



Операционные системы

Лекция 7 Файловые системы



Организация памяти





Числа, которые должен знать каждый программист

| Операция | Задержка | Масштаб |
|--|----------------|--------------------------------|
| Обращение к кэшу L1 | 0.5 нс | 1 секунда |
| Ошибка при предсказании условного перехода | 5 нс | 10 сек. |
| Обращение к кэшу L2 | 7 нс | 14 сек. |
| Открытие/закрытие мьютекса | 25 нс | 50 сек. |
| Обращение к главной памяти | 100 нс | 3 мин. 20 сек. |
| Сжатие 1 Кб быстрым алгоритмом | 3,000 нс | 1 ч. 40 м. |
| Пересылка 1Кб по сети со скоростью 1 Гб/с | 10,000 нс | 5 ч. 33 м. 20 сек. |
| Чтение 1 Мб последовательно из главной памяти | 250,000 нс | 5 дней 18 ч. 53 мин. 20 сек. |
| Передача сообщения туда/обратно в одном дата-центре | 500,000 нс | 11 дней 13 ч. 46 мин. 40 сек. |
| Произвольный доступ к жёсткому диску | 10,000,000 нс | 231 дней 11 ч. 33 мин. 20 сек. |
| Чтение 1 Мб последовательно с жёсткого диска | 20,000,000 нс | 462 дней 23 ч. 6 мин. 40 сек. |
| Передача пакета из Калифорнии в Нидерланды и обратно | 150,000,000 нс | 3472 дней 5 ч. 20 мин. |



Основные определения

Файл — поименованная совокупность данных

Файловая система — часть ОС, отвечающая за работу с файлами.

ФУНКЦИИ:

1. Создание, удаление, модификация файлов
2. Разделение файлов друг от друга, поддержание целостности
3. Совместная работа нескольких процессов с файлами
4. Изменение структуры файла
5. Восстановление после стирания
6. Обеспечение разных методов доступа и режима секретности
7. Обращение к файлу по символическому имени
8. Дружественный интерфейс

Физическая запись или **блок** — единица информации, которую можно считать с носителя или записать на него



Организация файлов

Последовательная – записи в файле располагаются в физическом порядке. Магнитные ленты, перфоленты, перфокарты. Возможно и на дисках

Индексно-последовательная – записи в файле располагаются в логическом порядке в соответствии со значением ключей, содержащихся в каждой записи. Имеется специальный файл – индексный, где расположены адреса записей, упорядоченные по значению ключа. Диски.

Прямая – доступ к записям осуществляется прямо по их адресам. ЗУ прямого доступа.

Библиотечная – файл представляется суммой последовательных подфайлов.



Распределение памяти

Связное — каждому файлу выделяется непрерывная область памяти.

Достоинства: высокая скорость доступа, простая директория.

Недостатки: файл можно записать на диск только при наличии подходящей по размеру непрерывной области. Необходимо использовать трудоемкую операцию «сжатие» (**sque**, а не defrag)

Несвязное — носитель разбивается на области (сектора). Файл представляется Последовательностью секторов, может быть и не связанных.

Достоинства: не требует операции сжатия.

Недостатки: сложная директория, необходимость в операции **defrag**.



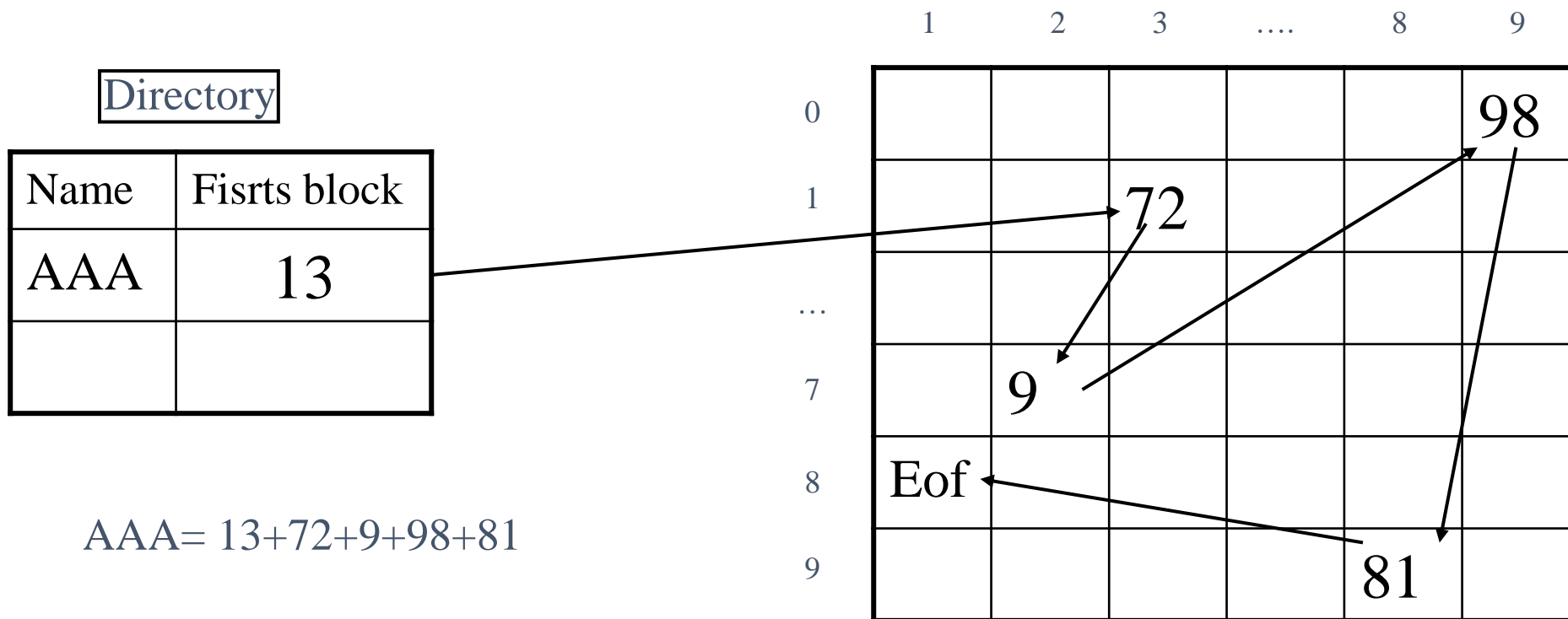
* Указатели на предыдущий и последующий секторы

Долго собирать информацию о файле!



Поблочное распределение

Таблица поблочного распределения – FAT (File Allocation Table)

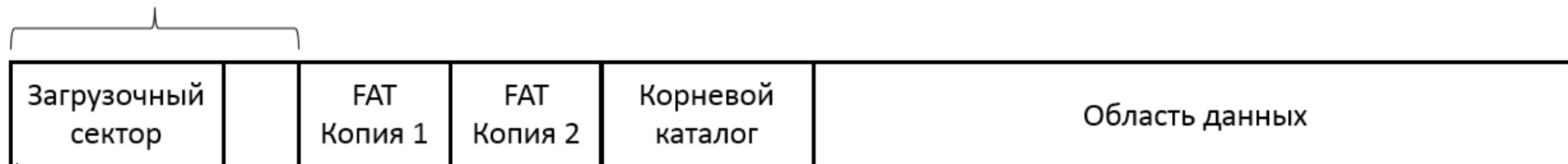


Достоинства: быстрее цепочки блоков.

Недостатки: фиксированное число файлов, сложность вставки блоков.



Зарезервированная область

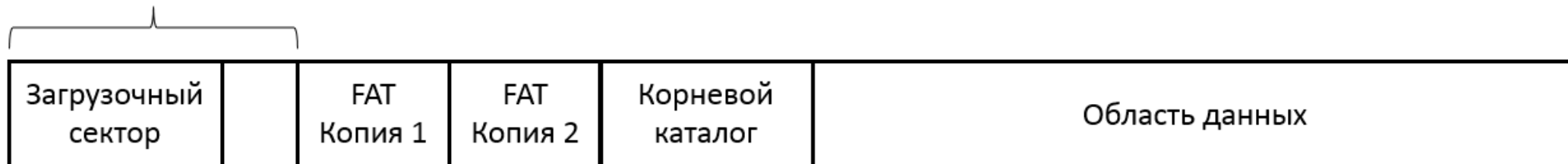


| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|------|--------------------------------------|--|---|--------------------------|---|--------------|------------------------------------|---|---|---|---|---------------------------|---|--------------------------------|---|---|
| 0x0 | Команда перехода к загрузочному коду | | | Имя OEM | | | | | | | | Количество байт в секторе | | Количество секторов в кластере | Размер зарезервированной области в секторах | |
| 0x10 | Количество копий FAT | Максимальное количество файлов в корневом каталоге | | Количество секторов в ФС | | Тип носителя | Размер каждой копии FAT в секторах | | - | - | - | - | - | - | - | - |



FAT16

Зарезервированная
область



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|------------------------|---|--|---|--------------------|---|----------------------------------|---|---------------------------------|---|---------------------------|---------------|--------------|-------------------------------|----------------------------|---|
| 0 | Имя файла | | | | | | | | Тип файла | | | Атрибу ты* | Резерв | 10ms создан ия файла | Время создания файла | |
| 10 | Дата создания файла | | Дата последнего обращения к файлу | | Не используется | | Время последнего изменения | | Дата последнего изменения | | Номер первого кластера | | Размер файла | | | |

Атрибуты

| Бит | 7 | 6 | 5 | 6 | 3 | 2 | 1 | 0 |
|----------|---|---|-------|----------------|---------------|--------------------|-----------------|------------------|
| Значение | 0 | 0 | Архив | Директор ия | Метка тома | Системн ый файл | Скрытый файл | Только чтение |



Пример корневого каталога

```
00005820 31 20 20 20 20 20 20 20 54 58 54 20 00 9C 17 9A 1 TXT ....
00005830 75 53 75 53 00 00 17 9A 75 53 03 00 36 01 00 00 uSuS....uS..6...
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|------------------------|---|--|---|--------------------|---|----------------------------------|---|---------------------------------|---|---------------------------|----------------|--------------|-------------------------------|----------------------------|---|
| 0 | Имя файла | | | | | | | | Тип файла | | | Атрибу- ты* | Резерв | 10ms создан ия файла | Время создания файла | |
| 10 | Дата создания файла | | Дата последнего обращения к файлу | | Не используется | | Время последнего изменения | | Дата последнего изменения | | Номер первого кластера | | Размер файла | | | |

Атрибуты

| Бит | 7 | 6 | 5 | 6 | 3 | 2 | 1 | 0 |
|----------|---|---|-------|----------------|---------------|--------------------|-----------------|------------------|
| Значение | 0 | 0 | Архив | Директор ия | Метка тома | Системн ый файл | Скрытый файл | Только чтение |

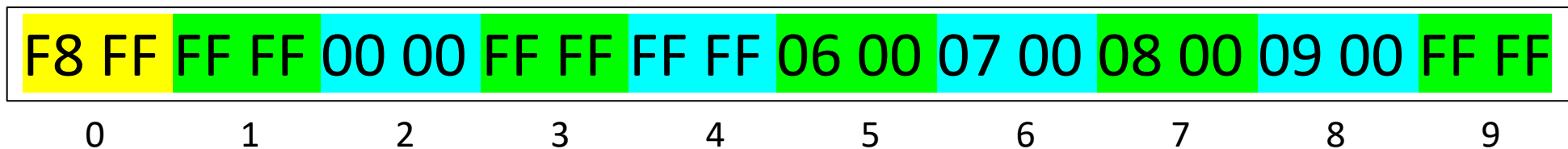


FAT таблица

| | | |
|----------|---|-------|
| 00000800 | F8 FF FF FF 00 00 FF FF FF FF 06 00 07 00 08 00 | |
| 00000810 | 09 00 FF FF 00 00 00 00 00 00 00 00 00 00 00 | |

0xF8 – дескриптор, определяющий FAT таблицу

FAT16 -> 2 байта





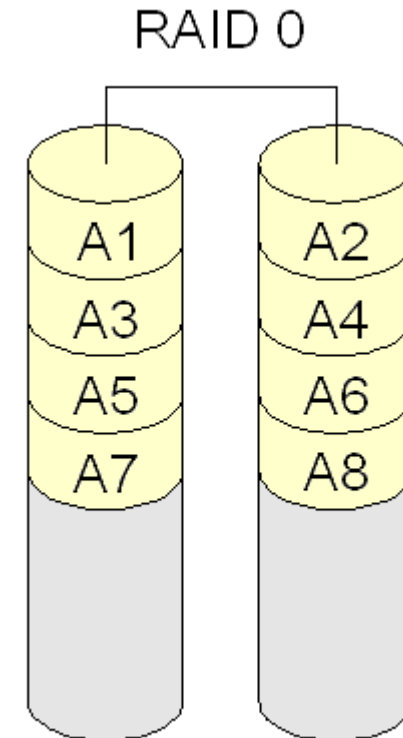
RAID 0

RAID 0 («*Striping*») — дисковый массив из двух или более дисков с отсутствием избыточности. Информация разбивается на блоки данных (A_i) и записывается на оба/несколько дисков одновременно.

(+): За счёт этого существенно повышается производительность (от количества дисков зависит кратность увеличения производительности).

(+): RAID 0 может быть реализован как программно, так и аппаратно.

(-): Страдает надёжность всего массива (при выходе из строя любого из входящих в RAID 0 винчестеров полностью и безвозвратно пропадает вся информация). Надёжность массива RAID 0 ниже надёжности любого из дисков, т.к. она равна произведению вероятностей безотказной работы составляющих его дисков, каждая из которых меньше единицы.





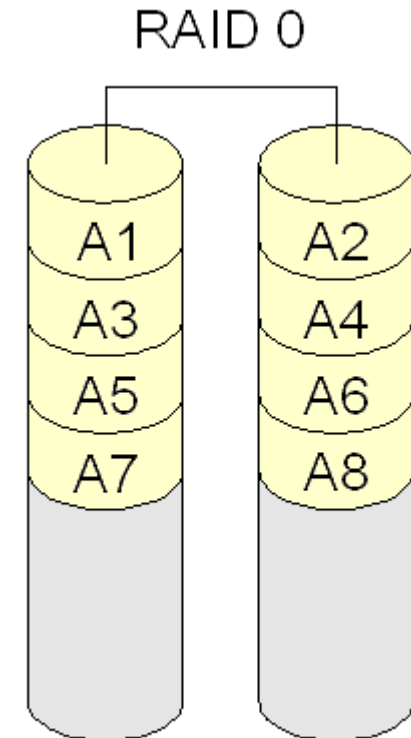
RAID 0

RAID 0 («*Striping*») — дисковый массив из двух или более дисков с отсутствием избыточности. Информация разбивается на блоки данных (A_i) и записывается на оба/несколько дисков одновременно.

(+): За счёт этого существенно повышается производительность (от количества дисков зависит кратность увеличения производительности).

(+): RAID 0 может быть реализован как программно, так и аппаратно.

(-): Страдает надёжность всего массива (при выходе из строя любого из входящих в RAID 0 винчестеров полностью и безвозвратно пропадает вся информация). Надёжность массива RAID 0 ниже надёжности любого из дисков, т.к. она равна произведению вероятностей безотказной работы составляющих его дисков, каждая из которых меньше единицы.





RAID 1

RAID 1 (Mirroring — «зеркалирование»).

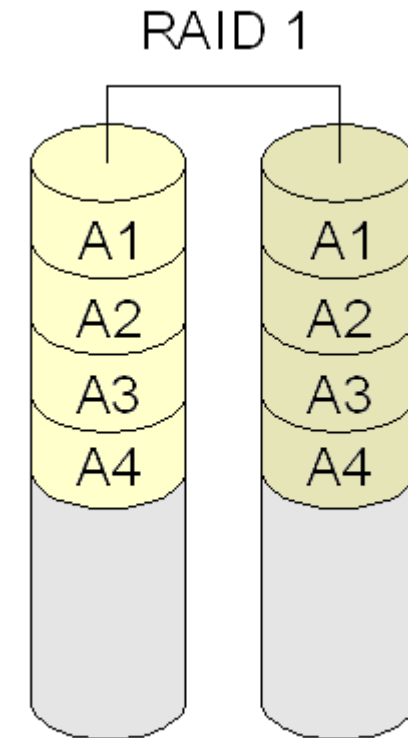
(+): Обеспечивает приемлемую скорость записи и выигрыш по скорости чтения при распараллеливании запросов.

(+): Имеет высокую надёжность — работает до тех пор, пока функционирует хотя бы один диск в массиве.

(-): Недостаток заключается в том, что приходится выплачивать стоимость двух жёстких дисков, получая полезный объем одного.

Вероятность выхода из строя сразу двух дисков ниже. Достоинство такого подхода — поддержание постоянной надёжности.

Зеркало на многих дисках — **RAID 1+0**. При использовании такого уровня зеркальные пары дисков выстраиваются в «цепочку», поэтому объём полученного тома может превосходить ёмкость одного жёсткого диска.



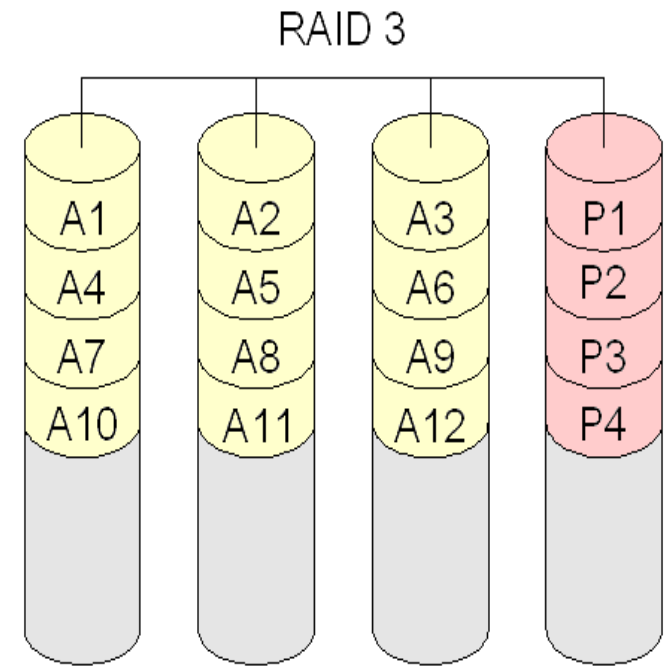


RAID 3

Структура RAID 3: в массиве из n дисков данные разбиваются на блоки размером 1 байт и распределяются по $n - 1$ дискам, а еще один диск используется для хранения блоков четности. В RAID 2 для этой цели стояло $n - 1$ дисков, но большая часть информации на этих дисках использовалась только для коррекции ошибок на лету, а для простого восстановления в случае поломки диска достаточно меньшего ее количества, хватает и одного выделенного винчестера. Соответственно, отличия RAID 3 от RAID 2 очевидны: невозможность коррекции ошибок на лету и меньшая избыточность.

(+): скорость чтения и записи данных высока, для создания массива нужно $\min 3$ диска.

(-): массив этого типа хорош только для однозадачной работы с большими файлами. Массив не получил распространения.





Приблизительные дальнейшие планы

9 декабря – прием лабораторных работ, БДЗ (с 9:00 до 11:50)

21 декабря – прием лабораторных работ, БДЗ (с 20:00 до 21:20)

23 декабря – зачет



Список вопросов к зачету

1. Типы и поколения ОС. Эволюция ОС и ее связь с развитием аппаратных ресурсов ЭВС.
2. Понятие об операционной системе, ее функциях и составе.
3. Архитектуры ОС. Функции ядра.
4. Концепция процесса. Состояния процесса.
5. Описание процесса в ОС. Операции над процессами.
6. Граф состояний процесса. Управление переходами.
7. Создание процесса в ОС linux
8. Организация межпроцессного взаимодействия в ОС linux
9. Процесс-зомби и процесс-сирота. Причины появления, влияние на производительность системы
10. Способы синхронизации процессов.
11. Процесс компиляции программ
12. Управление процессами. Переключение контекста, приоритеты. Основные стратегии.
13. Процессы и потоки.
14. Многопоточность и параллелизм. Сходства и различия.
15. Организация потоков в ОС. Механизмы синхронизации потоков. Data race и race condition
16. Управление потоками. Многопоточное программирование.
17. Прерывания ОС. Обработка прерываний.
18. Проблема тупиков в ОС. Необходимые условия возникновения тупиков.
19. Проблема тупиков в ОС. Алгоритм Деккера.
20. Проблема тупиков в ОС. Алгоритм Петерсона
21. Проблема тупиков в ОС. Алгоритм пекаря



Список вопросов к зачету

- 22. Организация оперативной памяти.
- 23. Виды памяти вычислительных систем.
- 24. Виртуальная память, назначение, основные проблемы. Методы организации.
- 25. Задачи менеджера виртуальной памяти.
- 26. Страничная организация виртуальной памяти.
- 27. Сегментная организация виртуальной памяти.
- 28. Сегментно-страничная организация виртуальной памяти.

- 29. Организация файлов. Функции файловой системы. Распределение внешней памяти.
- 30. Файловая система. Методы поблочного отображения.
- 31. Файловая система. FAT16
- 32. HPFS, NTFS и CDFS. Управление доступом к файлам.
- 33. Устройство НМД. Управление дисками. Критерии планирования.
- 34. Управление дисками. Основные стратегии управления.
- 35. Пути повышения производительности дисковых накопителей. RAID0, RAID1, RAID3.
- 36. Алгоритм шифрования RSA



Проведение зачета

Приблизительная дата зачета: **23 декабря**

На зачете нельзя ничем пользоваться

Билет – 2 вопроса из списка

Оценка за зачет суммируется с накопленной оценкой за семестр

0 - 49 – неуд.



50 - 69 – удовл.

70 - 85 – хорошо

86 - 100 – отлично

Алгоритм проведения зачета:

- 30 минут на подготовку – подробно расписываем билет на листочке
- Далее вызываем студента – устная защита билета

- 
- **1 балл** – возможно принести на зачет 1 лист А4 бумажной рукописной шпаргалки (с одной стороны)
 - **2 балла** – 5 минут пользования интернетом
 - **3 балла** – замена билета без понижения балла
 - **4 балла** – возможность ответа на один любой вопрос
 - **5 баллов** – автоматом оценка «Отлично»
- 



Спасибо за внимание!

Вопросы?

Если стесняемся, то можно сюда: @sergeybalabaev