

Список теоретических вопросов к экзамену

по дисциплине

«Программирование на Си»

Ломовской И. В.

Москва — 2021 — TS2112262002

1. Для чего используется указатель на void? Приведите примеры.
2. Каковы особенности использования указателя на void? Приведите примеры.
3. Функции для выделения и освобождения памяти malloc, calloc, free. Порядок работы и особенности использования этих функций.
4. Функция realloc. Особенности использования.
5. Общие «свойства» функций malloc, calloc, realloc.
6. Функция выделения памяти и явное приведение типа: за и против.
7. Особенности выделения 0 байт памяти.
8. Способы возвращения динамического массива из функции.
9. Типичные ошибки при работе с динамической памятью (классификация, примеры).
10. Подходы к обработке ситуации отсутствия свободной памяти при выделении.
11. Для чего используется указатель на функцию? Приведите примеры.
12. Указатель на функцию: описание, инициализация, вызов функции по указателю.
13. Функция qsort, примеры использования.
14. Особенности использования указателей на функцию.
15. Указатель на функцию и адресная арифметика.
16. Указателя на функцию и указатель на void.
17. Утилита make: назначение, входные данные, идея алгоритма работы.
18. Разновидности утилиты make.
19. Сценарий сборки проекта: название файла, структура сценария сборки.
20. Правила: составные части, особенности использования правил в зависимости от составных частей.
21. Особенности выполнения команд.

22. Простой сценарий сборки.
23. Алгоритм работы утилиты make на примере простого сценария сборки.
24. Ключи запуска утилиты make.
25. Использование переменных. Примеры использования.
26. Неявные правила и переменные.
27. Автоматические переменные и их использование.
28. Шаблонные правила. Примеры использования.
29. Условные конструкции в сценарии сборки. Примеры использования.
30. Переменные, зависящие от цели. Примеры использования.
31. Автоматическая генерация зависимостей.
32. Представление динамической матрицы с помощью одномерного массива. Преимущества и недостатки.
33. Представление динамической матрицы с помощью массива указателей на строки/столбцы. Преимущества и недостатки.
34. Объединенный подход для представления динамической матрицы (отдельное выделение памяти под массив указателей и массив данных). Преимущества и недостатки.
35. Объединенный подход для представления динамической матрицы (массив указателей и массив данных располагаются в одной области). Преимущества и недостатки.
36. Необходимо реализовать функцию, которая может обрабатывать как статические, так и динамические матрицы. Какими способами это можно сделать?
37. Умение читать сложные объявления и использовать это на практике.
38. Функции, возвращающие динамическую строку: strdup/strndup, getline, snprintf/asprintf.
39. Feature Test Macro.
40. Функции memchr, memmove, memcmp, memset.
41. Структуры с полями указателями и особенности их использования.
42. «Поверхностное» копирование vs «глубокое» копирование.
43. «Рекурсивное» освобождение памяти для структур с динамическими полями.
44. Структуры переменного размера. Приведите примеры.
45. Что такое «flexible array member»? Какие особенности использования есть у этих полей? Для чего они нужны? Приведите примеры.
46. Flexible array member до C99.
47. Flexible array member vs поле-указатель.
48. Дайте определение массива.

49. Дайте определение линейного односвязного списка.
50. Сравните массив и линейный односвязный список.
51. Динамически расширяемый массив:
52. Описание типа.
53. Добавление нового элемента.
54. Удаление элемента.
55. Особенности использования.
56. Почему при добавлении нового элемента память необходимо выделять блоками, а не под один элемент?
57. Линейный односвязный список:
58. Описание типа.
59. Добавление нового элемента в начало/конец списка.
60. Вставка элемента перед/после указанного.
61. Удаление элемента из списка.
62. Обход списка.
63. Удаление памяти из-под всего списка.
64. Возможные улучшения "классической" реализации.
65. Двоичное дерево поиска:
66. Описание типа.
67. Добавление элемента.
68. Поиск элемента (рекурсивный и нерекурсивный варианты).
69. Обход дерева.
70. Освобождение памяти из-под всего дерева.
71. Язык DOT, примеры использования. Утилита GraphViz.
72. Что такое область видимости имени?
73. Какие области видимости есть в языке Си? Приведите примеры.
74. Какие правила перекрытия областей видимости есть в языке Си? Приведите примеры.
75. Что такое блок?
76. Какие виды блоков есть в языке Си?
77. Что такое объявление? Приведите примеры.

78. Что такое определение? Приведите примеры.
79. Для чего нужны объявления?
80. Чем отличаются определения и объявления?
81. Что такое время жизни программного объекта?
82. Какие виды времени жизни есть у переменных?
83. Какие виды времени жизни есть у функций?
84. Как время жизни влияет на область памяти, в которой располагается программный объект?
85. Что такое связывание?
86. Какие виды связывания есть в языке Си?
87. Как связывание влияет на "свойства"объектного/исполняемого файла? Что это за "свойства"?
88. Какими характеристиками (область видимости, время жизни, связывание) обладает переменная в зависимости от места своего определения?
89. Какими характеристиками (область видимости, время жизни, связывание) обладает функция в зависимости от места своего определения?
90. Какие классы памяти есть в языке Си?
91. Для чего нужны классы памяти?
92. Какие классы памяти можно использовать с переменными? С функциями?
93. Сколько классов памяти может быть у переменной? У функции?
94. Какие классы памяти по умолчанию есть у переменной? У функции?
95. Расскажите о классе памяти auto.
96. Расскажите о классе памяти static.
97. Расскажите о классе памяти extern.
98. Расскажите о классе памяти register.
99. Для чего используется ключевое слово extern?
100. Особенности совместного использования ключевых слов static и extern.
101. Как описать автоматическую глобальную переменную?
102. Какая переменная называется глобальной?
103. Какая переменная называется локальной?
104. Каким значением по умолчанию инициализируются автоматические переменные?

105. Каким значением по умолчанию инициализируются переменные с глобальным временем жизни?
106. Какими недостатками есть у использования глобальных переменных?
107. Объектный файл, секции, таблица символов.
108. Что делает компоновщик?
109. Журналирование, подходы к реализации.
110. Процесс запуска программы («превращения в процесс»).
111. Абстрактное адресное пространство программы.
112. Опишите достоинства и недостатки локальных переменных.
113. Локальные переменные создаются в так называемой «автоматической памяти». Почему эта память так называется?
114. Для чего в программе используется аппаратный стек?
115. Что такое кадр стека?
116. Для чего в программе используется кадр стека? Приведите примеры.
117. Какие преимущества и недостатки есть у использования кадра стека?
118. Что такое соглашение о вызове?
119. Какое соглашение о вызове используется в языке Си? В чем оно заключается?
120. Что такое переполнение буфера? Чем оно опасно?
121. Почему нельзя из функции возвращать указатель на локальную переменную, определенную в этой функции?
122. Для чего в программе используется куча?
123. Происхождение термина «куча».
124. Свойства области памяти, которая выделяется динамически.
125. Как организована куча?
126. Алгоритм работы функции `malloc`.
127. Алгоритм работы функции `free`.
128. Какие гарантии относительно выделенного блока памяти даются программисту?
129. Что значит "освободить блок памяти" с точки зрения функции `free`?
130. Преимущества и недостатки использования динамической памяти.
131. Что такое фрагментация памяти?
132. Выравнивание блока памяти, выделенного динамически.

133. Что такое variable length array?
134. Чем отличается статический массив от variable length array?
135. Какую операцию языка Си пришлось реализовывать по-другому (не как для встроенных типов) специально для variable length array?
136. Особенности использования variable length array.
137. Справедлива ли для variable length array адресная арифметика?
138. Как вы думаете почему variable length array нельзя инициализировать?
139. Для чего используется variable length array? Приведите примеры.
140. В какой области и «кем» выделяется память под массив переменной длины?
141. Функция alloca.
142. alloca vs VLA.
143. Можно ли реализовать в языке Си функцию со следующим прототипом `int f(...)` ? Почему?
144. Покажите идею реализации функций с переменным числом параметров.
145. Почему для реализации функций с переменным числом параметров нужно использовать возможности стандартной библиотеки?
146. Опишите подход к реализации функций с переменным числом параметров с использованием стандартной библиотеки. Какой заголовочный файл стандартной библиотеки нужно использовать? Какие типы и макросы из этого файла вам понадобятся? Для чего?
147. Какая особенность языка Си упрощает реализацию функций (с точки зрения компилятора) с переменным числом параметров?
148. Почему при вызове `va_arg(argp, short int)` (или `va_arg(argp, float)`) выдается предупреждение?
149. Какая "опасность" существует при использовании функций с переменным числом параметров?
150. Как написать функцию, которая получает строку форматирования и переменное число параметров (как функция `printf`), и передает эти данные функции `printf`? (Подсказка: см. последний вариант реализации журналирования.)
151. Что делает препроцессор? В какой момент в процессе получения исполняемого файла вызывается препроцессор?
152. На какие группы можно разделить директивы препроцессора?
153. Какие правила справедливы для всех директив препроцессора?
154. Что такое простой макрос? Как такой макрос обрабатывается препроцессором? Приведите примеры.

155. Для чего используются простые макросы?
156. Что такое макрос с параметрами? Как такой макрос обрабатывается препроцессором? Приведите примеры.
157. Макросы с параметрами vs функции: преимущества и недостатки.
158. Макросы с переменным числом параметров. Приведите примеры.
159. Какими общими особенностями/свойствами обладают все макросы?
160. Объясните правила использования скобок внутри макросов. Приведите примеры.
161. Какие подходы к написанию "длинных" макросов вы знаете? Опишите их преимущества и недостатки. Приведите примеры.
162. Какие предопределенные макросы вы знаете? Для чего эти макросы могут использоваться?
163. Для чего используется условная компиляция? Приведите примеры.
164. Директива `#if` vs директива `#ifdef`.
165. Операция `#`. Примеры использования.
166. Операция `##`. Примеры использования.
167. Особенности использование операций
168. Директива `#error`. Примеры использования.
169. Директива `#pragma` (на примере `once` и `pack`). Примеры использования.
170. Ключевое слово `inline`.
171. Назовите основную причину, по которой ключевое слово `inline` было добавлено в язык Си.
172. Подходы к реализации ключевого слова `inline` компилятором. Проанализируйте их недостатки.
173. В чем разница между использованием `<>` и `" "` в директиве `include`?
174. Можно ли операцию `sizeof` использовать в директивах препроцессора? Почему?
175. Что такое библиотека?
176. Какие функции обычно выносят в библиотеку?
177. В каком виде распространяются библиотеки? Что обычно входит в их состав?
178. Какие виды библиотек вы знаете?
179. Преимущества и недостатки, которые есть у статических/динамических библиотек.
180. Как собрать статическую библиотеку?
181. Нужно ли "оформлять" каким-то специальным образом функции, которые входят в состав статической библиотеки?

182. Как собрать приложение, которое использует статическую библиотеку?
183. Нужно ли "оформлять" каким-то специальным образом исходный код приложения, которое использует статическую библиотеку?
184. Как собрать динамическую библиотеку (Windows/Linux)?
185. Нужно ли "оформлять" каким-то специальным образом функции, которые входят в состав динамической библиотеки (Windows/Linux)?
186. Какие способы компоновки приложения с динамической библиотекой вы знаете? Назовите их преимущества и недостатки.
187. Что такое динамическая компоновка?
188. Что такое динамическая загрузка (Windows/Linux)?
189. Нужно ли "оформлять" каким-то специальным образом исходный код приложения, которое использует динамическую библиотеку (Windows/Linux)?
190. Особенности реализации функций, использующих динамическое выделение памяти, в динамических библиотеках.
191. Ключи -I, -l, -L компилятора gcc.
192. Проблемы использования динамической библиотеки, реализованной на одном языке программирования, и приложения, реализованного на другом языке программирования.
193. Модуль ctypes. Загрузка библиотеки. Представление стандартных типов языка Си. Импорт функций из библиотеки. Проблемы, которые при этом возникают.
194. Написание модуля расширения для Python (основные шаги).
195. Что такое побочный эффект?
196. Какие выражения стандарт c99 относят к выражениям с побочным эффектом?
197. Почему порядок вычисления подвыражений в языке Си неопределен?
198. Порядок вычисления каких выражения в языке Си определен?
199. Что такое точка следования?
200. Какие точки следования выделяет стандарт c99?
201. Почему необходимо избегать выражений, которые дают разный результат в зависимости от порядка их вычисления?
202. Какие виды "неопределенного" поведения есть в языке Си?
203. Почему "неопределенное" поведение присутствует в языке Си?
204. Какой из видов "неопределенного" поведения является самым опасным? Чем он опасен?
205. Как бороться с неопределенным поведением?

206. Приведите примеры неопределенного поведения.
207. Приведите примеры поведения, зависящего от реализации.
208. Приведите примеры неспецифицированного поведения.
209. Что такое модуль?
210. Из каких частей состоит модуль? Какие требования предъявляются к этим частям?
211. Назовите преимущества модульной организации программы. Приведите примеры.
212. Какие виды модулей вы знаете? Приведите примеры.
213. Что такое тип данных?
214. Что такое абстрактный тип данных?
215. Какие требования выдвигаются к абстрактному типу данных?
216. Абстрактный объект vs абстрактный тип данных.
217. Средства реализации модулей в языке Си.
218. Что такое неполный тип данных в языке Си?
219. Приведите примеры описания неполного типа данных? (А кроме структур ;) ?)
220. Какие действия можно выполнять с неполным типом данных?
221. Для чего при реализации абстрактного типа данных используется неполный тип данных языка Си?
222. Проблемы реализации АТД на языке Си.
223. Есть ли в стандартной библиотеке языка Си примеры абстрактных типов данных?