

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 `>`. Если у вас Windows, то советую использовать консоль `cmd`.

2. Основы C: переменные, операторы, функции из стандартной библиотеки

Переменные. Типы целочисленных переменных (`char`, `short`, `int`, `long`, `long long`, `unsigned char`, `unsigned short`, `unsigned int`, `unsigned long`, `unsigned long long`, `size_t`). Размеры этих типов. Представление целочисленных переменных в памяти. Типы чисел, для хранения вещественных чисел (`float` и `double`). Размеры этих типов и представление их в памяти. Операторы. Арифметические операторы (`+`, `-`, `*`, `/`, `%`). Операторы присваивания (`=`, `+=`, `-=`, `*=`, `/=`, `%=`). Операторы инкремента и декремента (`++`, `--`). 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 (`->`). Приоритеты операторов. Явное и неявное приведение типов. Ввод и вывод в языке C. Функции `printf` и `scanf` из библиотеки `stdio.h`. 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`.

3. Основы C: указатели, массивы и строки

Указатель. Объявление указателя. Размер указателя. Адресная арифметика. Операции нахождения адреса (`&`) и операция разыменования `*`. Указатели на указатели. Массивы. Объявление массивов. Инициализация массивов. Размер массивов. Связь между массивами и указателями. Статический двумерный массив. Строки. Связь между строками, массивами и указателями. Кодировка ASCII. Переменные `char` для хранения символов. Чтение и запись символов (`%c`) и строк (`%s`). Библиотека `string.h`. Функции `strlen`, `strcpy`, `strcmp`, `strcat`, `strstr`, `memset`, `memcpy`.

4. Основы C: управляющие конструкции, функции и структуры

Управляющие конструкции `if`, `else`, `for`, `while`, `do while`, `switch`. Операторы `break` и `continue`. Функции. Объявления функций. Прототип 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 `void`. Реализация вызова функций с помощью сегмента памяти стек. Адрес возврата. Стековый кадр. Рекурсия. Переполнение стека при рекурсии. Структуры. Инициализация структур. Доступ к полю структуры. Размер структуры. Указатели на структуры. Доступ к полю по указателю на структуру. Передача структур в функции и возврат их из функций. Оператор `typedef`.

5. Сегменты памяти

Что такое сегменты памяти. Сегмент памяти стек (не путать с абстрактным типом данных - стек). Выделение и освобождение памяти в стеке. Переполнение стека. 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. Ошибка `Segmentation Fault`. Сегмент памяти `text`. Преобразование кода программы в код на языке ассемблера и в двоичный код. Указатели на функции.

6. Динамическое выделение памяти

Динамическое выделение и освобождение памяти в куче: `malloc`, `calloc`, `realloc` и `free`. Указатель `void*`. Преобразование типа указателя. Организация массива в памяти. Организация связанного списка в памяти. Хранение двумерного массива в виде одномерного массива. Динамический двумерный массив. Утечки памяти. Основы работы с `valgrind`.

7. 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 выводиться) сложности всех пройденных алгоритмов: алгоритм бинарного поиска в отсортированном массиве, алгоритмы сортировки (пузырьком, вставками, выбором, быстрая, сортировка слиянием, подсчётом, цифровая). Знать или уметь выводиться 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.

8. **Стек и очередь**
Структуры данных и абстрактный 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.
9. **Связный список**
Связный список. Узел связного списка. Реализация связного 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.
10. **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.
11. **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.
12. **Передача аргументов в функцию**
Три типа передачи аргументов в функцию. Передача по значению. Передача через указатель. Передача через указатель на константу. Передача массивов, строк и двумерных массивов в функцию. Передача структур в функцию. Возвращение переменных из функции. Возвращение массива из функции.
13. **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.
14. **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.
15. *** 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.
16. *** 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