ИТМО

Android vs Linux

Курс: Обеспечение безопасности мобильных устройств

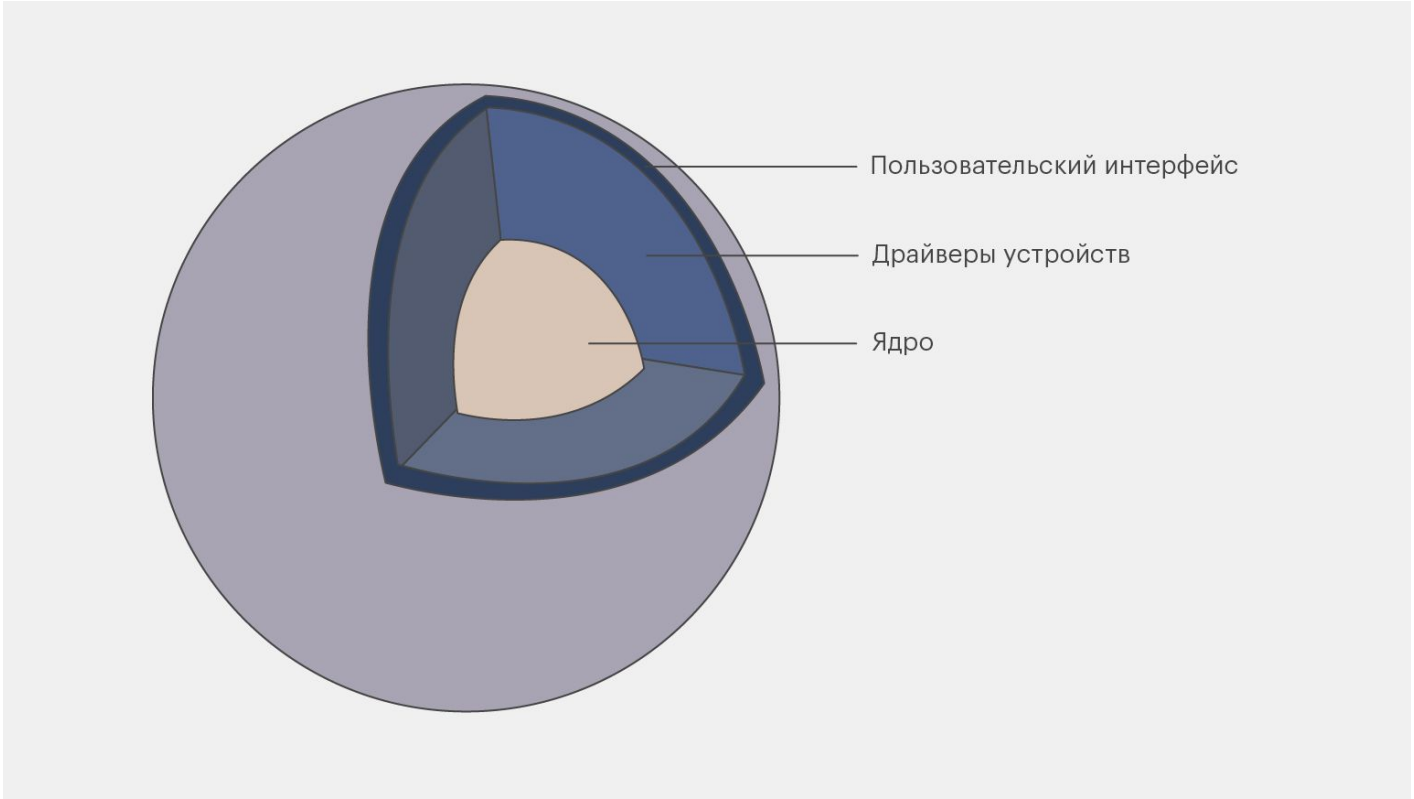
Преподаватель: доцент ФБИТ, к.т.н, Федоров Иван Романович

GNU Linux

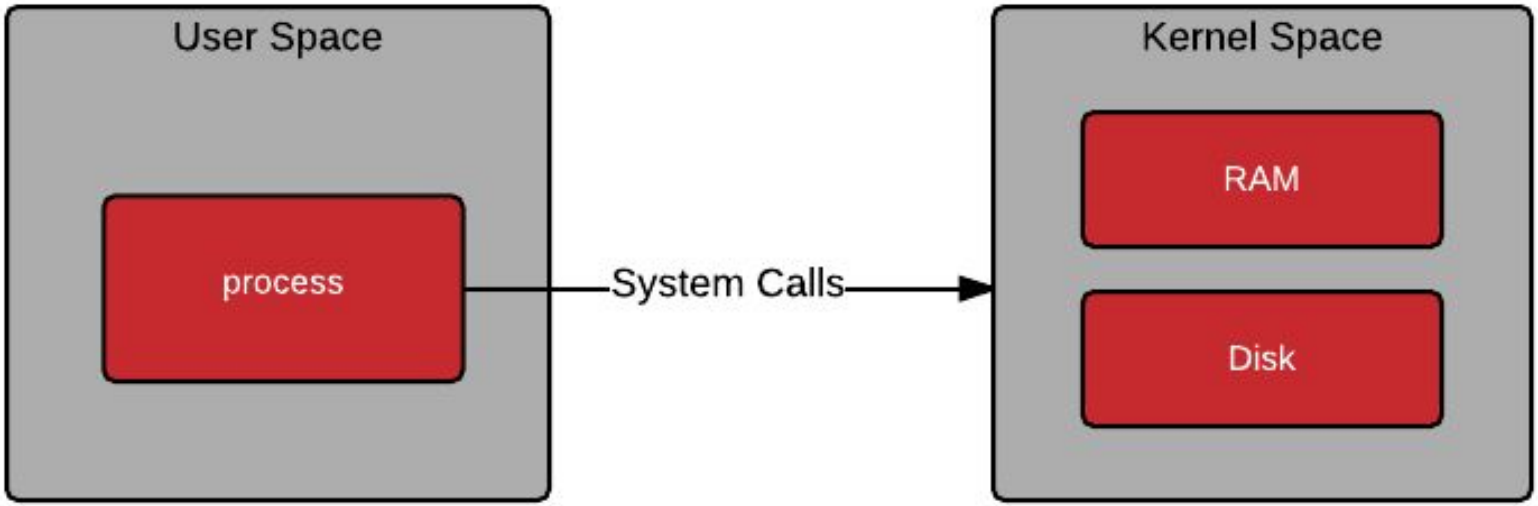
- Автор – Линус Торвальдс
- Первая версия – 1991 год
- На первых этапах связан ОС Minix
- Ядро – Linux
- Утилиты, библиотеки – проект GNU
- GNU/Linux – точное название
- Наследует многие принципы UNIX-систем

Состав ОС

ИТМО

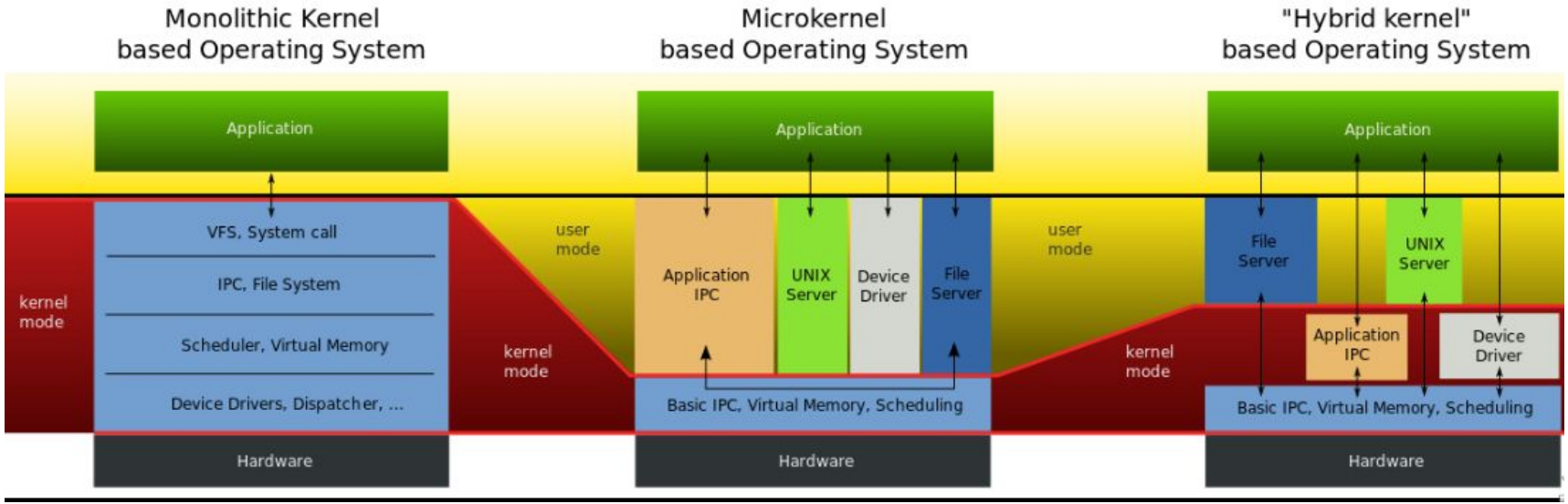


Kernel space/User space



Какие бывают ядра

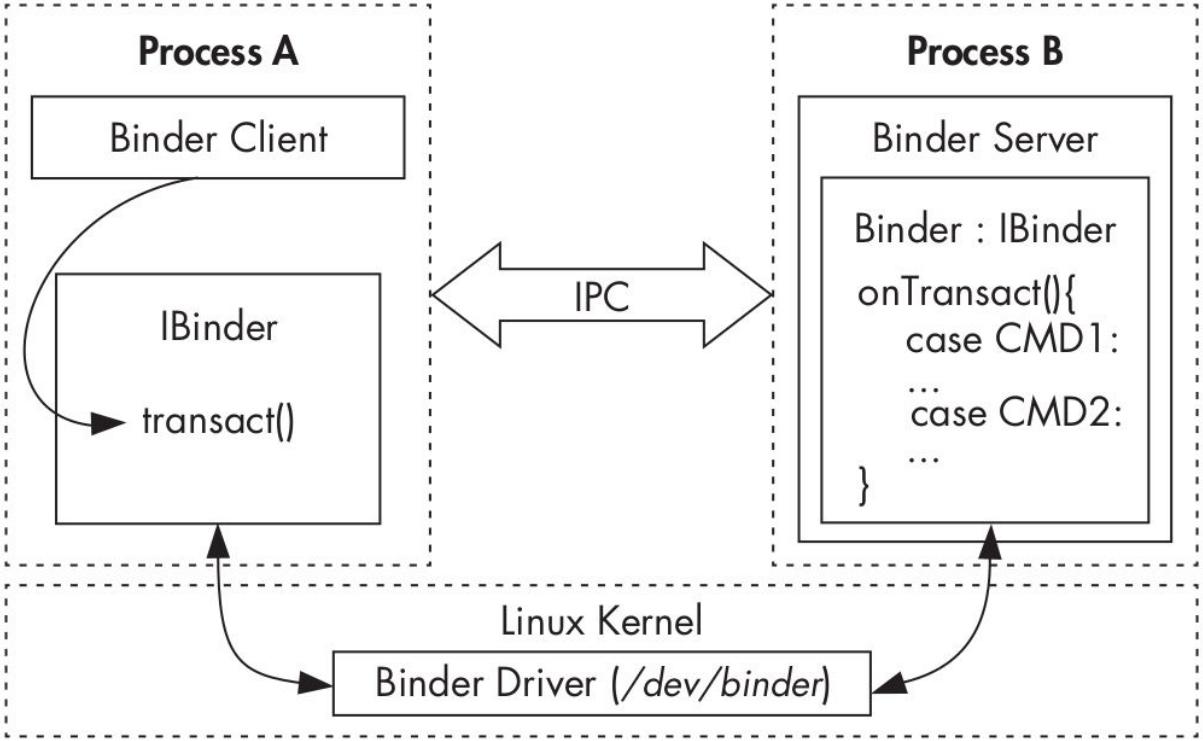
ИТМО



Особенности Android

- **1999 —** Построен на основе Linux.
- Отсутствуют привычные компоненты GNU/Linux-систем (systemd, X.Org и др.)
- **“**no GPL in userspace”
- Ядро модифицировано (добавлены ashmem, Binder driver, wakelocks, low memory killer)
- В качестве libc не GNU C library, a bionic
- Основная единица — приложение

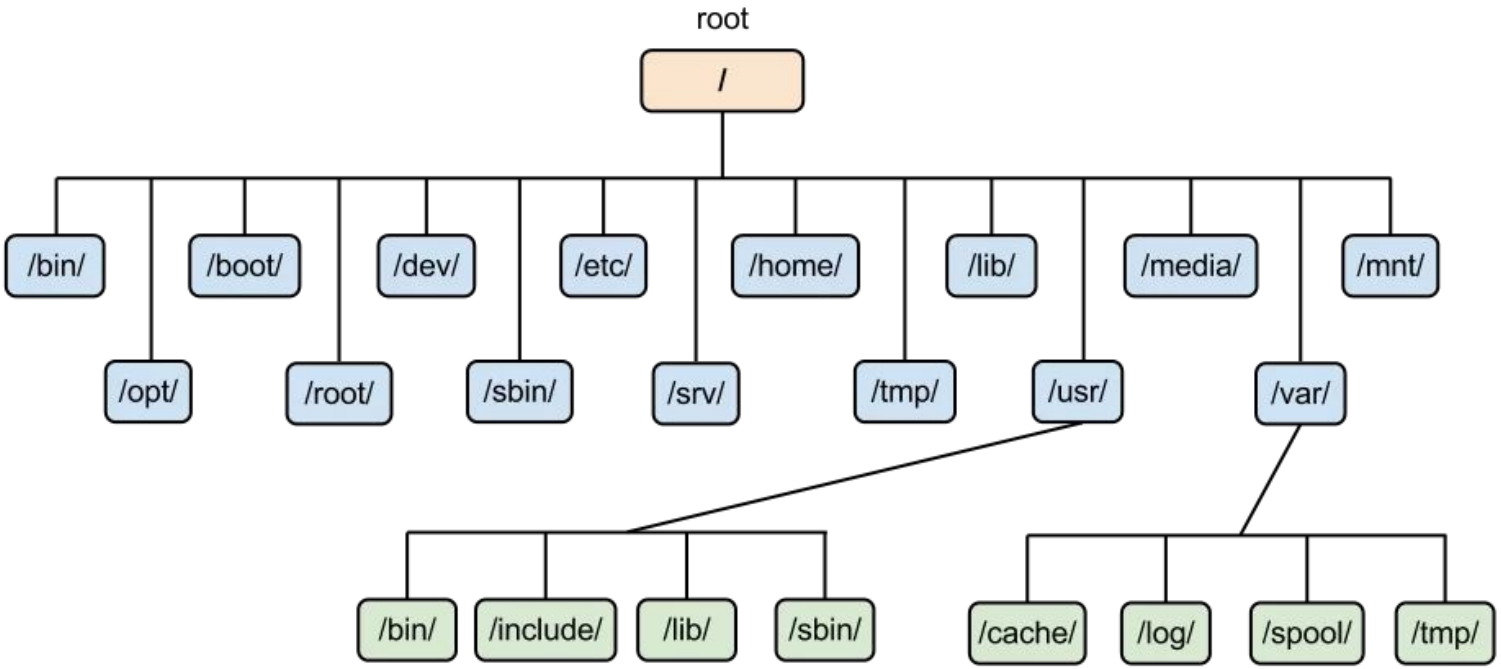
Binder


Process A
Binder Client
IBinder
transact()
IPC
Process B
Binder Server
Binder : IBinder
onTransact(){
case CMD1:
...
case CMD2:
...
}
Linux Kernel
Binder Driver (/dev/binder)

App lifecycle

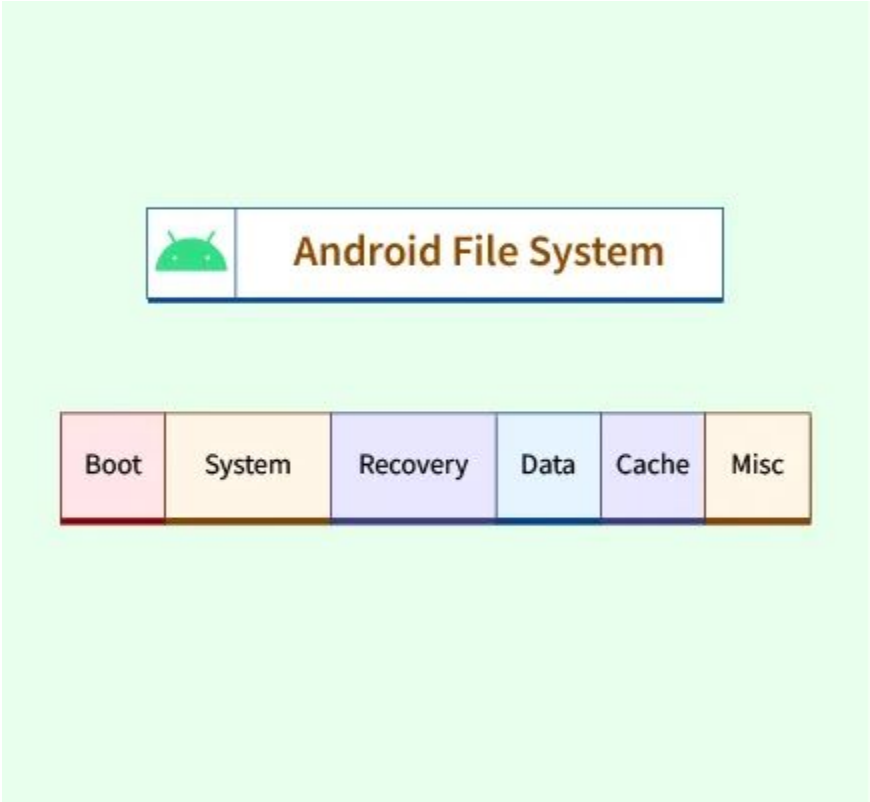
Структура каталогов Linux

ИТМО



Структура каталогов Android

ИТМО



Какие типы файлов вы знаете?

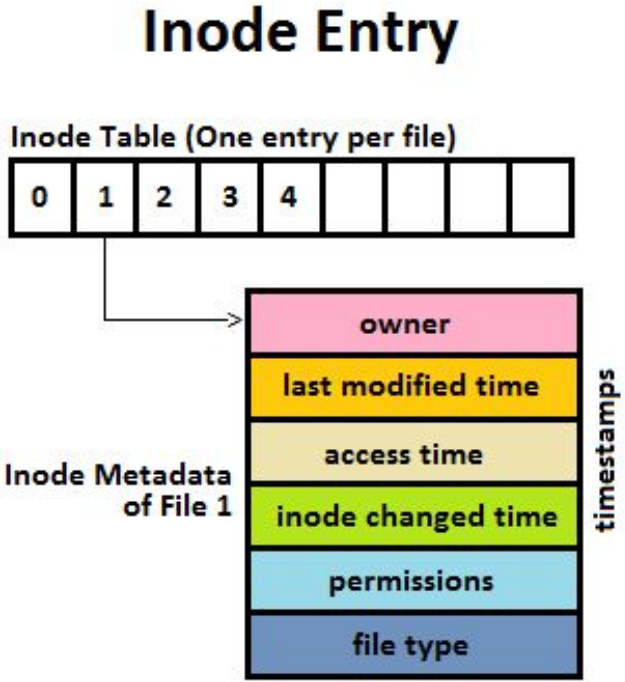
ITMO more than a UNIVERSITY

Все есть файл

ИТМО

Normal	-	**Normal file**
Directories	d	**Normal directory**
Hard link	-	
Symbolic link	l	**Shortcut to a file or directory**
Socket	s	**Pass data between 2 process**
Named pipe	p	**Like sockets, user can't work directly with**
Character device	c	**Processes character hw communication**
Block device	b	**Major & minor numbers for controling dev.**

Inode


Inode Entry
Inode Table (One entry per file)
0
1
2
3
4
Inode Metadata of File 1
owner
last modified time
access time
inode changed time
permissions
file type
timestamps

Место на диске есть, а файл создать нельзя, в чем причина?

ITMO more than a UNIVERSITY

Права доступа к файлам

ИТМО

Права доступа к файлам

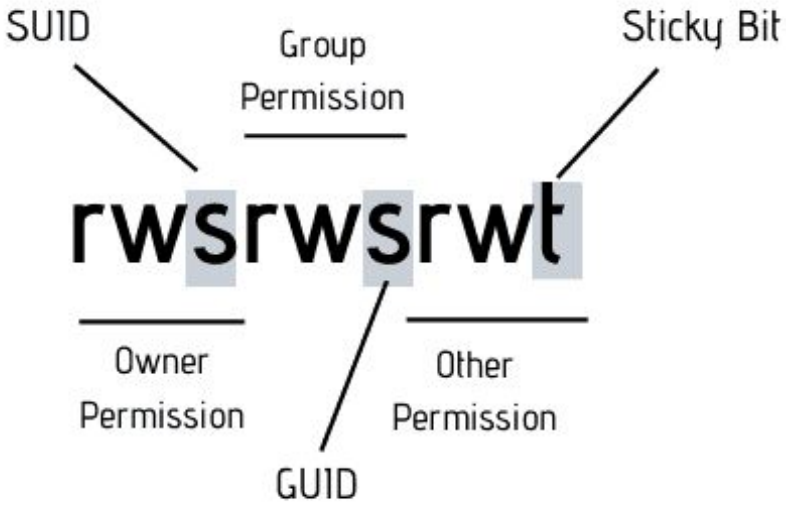
ИТМО

```
0645
```

owner (владелец)	group (группа)	other (остальные)
`rw-`	`r--`	`r-x`
`110`	`100`	`101`
`6`	`4`	`5`

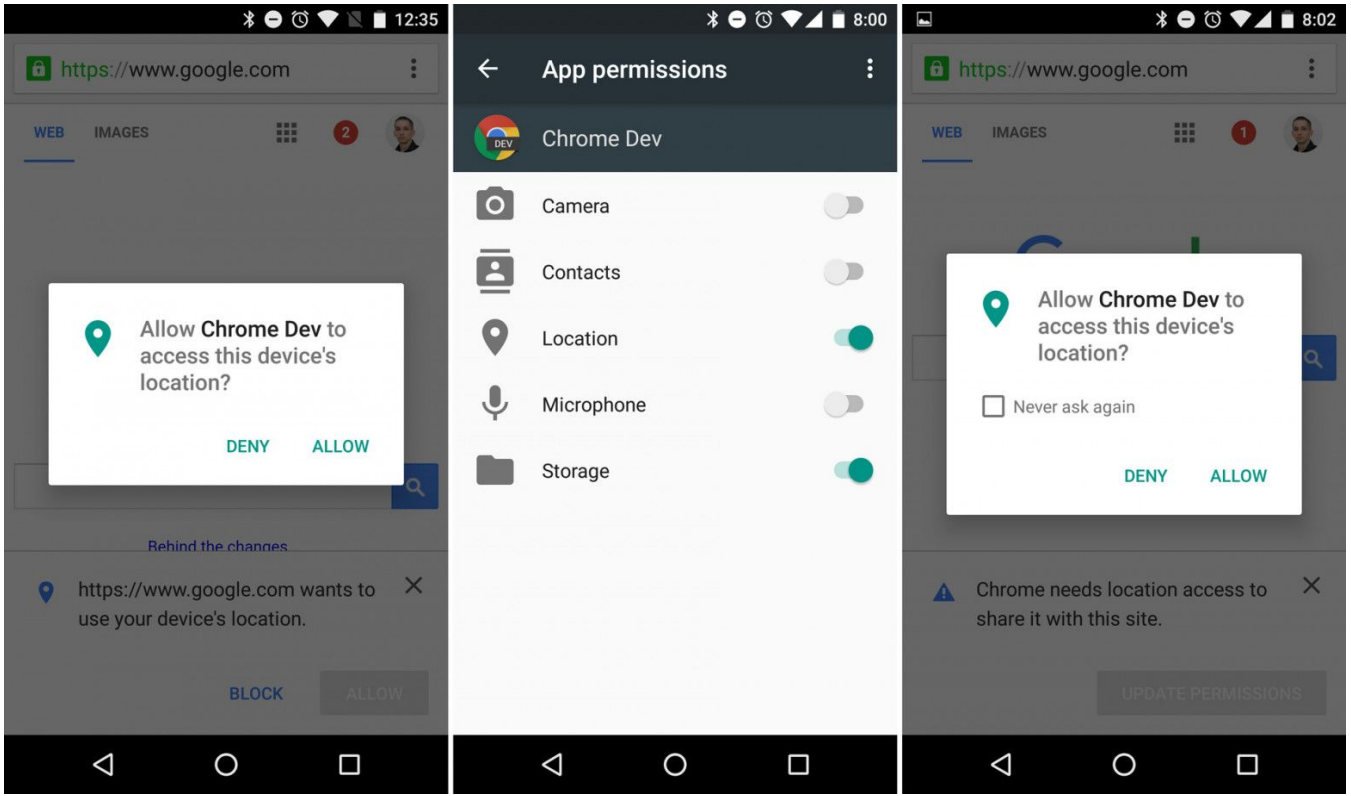
Права доступа к файлам

ИТМО



Права доступа приложений

ИТМО



Перекличка

ITMO more than a UNIVERSITY

SELinux

- **SELinux** (англ. Security Enhanced Linux) — система принудительного (мандатного) контроля доступа (Mandatory Access Control)
- Разработана в АНБ (NSA – National Security Agency)
- Первый релиз – 1998
- Фреймворк LSM (Linux Security Modules — модули безопасности Linux) – 2003
- Первый релиз AppArmor – 2009
- В Android не используются все возможности

DAC vs MAC

ИТМО

Дискреционный механизм доступа (DAC, Discretionary Access Control)

- Каждый файл имеет владельца
- Владелец может передавать права
- Владелец, группа, остальные
- Расширение: Posix ACL

Мандатный механизм доступа (MAC, Mandatory Access Control, матричное управление доступом)

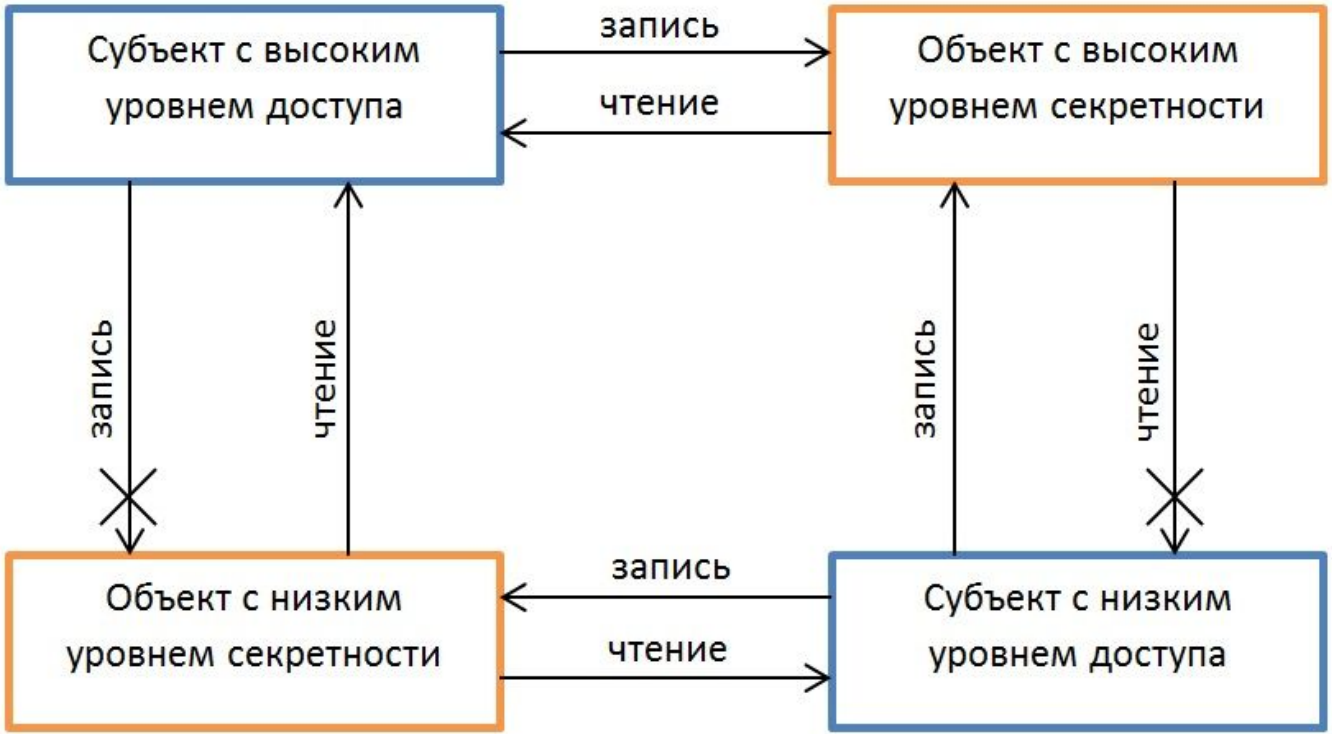
- Явные права на объекты (файлы, устройства, сокеты, порты, процессы)
- Права определяются политиками, а не владельцем
- Модель управления правами домен-тип (домен процесса, тип данных)
- Вариант в Debian: AppArmor

Механизмы MAC

ИТМО

- MLS (Multi-Level Security, многоуровневая система безопасности)
  - Модель Белла-Лападулы
  - Уровни доступа (секретности)
  - Объекты маркируются уровнями доступа
- MCS (Multi-Category Security, мультикатегорийная система безопасности)
  - Данные разбиты на категории
  - Объектам назначаются метки категорий
- RBAC (Roles Based Access Control) — управление доступом на основе ролей
- TE (type Enforcement) — принудительная типизация доступа

Модель Белла — Лападулы



MLS и MCS

ИТМО

Сессия пользователя
с категорией "IT"
и уровнем "Секретно"

Уровень 4
Особой важности

Уровень 3
Совершенно секретно

Уровень 2
Секретно

Уровень 1
Для служебного пользования

Уровень 0
Без уровня конфиденциальности

Категория
IT

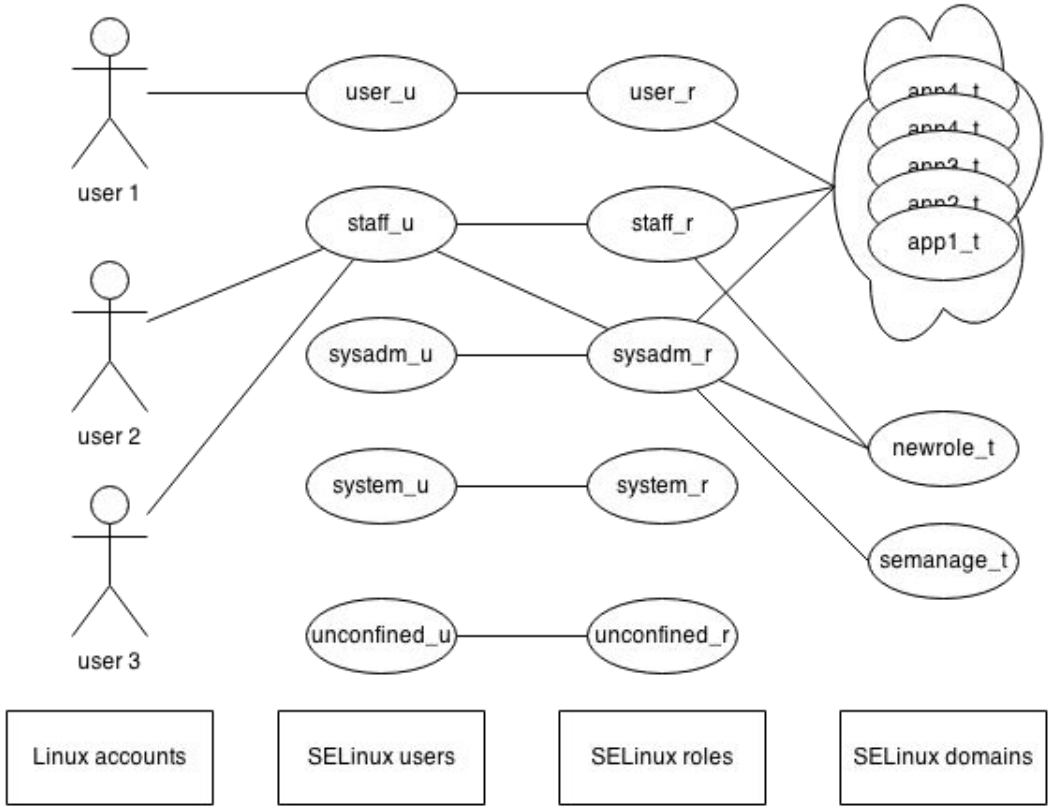
Категория
Finance

Категория
HR

Категория
N

RBAC

ИТМО


user 1
user 2
user 3
user_u
staff_u
sysadm_u
system_u
unconfined_u
user_r
staff_r
sysadm_r
system_r
unconfined_r
app1_t
newrole_t
semanage_t
Linux accounts
SELinux users
SELinux roles
SELinux domains

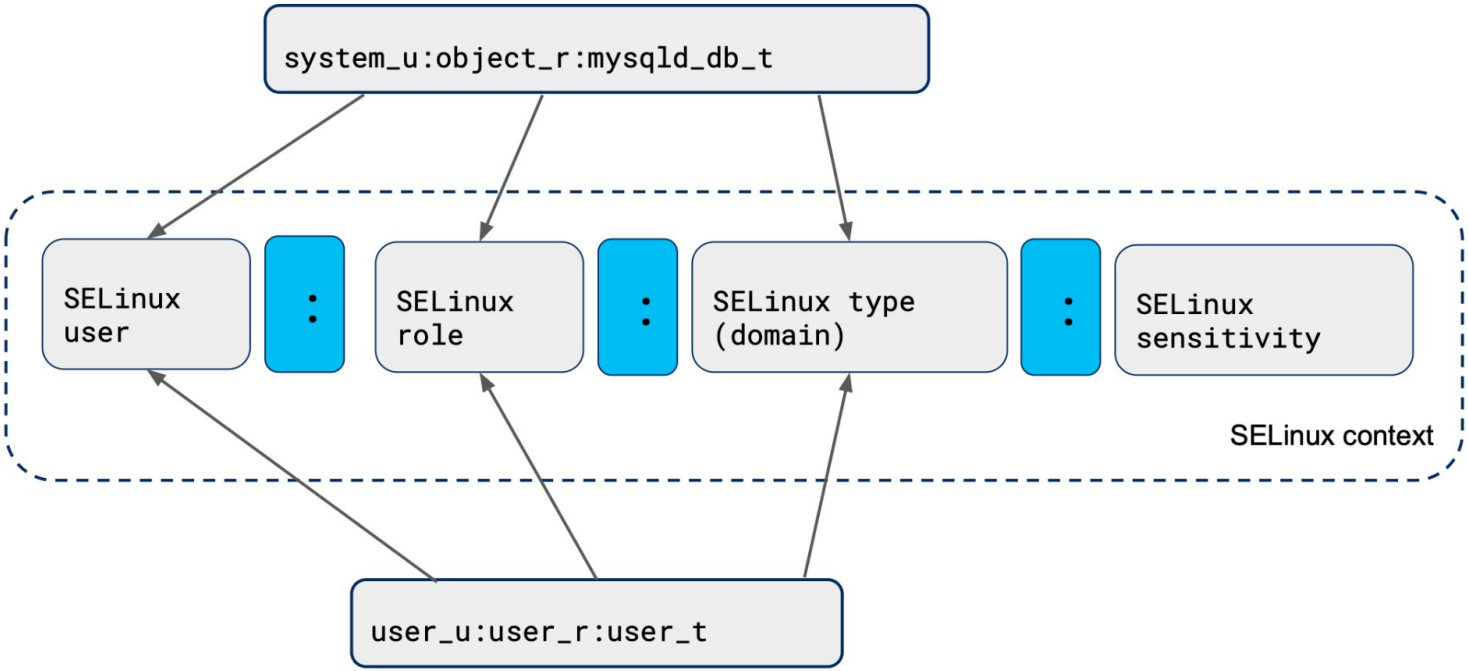
SELinux — термины MAC

ИТМО

- **TE** (Type Enforcement, принудительная типизация доступа)
- **Контекст безопасности** (context) — метка, выглядит как строка переменной длины и хранится в расширенных атрибутах файловой системы. Объединяет в себе роли, типы и домены
- **Домен** (domain) - список действий, которые может выполнять процесс по отношению к различным объектам
- **Тип** (type) - атрибут объекта, который определяет, кто может получить к нему доступ
- **Роль** - атрибут, который определяет, в какие домены может входить пользователь, то есть какие домены пользователь имеет право запускать

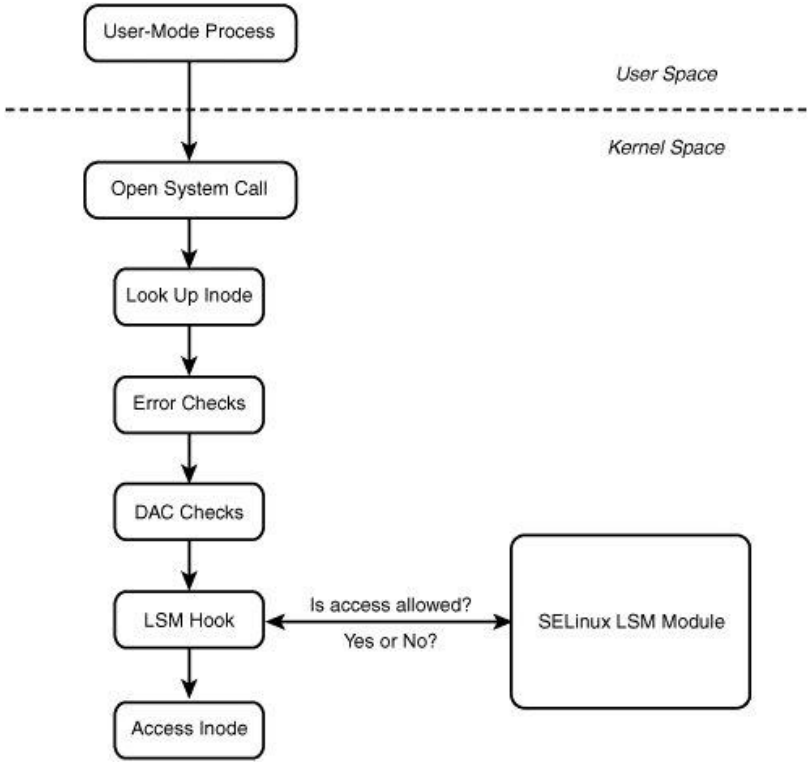
TE – Type Enforcement

ИТМО

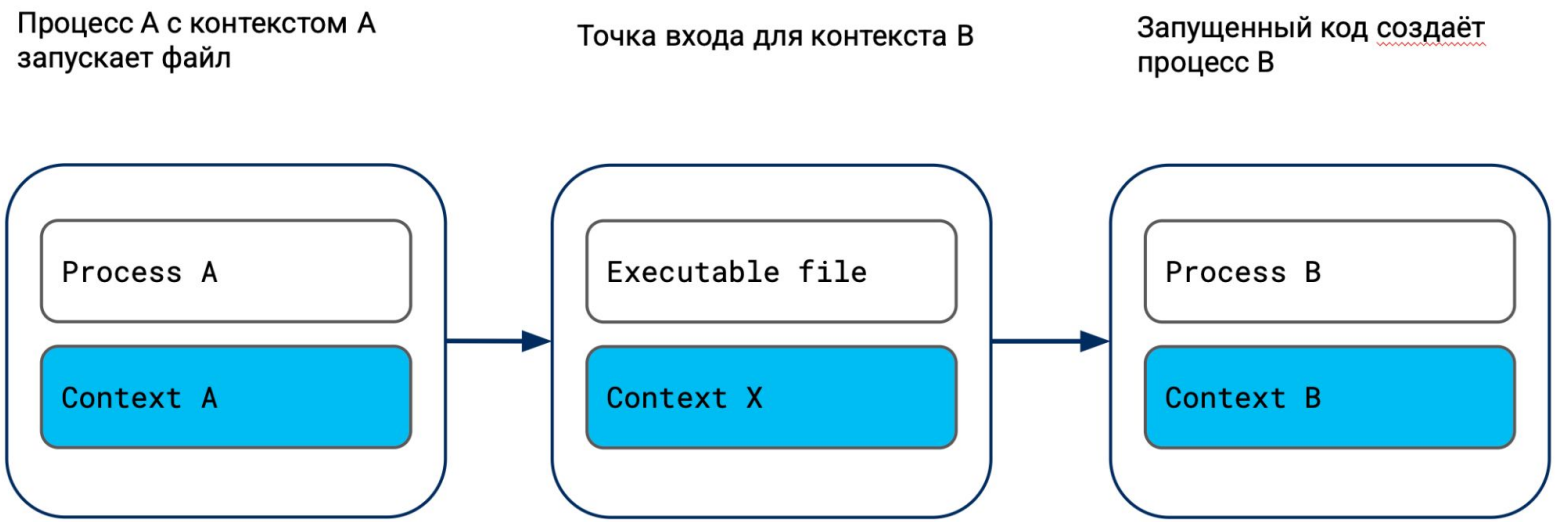


Совместная работа DAC и MAC

ИТМО



Переход контекста



Специфика SELinux в Android

ИТМО

- Большинство политик в AOSP определяются с использованием языка политики ядра. Есть некоторые исключения для использования общего промежуточного языка (CIL)
- Пользователи SELinux не используются. Единственный определенный пользователь — это u . При необходимости физические пользователи представлены с использованием поля категорий контекста безопасности.
- Роли SELinux и управление доступом на основе ролей (RBAC) не используются. Определены и используются две роли по умолчанию: r для субъектов и object_r для объектов.
- Чувствительность SELinux не используется. Чувствительность по умолчанию s0 всегда установлена.
- Логические значения SELinux не используются. Если политика создана для устройства, она не зависит от состояния устройства. Это упрощает аудит и отладку политик.

Ваши вопросы

ITMO more than a UNIVERSITY

Спасибо
за внимание!

ITMO more than a UNIVERSITY

ivanfedorov@itmo.ru
@VanesFedorov