1. Блочные устройства, основные параметры производительности (скорость линейного чтения, IOPS) для HDD и SSD. Производительность ввода-вывода в зависимости от доли случайных доступов.
2. POSIX FS interface (open, pread, pwrite, close, stat, link, unlink, symlink). Race scenarios в работе с ФС (lstat + readlink, access + open, chdir/chroot + open).
3. Объект ФС отделён от имени: hardlinks, безымянные файлы.
4. «Всё есть файл», особые файлы (char & block devices, sockets, pipes, etc.). Полиморфизм операций open/read/write на примере регулярных файлов и сокетов.
5. Иерархия каталогов, FHS, точки монтирования, bind-mounts.
6. UNIX-модель прав доступа к файлам. Различие между правами доступа к имени и к файловому дескриптору.
7. Синхронный и асинхронный ввод-вывод.
8. Очереди IO: pipelining, head-of-line blocking, multiplexing.
9. Virtual memory: зачем нужна виртуализация адресного пространства и как она реализуется в linux/x86. Shared memory, copy-on-write, demand paging.
10. Memory-mapped files: реализация, интерфейс mmap/munmap, обработка ошибок ввода-вывода.
11. Обмен данными между машинами разных архитектур. Endianness и memory layout структур.
12. Устройство ext2.
13. Обеспечение обратной совместимости ФС на примере ext2 и QCOW2: compat, ro-compat, incompat и compat-discard features. HTree-каталоги в ext3.
14. Журналирование изменений ФС. Транзакции. Журналирование логических изменений vs. журналирование физических изменений.
15. Гарантии консистентности данных, предоставляемые XFS.
16. Идемпотентность операций. Применение к журналированию и к проектированию сетевых протоколов.
17. Бинарные деревья поиска, сбалансированные деревья. 2-3-деревья и красно-чёрные деревья.
18. B-деревья и B^{link}-деревья. Вставка и удаление элементов из B-дерева.
19. Слияние B-деревьев. LSM-деревья и фильтры Блума.
20. Заполненность корзин хеш-таблиц с одной и с несколькими независимыми хеш-функциями. Power of 2 choices, применение к балансировке нагрузки и улучшению tail latency.
21. Способы проверки целостности данных ФС. CRC и криптографические хеши. Деревья Меркле. Примеры использования в ext4 и ZFS.
22. RAID-массивы. RAID 0, 1, 5, 6. RAID write hole.
23. Copy-on-write файловые системы. Устройство ZFS.
24. Путь записываемых данных: приложение -> libc -> pagecache -> block layer -> block device. Функции, выполняемые pagecache и block layer.
25. Протокол NFS и fallacies of networked computing.

Для допуска к зачёту надо знать интерфейс POSIX для работы с ФС, уметь открыть файл и отобразить его в память. В частности, надо знать про O_TRUNC, O_EXCL и прочие флаги для open(). Следующие man pages обязательны к прочтению:

- man 2 open,
- man 2 mmap.

Список рекомендуемой литературы:

1. M.K. McKusick: The design and implementation of the FreeBSD.
2. M. Kerrisk: The Linux programming interface.
3. Intel 64 and IA-32 Architecture Developer's Manual, volume 3A: System Programming Guide.
4. U. Drepper: What every programmer should know about memory. https://people.freebsd.org/~lstewart/articles/cpumemory.pdf
5. Proceedings of the USENIX conference on file systems and storage:
    a. W. Jannen: BetrFS: a right optimised write-optimised file system. https://www.usenix.org/system/files/conference/fast15/fast15-paper-jannen_william.pdf
    b. Ch. Lee: F2FS: a new file system for flash storage. https://www.usenix.org/system/files/conference/fast15/fast15-paper-lee.pdf
    c. R. Kesavan: Algorithms and data structures for efficient free space reclamation in WAFL. https://www.usenix.org/system/files/conference/fast17/fast17-kesavan.pdf
    d. H. Kumar: High-performance metadata integrity protection in the WAFL copy-on-write file system. https://www.usenix.org/system/files/conference/fast17/fast17-kumar.pdf
    e. A. Ganesan: Redundancy does not imply fault tolerance. https://www.usenix.org/system/files/conference/fast17/fast17-ganesan.pdf
    f. R. Alagappan: Protocol-aware recovery for consensus-based storage. https://www.usenix.org/system/files/conference/fast18/fast18-alagappan.pdf
6. Linux Weekly News articles and reviews:
    a. Mount namespaces and shared subtrees. https://lwn.net/Articles/689856/
    b. Kernel support for miscellaneous binary formats. https://lwn.net/Articles/679310/
    c. Handling writeback errors. https://lwn.net/Articles/718734/
    d. Ext4 and data loss. https://lwn.net/Articles/322823/
    e. Ensuring data reaches disk. https://lwn.net/Articles/457667/
    f. A journal for MD/RAID5. https://lwn.net/Articles/665299/
    g. LSFMM: https://lwn.net/Articles/lsfmm2016/, https://lwn.net/Articles/lsfmm2017/, https://lwn.net/Articles/lsfmm2018/
7. A. Langley: The QUIC transport protocol. https://research.google.com/pubs/archive/46403.pdf
8. D. Bernstein: HTTP/2 explained. https://legacy.gitbook.com/book/bagder/http2-explained/details
9. Ext2 on-disk format:
    a. http://www.nongnu.org/ext2-doc
    b. https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout
    c. http://wiki.osdev.org/Ext2
10. R. Sedgewick: Algorithms. https://algs4.cs.princeton.edu/home/
11. LSM and B^e trees:
    a. P. O'Neil: The log-structured merge tree. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.44.2782&rep=rep1&type=pdf
    b. B. Stopford: Log-structured merge trees. http://www.benstopford.com/2015/02/14/log-structured-merge-trees/

    c. M. Bender: An introduction to B^e-trees and Write-Optimisation. https://www.usenix.org/system/files/login/articles/login_oct15_05_bender.pdf
12. M. Mitzenmacher: The power of two random choices: a survey of techniques and results. https://people.cs.umass.edu/~ramesh/Site/PUBLICATIONS_files/MRS01.pdf
13. J. Dean: The tail at scale. http://cseweb.ucsd.edu/~gmporter/classes/fa17/cse124/post/schedule/p74-dean.pdf
14. M. Rosenblum, J. Ousterhout: The design and implementation of a log-sturctured file system. https://people.eecs.berkeley.edu/~brewer/cs262/LFS.pdf
15. J. Bonwick: The zettabyte file system. http://www.mckusick.com/bookrefs/zfs_overview.pdf
16. D. Hitz: File system design for a new NFS appliance. https://rcs.uwaterloo.ca/papers/wafl.pdf
17. ScyllaDB userspace IO scheduler:
    a. https://www.scylladb.com/2016/04/14/io-scheduler-1/
    b. https://www.scylladb.com/2016/04/29/io-scheduler-2/
    c. https://www.scylladb.com/2018/04/19/scylla-i-o-scheduler-3/
18. V. Jacobson: Controlling queue delay. https://queue.acm.org/detail.cfm?id=2209336
19. H. Zhou: Overload control for scaling WeChat microservices. https://www.cs.columbia.edu/~ruigu/papers/socc18-final100.pdf
20. H. Howard: Distributed consensus revised. https://pmc.acronis.com/browse/TTASK-31353
21. L. Lamport: Specifying systems. https://lamport.azurewebsites.net/tla/book-02-08-08.pdf