

Теория:

1. Основные команды командной строки linux (bash):

cd, ls (опции -l и -a), pwd, cp, mv, rm (опция -r), mkdir, find(опция -name), du(опции -s и -h), программа top, текстовый редактор nano (или текстовый редактор vim), компилятор gcc (опции -o, -std=c99, -lm и -S). Перенаправление вывода >.

2. Управляющие конструкции. Ветвление и циклы: Оператор ветвления if-else. Использование логических операторов в условии оператора ветвления. Чем отличается тернарный оператор от if-else? Цикл while. Цикл for. Цикл do while. Операторы break и continue. Оператор switch. Оператор break в switch выражениях. Чем отличается switch от многократного вызова if else.

3. Операторы Арифметические операторы(+ - * / %). Что делает оператор деления если аргументы целочисленные и если аргументы – числа с плавающей точкой. Оператор присваивания (=). Оператор присваивания сложения и подобные ему (+= -= *= /= %=). Операторы инкремента и декремента (++ --). Префиксный и постфиксный инкремент/декремент, чем они отличаются. Операторы сравнения (== != > < >= <=). Что возвращают операторы сравнения? Логические операторы (! || &&). Побитовые операторы (~ & | ^ << >>). Тернарный оператор (? :). Оператор нахождения адреса (&). Оператор нахождения размера переменной (sizeof). Оператор разыменования (*). Оператор обращения к элементу массива ([]) и его связь с оператором разыменования. Оператор доступа к полю структуры (.). Оператор доступа к полю структуры через указатель на структуру (->). Приоритет операторов. Создание новых названий для типов с помощью ключевого слова typedef.

4. Переменные и базовые типы Переменные. Понятия объявления, определения, в чём различие между ними. Инициализация и присваивание, в чём различие между ними. Типы целочисленных переменных: char, short, int, long, long long и их unsigned-аналоги. Размеры этих типов на современных системах и диапазоны значений, которые могут принимать данные типы. Представление целочисленных переменных в памяти. Как хранятся в памяти отрицательные числа? Дополнительный код. Что такое тип size_t. Когда он используется? Типы чисел с плавающей точкой: float, double и long double. Представление этих типов в памяти. Стандарт IEEE 754. Размеры этих типов и количество значащих цифр, которые могут хранить переменные этих типов. Неявное приведение типов. Когда оно происходит? Явное приведение типов, как привести один тип в другой. Перечисляемый тип. Константы Квалификатор типа const. Разница между определением константы с помощью директивы #define и квалификатора const.

5. Основные функции стандартной библиотеки

Ввод и вывод в языке C. Функции printf и scanf из библиотеки stdio.h. Что принимает и возвращает функция printf? Что принимает и возвращает функция scanf? Математическая библиотека math.h. Функции sqrt, exp, sin, cos, tan, asin, acos, atan, atan2, fabs, floor, log, pow. Сравнение двух чисел с плавающей точкой с помощью функции fabs. Библиотека stdlib.h. Функции exit, malloc, calloc, realloc, free, rand, srand, abs. Основные функции для работы с файлами: fopen, fscanf, fprintf и fclose.

6. Массивы

Массивы. Элемент массива и индекс массива. Как хранятся массивы в памяти? Объявление и определение массивов. Инициализация массивов. Можно ли присваивать массив другому массиву с помощью оператора =? Как распечатать массив? Размер массивов. Как узнать размер массива? Как передаются массивы в функции. Array to pointer decay. Как вернуть массив из функции. Двумерный массив. Объявление, определение и инициализация двумерного массива. Как двумерный массив хранится в памяти? Как двумерный массив передаётся внутрь функции?

7. Символы и строки

Символы. Кодировка ASCII. Использование целочисленного типа char для хранения кодов символов. Чтение и запись символов (спецификатор %c). Строки в языке C. Символ завершения строки. Чтение и запись строк (спецификатор %s). Библиотека string.h. Функции strlen, strcpy, strncpy, strcmp, strcat, strchr, strstr. Функции sprintf и sscanf, использование этих функций для конвертации числа в строку и наоборот.

8. Функции

Функции. Аргументы функции. Возвращаемое значение функции. Объявления функции. Прототип функции. Определение функции. Как переменные базовых типов и структуры передаются в функции? Как массивы передаются в функции? Три типа передачи аргументов в функцию (по значению, через указатель, через указатель на константу). Передача одномерных и многомерных массивов в функции. Возврат из функции. Ключевое слово return. Ключевое слово void. Рекурсия. Алгоритмы вычисления факториала,

чисел Фибоначчи и бинарного возведения в степень с помощью рекурсии. Реализация вызова функций с помощью сегмента памяти стек. Адрес возврата. Стековый кадр. Переполнение стека при рекурсии.

9. Структуры

Структуры. Объявление структуры. Определение структуры. Инициализация структуры. Доступ к полю структуры. Размер структуры. Выравнивание полей структур. Указатели на структуры. Доступ к полю по указателю на структуру. Передача структур в функции и возврат их из функций.

10. Память и указатели

Шестнадцатеричная и восьмеричная системы счисления. Печать и считывания целочисленных переменных в восьмеричной и шестнадцатеричной системах с помощью спецификаторов `%o` и `%x`. Порядок байт. Little endian и big endian. Указатель. Объявление указателя. Инициализация указателя. Размер указателя на 64-х битных системах. Адресная арифметика. Операторы, применимые к указателям и что они делают: `++`, `-`, прибавление целого числа, вычитание двух указателей, разыменовывание и оператор взятия индекса (`[]`). Операции нахождения адреса (`&`) переменной и операция разыменования `*` указателя. Указатели разных типов, чем они различаются. Указатель `void*`. Константный указатель и указатель на константу. Указатели на указатели. Функции `memset`, `memcpy`, `memmove` из стандартной библиотеки.

11. Сегмент памяти Стек

Что такое сегменты памяти? Ошибка Segmentation Fault. Виртуальная память. Сегмент памяти стек(не путать с абстрактным типом данных – стек). Выделение и освобождение памяти в стеке. Размер стека. Переполнение стека. Как можно переполнить стек?

12. Сегмент памяти Куча. Динамическое выделение памяти

Сегмент памяти куча (heap – не путать со структурой данных – куча). Динамическое выделение и освобождение памяти в куче: `malloc`, `calloc`, `realloc` и `free`. Преимущества и недостатки кучи перед стеком. Организация массива в памяти. Организация связного списка в памяти. Хранение двумерного массива в виде одномерного массива. Динамический двумерный массив. Утечки памяти. Основы работы с `valgrind`.

13. Сегмент памяти Текст

Сегмент памяти text. Преобразование кода программы в код на языке ассемблера и в двоичный код. Указатели на функции. Объявление указателей на функции. Передача указателей на функции в другие функции. Стандартная функция `qsort` и передача в ней компаратора.

14. Сегмент памяти Данные

Сегменты памяти data и bss. Чем они различаются? Что такое глобальные переменные. Что такое статические переменные. Где хранятся глобальные и статические переменные? Когда и как инициализируются глобальные и статические переменные? Строковые литералы. Где хранятся строковые литералы?

15. Вычислительная сложность, $O(n)$ нотация

Что такое $O(n)$, $\Omega(n)$ и $\Theta(n)$ нотации, математическое определение и смысл. Машина Тьюринга (детерминистическая и недетерминистическая). Классы сложности задач: P , NP , $PSPACE$, $EXPTIME$ и $EXPSPACE$. Знать(или уметь выводить) сложности всех пройденных алгоритмов: алгоритм бинарного поиска в отсортированном массиве, алгоритмы сортировки (пузырьком, вставками, выбором, быстрая, сортировка слиянием, подсчётом, цифровая). Знать или уметь выводить сложности операций с пройденными структурами данных: поиск элемента по индексу в массиве и списке, вставка в массив(статический и динамический) и список; удаление из массива и списка; поиск по массиву и списку.

16. Сортировка

Парадигма разделяй и властвуй. Как возникает логарифмическая сложность в задачах, решаемых с помощью метода разделяй и властвуй. Алгоритм бинарного поиска в отсортированном массиве. Сортировки пузырьком, вставками, выбором, быстрая, слиянием, подсчётом, цифровая и *Bogosort*. Вычислительные сложности всех этих алгоритмов. Уметь писать сортировку выбором, быструю сортировку и сортировку подсчётом. Стандартная функция `qsort`. Функция компаратор `cmp`. Сортировка с помощью бинарного дерева поиска и с помощью бинарной кучи.

17. Абстрактные типы данных. Стек и очередь

Структуры данных и абстрактные типы данных. Абстрактный тип данных Стек. Операции `push` и `pop`. Реализация стека на основе динамического массива. Динамическое выделение памяти для стека. Стратегии перевыделения памяти: аддитивная и мультипликативная. Абстрактный тип данных Очередь. Реализация очереди на основе динамического массива. Реализация стека и очереди на основе связного списка. Алгоритмические сложности операций со стеком.

18. Связный список

Связный список. Узел связанного списка. Реализация связанного списка на языке C. Функции нахождения длины списка, вставки элемента в начало/конец, удаления элемента из начала/конца, поиска в списке, обращения списка. Алгоритмические сложности операций со списком. Двусвязный список.

19. Деревья

Определение графа. Определение дерева. Бинарное дерево. Бинарное дерево поиска(bst). Высота дерева. Представление бинарного дерева в языке C. Операции поиска, добавления и удаления элемента в бинарном дереве поиска. Вычислительные сложности этих операций. Сбалансированное дерево. Два распространённых приёма для балансировки бинарных деревьев поиска (AVL-деревья и красно-чёрные деревья). Определение AVL-дерева. Операции вращения. Операции поиска, добавления и удаления элемента в AVL-дерево. Вычислительные сложности этих операций. Абстрактные типы данных множество (set) и словарь (dictionary).

20. Хеш-таблица

Что такое хеш-функция? Свойства хорошей хеш-функции. Что такое хеш-таблица? Добавление/удаление элемента в хеш-таблицу. Поиск элемента в хеш-таблице. Вычислительная сложность операций с хеш-таблицей. Абстрактные типы данных множество и словарь на основе хеш-таблицы. Чем различие между реализацией множества и словаря на основе сбалансированного дерева и на основе хеш-таблицы.

21. Файлы и аргументы командной строки

Системные вызовы. Системные вызовы для работы с файлами в Linux (open, read, write и lseek). Буферизация. Работа с файлами в языке C - библиотека `stdio.h`. Структура `FILE`. Функции `fopen` и `fclose`. Режимы открытия файлов ("r" "w" "a" "r+" "w+" "a+"). Функции `fgetc`, `fputc`, `fscanf`, `fprintf`, `fgets`, `fputs`, `rewind` и `fseek`. Посимвольное чтение файла в цикле. Константа `EOF`. Чтение и запись в бинарные файлы. Функции `fread` и `fwrite`. Формат файла `.csv`. Форматы файлов для хранения изображений `.ppm` (текстовый и бинарный) и `.jpg`. Аргументы командной строки (`argc` и `argv`). Преобразование строки в число и обратно с помощью `sprintf` и `sscanf`.

22. Этапы компиляции

Что такое файл исходного кода и исполняемый файл. Этап компиляции: препроцессинг, компиляция и линковка. Директивы препроцессора `#include`, `#define`, `#ifdef`, `#else` и `#endif`. Компиляция программы с помощью `gcc`. Опции `gcc`: `-E`, `-c`, `-S`.

23. * Очередь с приоритетом. Двоичная куча

Хранение бинарных деревьев в массиве. Структура данных - двоичная куча. Операции добавления элемента в кучу. Операция удаления максимального элемента из кучи. Вычислительные сложности этих операций. Абстрактный тип данных - очередь с приоритетом. Реализация очереди с приоритетом с помощью двоичной кучи. Пирамидальная сортировка.

24. * Динамическое программирование

Суть динамического программирования. Задача о вычислении чисел Фибоначчи. Задача о поиске подмассива с максимальной суммой. Задача о наибольшей общей подпоследовательности. Задача о рюкзаке.

Материалы для подготовки:

1. Кормен, Лейзерсон, Ривест. Алгоритмы: построение и анализ.
lib.mipt.ru/book/16736/
2. Керниган Ритчи Язык программирования С
lib.mipt.ru/book/266005/
3. Лекторий МФТИ:
lectoriy.mipt.ru/course/ComputerTechnology-Informatics-14L#lectures
4. Фоксфорд
www.youtube.com/watch?v=J-7XmpSUEQE&list=PL66kLi3dt8A5sa_qBur8uxmtuuwuJQGS1&index=28
5. Язык С и структуры данных в Йелле.
www.cs.yale.edu/homes/aspnes/classes/223/notes.html

Дополнительные материалы по некоторым вопросам:

- Сегменты памяти:
 - youtube.com/watch?v=_8-ht2AKyH4 (англ.)
- Этапы компиляции:
 - youtube.com/watch?v=Je9FFb2zldk
 - youtube.com/watch?v=ylA55D4B4_M
- Стек вызовов:
 - youtube.com/watch?v=hU3ONaqZzD8