# Ответьте на следующие вопросы

1. **Поясните понятие «файл».**

Ебать (ген. ebat’, “именновая совукопность взаимосвязанных данных, распозваваемая компьютером как единое целое”)

1. **Поясните понятие «файловая система».**

Способ хранения и упорядочивания на диске данных

1. **Перечислите типы файловых систем.**

* FAT – супер древняя
* FAT16 – очень древняя, максимальный размер файла – 2 ГБ
* FAT32 – древняя, но не настолько, как предыдущие, максимальный размер 1 файла – 4 ГБ
* NTFS – самая популярная, хотя по факту тоже древняя, максимальный размер файла – от 16 ГБ до 8 ПБ (зависит от размера кластера)

1. **Поясните понятие «поток данных».**

Поток данных (Stream) можно использовать в Node.JS в основном для работы с файлами. Если файл читается без использования потока (fs.ReadFileSync), то ожидается, пока файл не будет считан полностью. Потоки же могут считывать данные по частям и сразу начинать с ними работу, например, сразу начать отправлять порции файла клиенту.

Для отправки данных, считанных из потока, используется метод pipe().

1. **Поясните понятие «системные потоки данных».**

Наверно речь о stdin/stdout или потоках Readable/Writable, хуй знает

Если о stdin/stdout, то это стандартные унифицированные программные интерфейсы для записи/чтения данных

1. **Перечислите типы потоков данных, поддерживаемых Node.js.**

* Поток для чтения (Readable)
* Поток для записи (Writeable)
* Дуплексный поток (Duplex) – можно и читать, и записывать. Readable и Writable части работают независимо
* Трансформирующий поток (Transform) – похож на дуплексный, совмещает в себе запись и чтение; можно использовать для *трансформации* данных – шифрования, декомпресса данных, конвертирования из JSON в XML etc.

**Модули, предоставляюшие работу с потоками:**

* process.stdin, stdout, stderr – поток консоли
* fs.createReadStream() / createWriteStream() – поток для работы с файлами
* http.request() – отправляет chunk’и по потоку
* http.createServer() – данные обрабатываются порционно
* TCP-sockets – так написано в лекции
* zlib.createGzip() – сжимает данные в GZip и отправляет в поток
* net.connect() – соединение, сонованное на потоке