

Вопросы

1 Указатель на void. Стандартные функции обработки областей памяти.

Для чего используется. Примеры.

Особенности указателя на void. Примеры.

memchr, memmove, memcmp, memset. Особенности работы. Отличия.

Можно ли использовать указатели на void совместно с указателями на функции.

2 Функции динамического выделения памяти

Что есть общего, по каждой функции: malloc, calloc, free, realloc (отдельно).

Типичные ошибки. Выделение 0 байт памяти, необходимость приведения типов.

3 Выделение памяти по динамический массив, типичные ошибки при работе с динамической памятью

Два способа выделения памяти под динамический массив.

Классификацию основных ошибок, примеры.

Подходы к обработке ситуации отсутствия динамической памяти.

(Как из функции вернуть динамически созданный массив, два способа)

.

4 Указатели на функции. Функция qsort

Для чего. Примеры.

Описать указатель на функцию, как присвоить значение, как вызвать.

Адресная арифметика.

5 Утилита make, назначение. Простой сценарий сборки

Для чего нужна, что на вход утилиты, идея в основе работы утилиты.

Разновидности утилиты, что собой представляет сценарий сборки(содержит переменные и правила, из каких составных частей, назначения у правил).

Пример простого сценария, как утилита make выполняет сборку.

(Все что нужно, сценарий сборки и время модификации, никакие БД не использует, сценарий отображает зависимости).

6 Утилита make, назначение, переменные, шаблонные правила

Для чего нужна, что на вход утилиты, идея в основе работы утилиты.

Пример использования переменных.

Неявные переменные, неявные правила, автоматические переменные, шаблонные правила. Везде примеры.

7 Утилита make, назначение, условные конструкции, анализ зависимостей

Для чего нужна, что на вход утилиты, идея в основе работы утилиты.
Переменные, зависящие от цели.
Вручную зависимости, любой C от всех H, или все на компилятор.

8 Динамические матрицы. Представление в виде одномерного, в виде указателя на строки

Графическая интерпретация
Выделение памяти, освобождение
Таблица для сравнения способов.

9 Динамические матрицы. Представление в виде одномерного, в виде указателя на строки

Графическая интерпретация
Выделение памяти, освобождение
Таблица для сравнения способов.

10 Динамические матрицы. Представление в виде одномерного, в виде указателя на строки

Графическая интерпретация
Выделение памяти, освобождение
Таблица для сравнения способов.

11 Динамические матрицы. Представление в виде одномерного, в виде указателя на строки

Графическая интерпретация
Выделение памяти, освобождение
Таблица для сравнения способов.

12 Динамические матрицы. Представление в виде одномерного, в виде указателя на строки

Графическая интерпретация
Выделение памяти, освобождение
Таблица для сравнения способов.

13 Динамические матрицы. Представление в виде одномерного, в виде указателя на строки

Графическая интерпретация
Выделение памяти, освобождение
Таблица для сравнения способов.

14 Чтение сложных объявлений

Примеры

15 Строки в динамической памяти. Функции Posix и расширения GNU, возвращающие такие строки

strdup, strndup, getline, sprintf

Что делают, как реализовать.

feature test macro

16 Особенности использования структур с полями указателями

Для структурного типа операция присваивания, побитовое копирование.

Пример, глубокое и поверхностное копирование.

Рекурсивное освобождение памяти. (Сначала память под внутренние поля)

17 Структуры переменного размера

Где встречается, примеры.

Flexible array member

Как обходились до появления, сравнить структуру с полем указателя и Flexible array member

18 Динамически расширяемый массив

Определение, тип, почему память надо выделять крупными блоками, добавление, освобождение.

19 Линейный односвязный список, добавление, удаление

20 Линейный односвязный список, вставка, удаление

21 Линейный односвязный список, обход

22 Двоичное дерево поиска, добавление

23 Двоичное дерево поиска, поиск

24 Двоичное дерево поиска, обход

25 Двоичное дерево поиска, удаление

26 Куча в программе на C, алгоритм работы malloc, пример реализации

Происхождение термина куча, для чего используется куча, преимущества и недостатки. Гарантии даются программисту, алгоритм работы функции malloc, пример ее реализации

27 Куча в программе на С, алгоритм работы free, пример реализации

Происхождение термина куча, для чего используется куча, преимущества и недостатки. Гарантии даются программисту, алгоритм работы функции free, пример ее реализации. Дефрагментация.

28 Куча в программе на С, проблемы выравнивания, пример реализации

Происхождение термина куча, для чего используется куча, преимущества и недостатки. Гарантии даются программисту, алгоритм работы функции free, пример ее реализации

29 VLA, alloca

30 Функции с переменным числом параметров

Идея реализации(которые кринж). Есть файл stdarg, есть соответствующие макросы. Как написать функцию, аналогичную printf.

31 Препроцессор. Общие понятия. Директивы include. Простые макросы. Предопределенные макросы

Что такое препроцессор, когда вступает в работы, его функции. Управляется директивами. На какие группы можно разделить. Какие правила справедливы. Чем использование двойных кавычек отличается от угловых скобок. Простые макросы, как обрабатываются препроцессором, для чего простые макросы, какие предопределенные макросы вам известны.

32 Препроцессор, макросы с параметрами

Что такое препроцессор, когда вступает в работы, его функции. Управляется директивами. Как макросы с параметрами обрабатываются препроцессорами. Макросы с переменным числом параметров. Почему внутри тела макросы нужна использовать круглые скобки. Подходы к написанию длинных макросов.

33 Препроцессор, директивы условной компиляции. error. pragma

Что такое препроцессор, когда вступает в работы, его функции. Управляется директивами. Для чего условной компиляции, какие бывают. Группа if, ifdef. Error, Pragma.

34 Препроцессор, операция # и ##

Что такое препроцессор, когда вступает в работы, его функции. Управляется директивами. Особенности работы при раскрытии макросов.

35 Встраиваемые функции

Ключевое слово inline. Почему появилось, сравнить с макросами, с обычными функциями. Способы борьбы undefined reference.

36 Библиотеки, статические библиотеки

Что такое библиотека. Какие функции выносятся, в какой форме распространяются. Виды, преимущества, недостатки. Как собрать статическую, как собрать приложение.

37 Библиотеки, динамические библиотеки, динамическая компоновка

Что такое библиотека. Какие функции выносятся, в какой форме распространяются. Виды, преимущества, недостатки. Как собрать, особенности в Linux и Windows. Как оформляется код приложения. Для линукса нет, для винды есть. Сравнить динамическую компоновку и динамическую загрузку

38 Библиотеки, динамические библиотеки, динамическая загрузка

В любой ОС есть `api`, с помощью которой динамическую библиотеку можно загрузить в память, как ее найти, как выгрузить. `-l -L`, особенности именования.

39 Библиотеки, динамические библиотеки, подходы к реализациям функций, которым требуется создавать буфер динамически

два способа. Внутри динамической библиотеки функция выделения, освобождение. Переключается на вызывающую сторону. Надо передавать размер.

40 Библиотеки, динамическая библиотека на C, приложение на Python Ctypes

41 Библиотеки, динамическая библиотека на C, приложение на Python, модуль расширения

42 Неопределенное поведение

Побочный эффект, с точки зрения стандарта. Выражение, анализ. Два разных результата получили. Точка следования, какие точки следования стандарт выделяет. как с этим явлением в программе бороться. Виды неопределенного поведения в программах на Си.

43 АТД, модуль, разновидности модулей, АО, стек целых чисел

Определение модуля, преимущества модульной организации, какие части модуля, требования, что язык Си предоставляет, что такое тип данных, что такое АТД. Реализация абстрактного объекта.

44 АТД, модуль, АТД, стек целых чисел

Здесь не про АО, а про АТД. (10 работу делали? Ну тогда на вопрос ответить можем)

45 Списки ядра Linux, идея, основные моменты использования

Что собой представляет список с точки зрения СД. Как достичь реализации, в чем отличие указателя на `void`, как создать, добавить, удалить, обойти. Сравнить списки ядра и обычные списки. Говорите "у меня вот".

46 Списки ядра Linux, идея, основные моменты реализации

Не говорите "у меня вот". Каким образом универсальность, обходы, что за особенность там есть. По списку движемся указателем `list head`, а нужно обработать данные, как из этого указателя получить указатель на данные. Продемонстрировать идею, какая идея используется. Макрос `containeroff, offsetoff`. Грозите пальчиком, так делать нельзя, используем `stddef.h`