

Первый семестр

1. RAM. SRAM, DRAM. Структурно-операционная схема обычной планки памяти (DDR). Кешы CPU: L1, L2, L3, ... Схема параллельного извлечения. Извлечение демультимплексором. Ассоциативность. TLB. Общее влияние кеша на работу с памятью. Кешы в мультимикропроцессорных системах и когерентность кешей.
2. Пайплайн (pipeline) CPU: fetch, decode, ..., execute, commit. Регистровый файл. Пузыри (pipeline bubbles). Предсказание переходов (branch prediction). Out of order исполнение. Интерфейс между устройствами ввода и CPU. Прерывания. DMA.
3. Виртуальная память. MMU: TLB, каталог страниц (page table). Биты: readable, writeable, executable, present, dirty, copy-on-write. IOMMU. Память процессов. Общая память. Пулы памяти со специальными требованиями.
4. "География" адресного пространства процесса. Структуры ядра описывающие процесс с MMU: работа с физической памятью, VMA. Системные вызовы: brk, sbrk, mmap. Общая память: mmaping файлов и shm_open. Реализация malloc.
5. Процессы и треды. init, родители, дети, зомби. Треды, группы тредов, процессы, группы процессов, сессии. Системные вызовы fork, clone, exec, wait. Интерфейс bash. Реализация переключения контекстов процессов: структуры данных ядра, состояния процессов, различные методы реализации CPS-преобразования.
6. Файловые дескрипторы и пайпы. Системные вызовы open, read, write, close. Структуры данных ядра: таблица файловых дескрипторов, файловые объекты POSIX, флаг CLOEXEC. Системные вызовы dup2, fcntl, flock. Структуры данных ядра для реализации пайпов, семафоров, блокировок на файлы.
7. Драйвера устройств в пространстве ядра. Прерывания. Монолитная обработка прерываний. Hi/Lo прерывания. Polling. Драйвера как контексты исполнения и их отличия от процессов. Реализация драйверов: структуры данных ядра, различные методы реализации CPS-преобразования.
8. Файловые системы. Структура данных inode. VFS. Структуры данных: FSObject, Namespace. Path resolution. Операции над неймспейсами: mount, bind mount, move mount, chroot, pivot_root. Linux FUSE. ФС как функция inoden0 → inode. Структуры ядра: файловый дескриптор (на устройство, файл, директорию), различные кешы. mmaping файлов.
9. Пользователи и права. Модели прав доступа к объектам: дискретная и ролевая. Права на объекты файловой системы. Пользователи и группы с точки зрения ядра. Пользователи и группы с точки зрения пространства пользователя. Системные вызовы setuid, setgid и товарищи. setuid bit. PAM. /etc/passwd, /etc/shadow, /etc/group. Capabilities.
10. Стандартные сигналы. Маски. Правила доставки. Реалтаймовые сигналы. Маски и очереди. Правила доставки. Системные вызовы kill, sigaction. Прерывание сигналами: кода программы, обработчиков сигналов, системных вызовов. Реентрабельность и безопасные системные вызовы. Сигналы и треды. Семантика сигналов: TERM, KILL, STOP, CONT, CHLD, PIPE, ILL/FPE, SEGV, BUS.
11. Мультиплексирование ввода-вывода. O_NONBLOCK. Edge и level triggered события. Преобразование асинхронного ввода-вывода в синхронный CPS-преобразованием. Структуры данных пространства ядра для реализации мультиплексора файловых дескрипторов. Системные вызовы select, poll, epoll. Управление скоростью передачи данных через файловые дескрипторы.
12. Синхронизация. Спинлоки. Ядерные семафоры. Блокировки и лизинги на файлы. Структуры данных пространства ядра для реализации блокировок и лизингов. Системные вызовы: flock, fcntl.
13. Сетевой стек. PPP, Ethernet. IP. TCP, UDP, SCTP. BSD sockets: API, Stream-сокеты, Datagram-сокеты, RAW-сокеты, файловый объект для accept-сокета.
14. Терминалы и управление заданиями в POSIX. Терминалы, псевдотерминалы и режимы их работы. Группы процессов и сессии. Foreground и background группы. Сигналы: INT, HUP, TSTP, TTIN, TTOU, WINCH. Демоны и демонизация.
15. Загрузка: BIOS → MBR (DOS Label), DOS/Windows boot, GRUB. initrd. Инициализация системы: последовательная, учитывая зависимости, resource/socket activation, lazy activation. Стандартные init системы: System V init (не забыв про runlevel), Upstart, OpenRC, systemd. Стандартные демоны: init, syslog, klog, cron, at, ssh. Стандартные файлы /etc: fstab, mtab, sysctl.conf, motd, issue, nologin.

16. Запуск программ, динамическая линковка и загрузка. Ehes magic и интерпретаторы. Релокация кода: релокационные дырки, кеширование релокаций, GOT, PIC. Объектные, исполняемые и библиотечные файлы. Формат ELF. ld-linux и его x86₃₂шные ужасы.

Второй семестр

1. Аллокация памяти. Алгоритмы и их свойства: linear list, log, SLAB, Buddy, Hoard. Выделение выравненной памяти. Трюки при реализации: mmap выравненной памяти. per-thread storage. Аллокация дискового пространства, device mapper. Логическое управление томами. RAID. Шифрование данных. Примеры: mdadm, LVM, cryptsetup/luks.
2. Файловые системы. Статические, динамические на блоках и экстендах. Используемые структуры данных. Примеры: tmpfs, cpio, iso9660, squashfs, FAT, ext2, XFS. Журналирование: физическое, логическое. LogFS. Журналирование с жертвами. Примеры: ext3/4 (journal, ordered, writeback), raiser 3/4. Продвинутое возможности. Примеры: ZFS, btrfs.
3. Системы контроля версий: встроенные в файловую систему, централизованные, распределённые. Структуры данных. Примеры: ext3-cow, cvs, svn, git, mercurial.
4. Сетевые диски. Примеры: ndb, drbd. Сетевые файловые системы. Примеры: NFS, sshfs, 9P. Распределённые файловые системы. Примеры: Google GFS, git. Дисковые устройства и файловые системы на кластерах. Fencing. Примеры: Clustered LVM, Redhat GFS 1/2.
5. Планирование исполнения процессов. Постановка задачи. FIFO, RR, пропорциональное (точный алгоритм за $O(\log n)$, алгоритм ядра Linux). Планирование с deadline'ами. Нечестность и инверсия приоритетов. Обычные и реалтаймовые приоритеты POSIX и их планирование. Системные вызовы shed_getscheduler, shed_setscheduler. Планирование в реалтаймовых системах.
6. Планирование дискового ввода-вывода. Постановка задачи. Планирование для SSD: FIFO, write-anticipatory. Планирование для HDD: двунаправленный лифтовый планировщик, однонаправленный лифтовый планировщик. Deadline. Anticipatory. Инверсии приоритетов. CFQ.
7. Планирование передачи пакетов. Классификация пакетов, классификация соединений, QOS. Планирования для сети: дырявое ведро (leaky bucket), токенизированное ведро (token bucket). Алгоритм поиска оптимальной скорости передачи. RR, WRR. HTB. SFQ, QFQ. HFSC. Самое узкое место и инверсия приоритетов. Полезная инверсия приоритетов.