

Системы счисления и хранение данных, булевы операции, память, кодирование данных, сжатие данных

1. Различие винчестера от ssd.
2. Иметь представление о том, как представляются данные в виде двоичного кода.
3. Данные такие как звук изображение текст.
4. Как кодируются эти данные?
5. Иметь представление как хранятся числа в памяти целые и из плывущий точкой.
6. Знать, что такое сжатие данных. Знать разницу сжатия изображения, видео и аудио.

Манипуляция данных и железо, архитектура компьютеров, периферия

1. Как работает манипуляция данными?
2. Основы компьютерной архитектуры.
3. Как работает процессор?
4. Из чего состоит процессор?
5. Взаимодействие процессора с другими частями системы.
6. Понимание принципа исполнения программ, на уровне железа.
7. Знать, что такое конвейер.
8. Представление о том, что такое многопроцессорная архитектура.
9. Что такое машинный язык и зачем он нужен.
10. Понимание что такое логические операции и по-битовых сдвигов. (арифметика)
11. Как реализована и работает коммуникация с другими устройствами в системе. То есть коммуникация процессора памяти контроллеров и устройств, которые привязаны к ним. То есть модем монитор жесткий диск джойстик и т.п.
12. Что такое контроллер и зачем он нужен?
13. Что такое бит байт килобайт разница килобайта от килобита? И зачем это нужно и как это было придумано и так далее.

Операционные системы, компоненты, многопоточность

1. Знать предысторию появления ОС.
2. Знание архитектуры ОС, и его компонентов. И их взаимодействие.
3. Что такое shell? (Если вы можете ответить на этот вопрос, значит вы хорошо знаете эту тему.)
4. Что такое файловый менеджер?
5. Что такое менеджер устройств?
6. Что такое менеджер памяти?
7. Что такое виртуальная память?
8. Что такое пейджинг? (paging)
9. Понимание процесса запуска ОС.
10. В чём разница между user mode и kernel mode.
11. Концепция процесса.
12. Понимание разницы между программой и процессом.
13. Как реализована многопоточность.
14. Что такое прерывание? То есть понимать шаги выполнения программы во время прерывания.
15. Приоритеты в многопоточности. То есть понимание как процесс с высоким приоритетом может работать быстрее чем другие.
16. Что такое семафоры и дедлоки?

Сети и интернет

1. Что такое сети и зачем они нужны? То есть что такое протоколы, объединение сети, и понятие интернета и его архитектуры.
2. Как работает адресация?
3. Что такое HTTP?
4. Разница между URL и URI?
5. Что такое XML и HTML?
6. Что такое консорциум?
7. Что такое w3c и на чем был основан?
8. Основная разница между маршрутизатором и (switch statement) и (Сетевой концентратор)
9. Что такое mac адрес?
10. Что такое маска подсети?
11. Что такое понятие клиент-серверная архитектура?
12. Что такое pita by архитектура?
13. Что такое SGA и почему он больше не используется?
14. Что такое домены и его уровни?
15. Что такое хостинг?
16. Разница между OSI и TCP/IP.
17. Что такое протокол? И где он находится? И зачем он нужен?
18. Уровни протоколов.
19. Понятие пакетов.
20. Понятия порта.
21. Что такое проброс портов. И зачем это надо делать?

Сетевая безопасность

1. Виды атак в сети.
2. Иметь представление между трояном и червем.
3. Что такое D-DOS атака.
4. Зачем нужен firewall?
5. Что такое прокси сервер и зачем он нужен? Как его использовать?
6. Что такое VPN.
7. Виды шифрования. Что такое https и ssl, как используются ключи? И общее понимание шифрования в сети.

Алгоритмы

1. Понимать, что такое синтаксический анализ.
2. Понимать, что такое: сортировки, поиски сжатия, нахождение кратчайших путей, работа с разными структурами данных (то есть графами деревьями и т. д), фибоначчи, динамическое программирование

Языки программирования

1. Что такое языки программирования?
2. С чего началось появление языков программирования?
3. Что такое assembly? Как он возник?

4. Как появились трансляторы компиляторы и интерпретаторы?
5. Понимание разницы между этими программами. Знание парадигм программирования. А именно под множество императивный и декларативный. (Что такое что такое ООП, процедурное программирование, обобщённое программирование?)
6. Понимать, что в себе должен содержать завершений язык по Алану Тьюрингу.
7. Что такое структуры данных и зачем они нужны?
8. Понятие переменные понятие типа данных.
9. Что такое константы и литералы?
10. Что такое операторы и операнды?
11. Что такое control flow или инструкций по поведению кода?
12. Что такое комментарии, понятия процедуры?
13. Принцип работы области видимости, областей видимости.
14. Функций параметры функций. Способы передачи параметров в функции.
15. Что такое лексический анализ и синтаксическое дерево?
16. Что такое парсер и генератор кода?
17. Что такое токен?
18. Понимать, как работает рекурсия.
19. Что такое класс и объект?
20. Что такое instance(экземпляр)?
21. Что такое конструктор?
22. Три принципа ООП(наследование, инкапсуляция, полиморфизм).
23. Что такое композиция, и когда она используется?
24. Принцип многопоточного программирования.
25. Разница между процессом и потоком.

Разработка ПО

1. Имплементация алгоритмов, и структуры данных.
2. Что такое прототипизация модальность, и зачем это нужно?
3. Понимать зачем нужно тестирование и тесты.
4. Уметь продумывать дизайн и архитектуру перед тем как писать код.
5. Уметь ставить требования софта. Чтобы не придумывать их на ходу.
6. Что такое UML диаграммы? И как их использовать?
7. Зачем нужны парадигмы программирования?

Структур данных

1. Понимать, что такое абстрактные данные, и как они работают.
2. Какие преимущества одной структуры данных над другой?
3. Как работает память?
4. Понимание концепцию указателей.
5. Понимать разницу между указателем, ссылкой, и значением.
6. Что такое арифметика указателей?
7. Разница между LVALUE RVALUE?
8. Что такое умные указатели, и зачем они нужны?
9. Что такое побитовые сдвиги?
10. Что такое прямой и обратный код?
11. Что такое big endian и зачем он нужен?
12. Как хранятся данные в памяти.
13. Как расположены биты в памяти? (Справа налево, или слева на право?)
14. В чём разница хранения integer и double?
15. Что такое стек и где он расположен? И как он работает?
16. Умение управлять стеком.