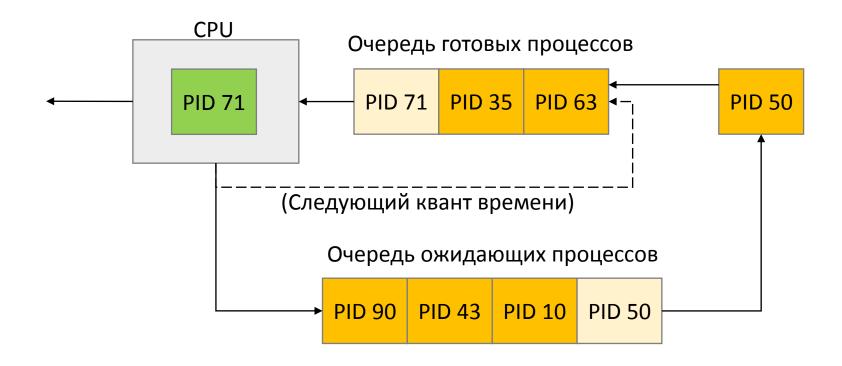


Планировщики выполнения

- FIFO
- Round Robin
- **O**(n)
- **O**(1)
- Completely Fair Scheduler

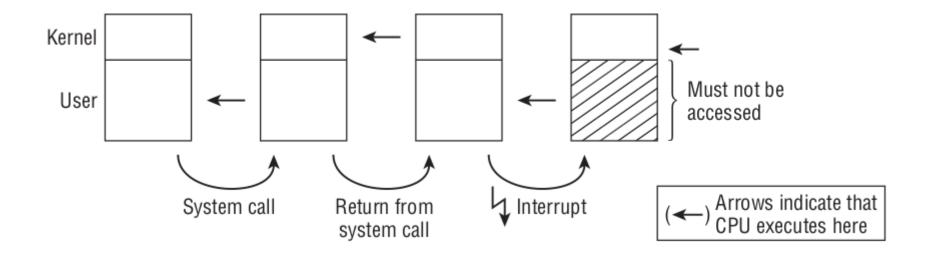


RR планирование выполнения



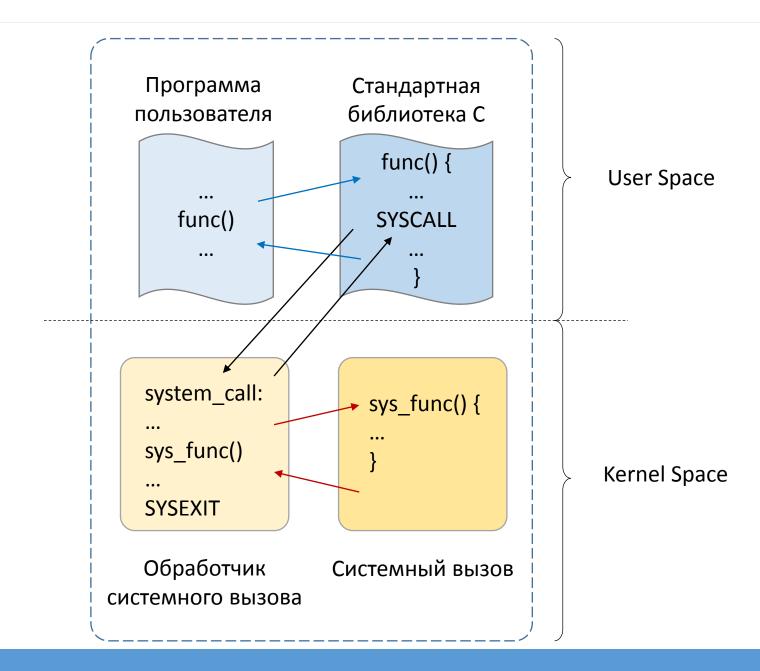


Системный вызов





Системный вызов





Процесс vs поток

Информация о процессе

Раздел стеков

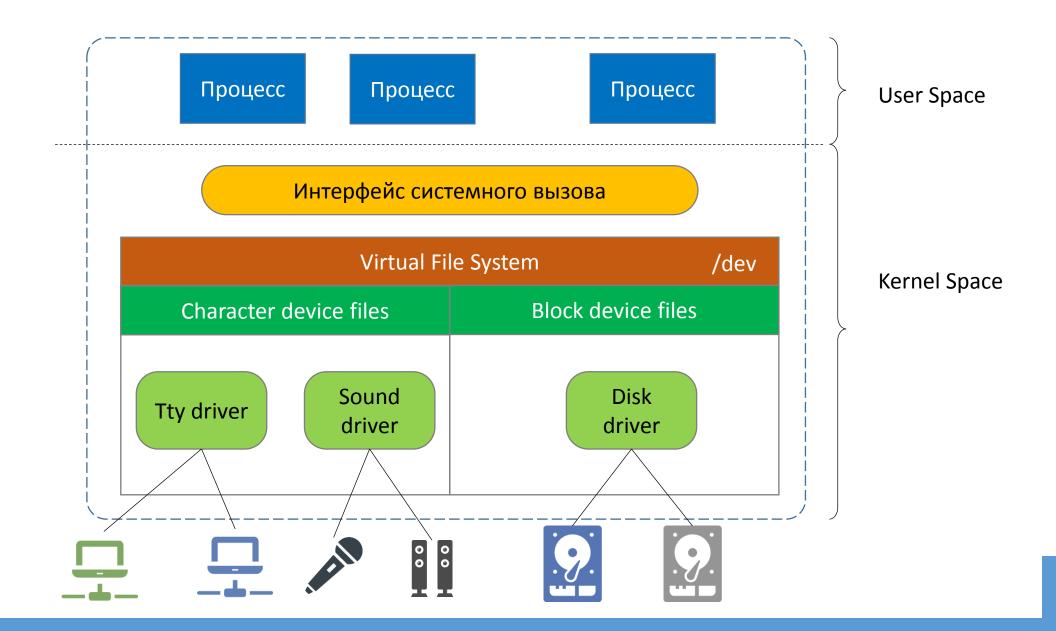
Раздел данных

Раздел кода

Раздел стеков	Раздел данных	Раздел кода
Локальные переменные	 Глобальные структуры данных Глобальные переменные Константы Статически переменные 	
Стек потока А	ДеревьяГрафы	— Код потока А
Стек потока В	ОчередиСтеки	— Код потока В



Драйверы





Источники

Камерон Хьюз, Трейси Хьюз «Параллельное и распределенное программирование с использованием С++»

Understanding the Linux. Third edition by Daniel P. Bovet and Marco Cesati. Third edition

Linux Device Drivers, by Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman