# Темы

(тут разбить можно наверное)

* Управление внешними устройствами, Система прерываний, верхние и нижние половины (все типы как раз со сравнением) - [Билет 1](#_766mbfjvlgf7), [Билет 1 Старый](#_tmm273go1fk8), [Билет 2](#_ohy2hhi51pf4), [Билет 3 старый.](#_fpomiqcsk1fn) [Билет 11](#_hbvi5zgomiq)
* Управление устройствами: драйверы, 4 типа передачи данных. - [Билет 5](#_zgl7fwgizdb), [Билет 5 старый](#_10kduauco9xe),[Билет 18 (не знаю год)](#_g0xbiwdby832) [Билет 19 (не знаю год)](#_pxaz8593zq3c),

(тут прям вместе)

* Классификация типов ввода-вывода, моделей ввода-вывода. Мультиплексирование (сокеты) - [Билет 3](#_ntf57r4yibwc) [Билет 7](#_cdf31f9req5j) [Билет 7 старый.](#_dmwqkxcix8q1)
* Сокеты [Билет 8](#_d27lc738wc6k) [Билет 8 старый](#_y3dy387c471c)

(этот блок надо разбивать, но пересечений много прям)

* Файловая подсистема /ргос. Передача данных из адресного пространства пользователя в адресное пространство ядра. - [Билет 4](#_edxnrm9iepf3), [Билет 4 старый](#_ikdpg0t9ewt5) [Билет 14](#_fyczadv2n3eo)
* Файловая система, VFS, создание и монтирование - [Билет 6](#_e9wq6u2nwkop) [Билет 6 старый.](#_390sdjjvmjok) [Билет 10](#_88hwsomqlb9x) [Билет 10 старый](#_4qbcuqqj7vt7) [Билет 12](#_kk9ikl13ibxo) [Билет 13](#_s57tg0nste5w) [Билет 13 старый](#_kzhy8u2hyuz6) [Билет 14 старый](#_8f6l88to0r67) (он еще в теме ниже) [Билет 15 старый](#_rt33587yiq5b)
* Загружаемые модули ядра, информации о запущенных процессах, printk() Регистрация функций работы с файлами [Билет 15](#_ptecr6p9vba7)
* файловая система (лаба про разные открывания файлов) - [Билет 9](#_6lebnq1lvhxz) [Билет 9 старый](#_22v1aa2bzkjr) [Билет 14 старый](#_8f6l88to0r67) (он еще в теме выше)
* чисто про open [Билет 15](#_ptecr6p9vba7) [Билет 15 старый](#_rt33587yiq5b) [Билет 16](#_9nfpatthjpgo) [Билет 16 старый](#_9lmugsz0h8cw)

(пересекаются: 14)

# **Билет 1**

Управление внешними устройствами: специальные файлы устройств, адресация внешних устройств и их идентификация в системе, тип dev\_t. Система прерываний: типы прерываний и их особенности. Прерывания в последовательности ввода-вывода — обслуживание запроса процесса на ввод-вывод (диаграмма). Быстрые и медленные прерывания. Обработчики аппаратных прерываний: регистрация в системе — функция и ее параметры, примеры. Тасклеты — объявление, планирование (пример лаб. раб).

# Билет 1 старый

Управление внешними устройствами: подключение внешних устройств и их идентификация в системе. Система прерываний: типы прерываний и их особенности, прерывания в последовательности ввода-вывода — обслуживание запроса процесса на ввод-вывод. Быстрые и медленные прерывания. Обработчики аппаратных прерываний: регистрация в системе, примеры. Тасклеты — объявление, планирование (пример лабораторной работы)

# Билет 2

Управление внешними устройствами: специальные файлы устройств, идентификация внешних устройств в системе (тип dev\_t), символьные и блочные устройства и их inode (структуры,описывающие символьные и блочные устройства). Система прерываний: типы прерываний и их особенности. Быстрые и медленные прерывания. Обработчики аппаратных прерываний: регистрация в системе, примеры. Верхние и нижние половины обработчиков прерываний. Нижние половины: тасклеты и очереди работ — объявление, создание, постановка работы в очередь, планирование (пример лаб. раб).

# Билет 3

Модели ввода-вывода: представление с помощью диаграмм, описание и особенности. Классификация моделей ввода-вывода. Мультиплексирование при взаимодействии процессов в распределенных системах по модели клиент-сервер. Сетевой стек. Примеры мультиплексоров и пример из лабораторной работы.

# **Билет 3 старый**

Система прерываний: типы прерываний и их особенности. Быстрые и медленные прерывания. Обработчики прерываний: деление на верхнюю и нижнюю половины; обработчики аппаратных прерываний-регистрация в системе, разделение линии IRQ и отложенные действия: softirg, тасклеты, очереди работ - особенности, сравнение, примеры (лабораторная работа).

# Билет 4

Файловая подсистема /proc — назначение, особенности, файлы, поддиректории, ссылка self, информация об окружении, состоянии процесса, прерываниях. Структура proc\_dir\_entry: функции для работы с элементами /proc. Использование структуры file\_operations для регистрации функций работы с файлами. Передача данных из адресного пространства пользователя в адресное пространство ядра и обратно (лаб. раб.). Обоснование необходимости использования специальных функций для передачи данных из пространства пользователя в ядро и из ядра в пространство пользователя.

# Билет 4 старый

Файловая подсистема /ргос: назначение, особенности, файлы, поддиректории, ссылка, self, interrupt. Структура proc\_dir\_entry: функции для работы с элементами /ргос. Использование структуры file\_operations для регистрации функций работы с файлами. Передача данных из адресного пространства пользователя в адресное пространство ядра. Пример передачи данных из пространства пользователя в пространство ядра и обратно, вывод информации о выполняемых в системе процессах (лабораторная работа).

# Билет 5

Управление устройствами: абстракция устройств, типы устройств и идентификация в ядре Unix/Linux. Управление устройствами: драйверы. USB-шина: особенности, хост и хабы, конечные точки и каналы, 4 типа передачи данных. Структура USB-драйвера (struct usb\_driver), таблица id\_table, основные точки входа драйвера USB. Регистрация usb-драйвера в системе

# **Билет 5 старый**.

Управление устройствами: абстракция устройств, типы устройств и идентификация Unix/Linux. Драйверы и обработчики прерываний в Linux. USB-шина: особенности, usb-core, хост и конечные точки, 4 типа передачи данных. Структура USB-драйвера (struct usb\_driver), таблица id\_table, основные точки входа драйвера USB. Регистрация usb-драйвера в системе.

# Билет 6

Файловая система, задачи файловой системы и иерархическая организация ФС. Файловая подсистема LINUX: поддержка большого числа файловых систем и структура, описывающая файловую систему. VFS: четыре основные структуры файловой системы и связь между ними. Раздел жесткого диска и суперблок. Структура struct super\_operations. Монтирование файловой системы, команда mount и функции монтирования и их параметры, точка монтирования – корневой каталог и inode. Пример (лаб. раб.)

# Билет 6 старый.

Файловая система, задачи файловой системы и иерархическая организация ФС. Файловая подсистема Linux: поддержка большого числа файловых систем. VFS: четыре основные структуры файловой системы и связь между ними. Раздел жесткого диска и суперблок, точка монтирования-корневой каталог и inode. Пример (лабораторная работа).

# Билет 7

Классификация типов ввода-вывода с точки зрения программиста: диаграммы последовательности действий для каждого типа ввода-вывода и описание. Классификация моделей ввода-вывода. Особенности и назначение асинхронного ввода-вывода. Мультиплексирование. Пример мультиплексирования для сокетов AF\_INET, SOCK\_STREAM. Сетевой стек. Пример (лаб. раб.)

# Билет 7 старый.

Классификация типов ввода-вывода с точки зрения программиста: описание и диаграммы последовательности действий для каждого типа ввода, вывода. Пример мультиплексирования для сокетов AF\_INET, SOCK\_STREAM. Пример (лабораторная работа).

# Билет 8

Средства взаимодействия процессов — сокеты Беркли. Создание сокета — семейство, тип, протокол. Системный вызов sys\_socket() и struct socket. Состояния сокета. Адресация сокетов и ее особенности для разных типов сокетов. Модель клиент-сервер. Сетевые сокеты — сетевой стек, аппаратный и сетевой порядок байтов. Примеры реализации взаимодействия процессов по модели клиент-сервер с использованием сокетов и мультиплексированием (лаб. раб.).

# Билет 8 старый

Средства взаимодействия процессов-сокеты Беркли. Типы сокетов. Адресация. Сокеты AF\_UNIX. Сетевые сокеты — сетевой стек, аппаратный и сетевой порядок байтов. Примеры реализации взаимодействия процессов по модели клиент-сервер с использованием сокетов (лабораторная работа).

# Билет 9

Файловая система: процесс и файловые структуры связанные с процессом. Файлы и открытые файлы, связь структур, представляющих открытые файлы на разных уровнях. Системный вызов open() и библиотечная функция fopen(): параметры и флаги, определенные на функции open(). Реализация системного вызова open() в ядре Linux. Пример: файл открывается два раза системным вызовом open() для записи и в него последовательно записывается строка «аааааааааааа» по первому дескриптору и затем строка «вввв» по второму дескриптору, затем файл закрывается два раза. Показать, что будет записано в файл и пояснить результат.

# Билет 9 старый

Файловая система: файлы и открытые файлы, процесс и файлы открытые процессом. Структуры описывающие процесс и открытые им файлы в системе, основные поля структур. Пример: файл два раза открывается для записи и в него последовательно записывается строка “aaaaaaaa” и строка “bbbb”, затем файл закрывается два раза. Показать, что будет записано в файл и пояснить результат, если используется библиотека буферизированного ввода/вывода.

# Билет 10

Создание собственной файловой системы. Структура, описывающая файловую систему и пример ее заполнения. Регистрация и дерегистрация файловых систем. Монтирование файловой системы. Структура struct super\_operations. Структура inode\_operations. Функции simple и generic. Точка монтирования. Функции монтирования. Функция printk(). Пример создания файловой системы, ее регистрация и монтирование (лаб. раб.).

# Билет 10 старый

Создание собственной файловой системы. Регистрация и дерегистрация файловых систем. Монтирование файловой системы. Пример файловой системы, ее регистрации и монтирования (лабораторная работа).

# Билет 11

Аппаратные прерывания в Linux: запрос прерывания и линии IRQ. Простейшая схема аппаратной поддержки прерываний (концептуальная трехшинная архитектура системы). Быстрые и медленные прерывания, пример быстрого прерывания, флаги. Нижняя и верхняя половины обработчиков прерываний: регистрация обработчика аппаратного прерывания, функция регистрации и ее параметры. Нижние половины: softirq, tasklet, work queue — особенности реализации и выполнения в SMP-системах. Примеры, связанные с планированием отложенных действий (лаб. раб.)

# Билет 12

Создание виртуальных файловых систем. Структура, описывающая файловую систему. Регистрация и дерегистрация файловой системе. Монтирование файловой системы. Точка монтирования. Кэширование в системе. Кэши SLAB, функции для работы с кэшем SLAB. Примеры из лабораторной работы. Функции, определенные на файлах (struct file\_operations), функции, определенные на файлах, и их регистрация. Пример из лабораторной работы по файловой системе /proc.

# Билет 13

Файловая подсистема: особенности файловой подсистемы Unix/Linux.: иерархическая структура файловой подсистемы. Виртуальная файловая система VFS в Linux. Четыре структуры VFS – super\_block, inode, dentry, file их назначение. Адресация файлов большого размера в файловой системе extX и пример, показывающий доступ к файлу /usr/ast/mbox. Монтирование файловых систем. Команда mount и функции монтирования, пример из лаб. раб.

# Билет 13 старый

Файловая подсистема: особенности файловой подсистемы Unix/Linux.: иерархическая структура файловой подсистемы. Монтирование файловых систем. Виртуальная файловая система - интерфейс vfs, суперблок, индексные узлы inode и dentry. Адресация файлов большого размера и пример, показывающий доступ к /usr/ast/mbox.

# Билет 14

Файловая подсистема /proc – назначение, особенности, файлы, поддиректории, ссылка self, информация об окружении, состоянии процесса, прерываниях. Структура proc\_dir\_entry: функции для работы с элементами /proc. Структура, перечисляющая функции, определенные на файлах. Использование структуры file\_operations для регистрации собственных функций работы с файлами. Передача данных их пространства пользователя в пространство ядра и из ядра в пространство пользователя. Обоснование необходимости этих функций. Функция printk() – назначение и особенности. Пример программы «Фортунки» из лаб. работы.

# Билет 14 старый

Файловая система Linux - открытые файлы: структуры, связанные с процессом и struct file, struct file\_operations. Примеры. Регистрация и дерегистрация файловых систем. Примеры (лаб. раб.)

# Билет 15

Загружаемые модули ядра. Структура загружаемых модулей. Информация о процессах, доступная в ядре. Пример вывода информации о запущенных процессах, символ current (лаб. раб.). Взаимодействие загружаемых модулей в ядре. Экспорт данных. Пример взаимодействия модулей (лаб. раб.). Функция printk() – назначение и особенности. Регистрация функций работы с файлами. Пример заполненной структуры. Передача данных из пространства ядра в пространство пользователя и из пространства пользователя в пространство ядра. Примеры из лабораторный работ.

# Билет 15 старый

Файловые системы. Иерархическая структура файловой системы. Задачи уровней файловой системы. Доступ к файлам - работа с директориями: struct dentry, cтруктура inode каталога. Пример, демонстрирующий доступ файлу.

# Билет 16

Открытые файлы: системный вызов open(): int open(const char\* pathname, int flags); int open(const char\* pathname, int flags, mode\_t mode); пояснить смысл параметров. Основные флаги. Флаг CREATE. Реализация системного вызова open() в системе – действия в ядре: SYSCALL\_DEFINE3(open,…) -> ksys\_open(filename, flags, mode) -> do\_sys\_open() -> do\_sys\_openat2() … найти наименьший файловый дескриптор... Действия, если флаг O\_CREATE установлен?

# Билет 16 старый

Открытые файлы: системный вызов ореп(): int open(const char\* pathname, int flags); int open(const char pathname, int flags, mode\_t mode); пояснить смысл параметров. Реализация системного вызова open() в системе: связь функций ореn(), sys\_open(), filp\_open(). f\_op->ореn(). Назначение inode кешей и их виды, slab cache- особенности.

# Билет 18 (не знаю год)

Специальные файлы устройств, каталог /dev, старший и младший номера устройств. Структура usb\_driver: функции probe() и disconnect(), параметры и возвращаемое значение. Обработчики аппаратных прерываний: регистрация. Верхняя и нижняя половины обработчиков прерываний. Примеры тасклета и очереди работ.

# 

# Билет 19 (не знаю год)

Управление устройствами: абстракция и типы устройств и идентификация в Unix/Linux. Драйверы и обработчики прерываний в Linux. USB-шина: особенности, usb-core, хост и конечные точки, 4 типа передачи данных. Структура USB-драйвер (struct usb\_driver), таблица id\_table, основные точки входа драйвера USB, передаваемые им параметры. Регистрация usb-драйвера в системе. Пример.

# Билет 20 (не знаю год)

Файловая подсистема /ргос: тип и предоставляемая информация. Загружаемые модули ядра: правила программирования и сборки. Взаимодействие модулей ядра. Пример. Перемешивание данных между ядром и пользовательским пространством: функции. Функция printk() и стринговый код приоритета(макросы). Пример (лаб.раб.).