

Теория:

Модуль 1

1. Основные команды командной строки linux (bash)

`cd`, `ls` (опции `-l` и `-a`), `pwd`, `cp`, `mv`, `rm` (опция `-r`), `mkdir`, `find`(опция `-name`), программа `top`, компилятор `gcc` (опции `-o`, `-std=c99`, `-lm` и `-S`). Перенаправление вывода `>`. Возвращаемое значение функции `main`. Функция `exit` из библиотеки `stdlib.h`.

2. Ввод/вывод. Ветвление и циклы:

Ввод и вывод в языке C. Функции `printf` и `scanf` из библиотеки `stdio.h`. Оператор ветвления `if-else`. Использование логических операторов в условии оператора ветвления. 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`. Тернарный оператор. Чем отличается тернарный оператор от `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.

4. 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 присваивания? 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? Нужно знать один простой алгоритм сортировки (выбором или пузырьком) и алгоритм перемножения матриц.

5. Функции

Функции. Параметры и 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 константный указатель). 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.

6. Целочисленные типы данных

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. Типичные 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. Целочисленное переполнение. Неопределённое поведение при целочисленном переполнении. Перечисляемый тип. Константы. Квалификатор типа `const`. Разница между определением константы с помощью директивы `#define` и квалификатора `const`. Создание новых названий для типов с помощью ключевого слова `typedef`. Что такое тип `size_t`. Когда он используется? Типы фиксированной ширины: `int8_t`, `uint8_t`, `int16_t` и другие.

7. Типы чисел с плавающей точкой

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. 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`.

8. Указатели

Указатель. Объявление указателя. Инициализация указателя. Размер указателя на 64-х битных системах. Арифметика указателей. Операторы, применимые к указателям и что они делают: ++, -, прибавление целого числа, вычитание двух указателей, разыменование и оператор взятие индекса ([]). Операции нахождения адреса (&) переменной и операция разыменования * указателя. Указатели разных типов, чем они различаются. Указатель void*. Константный указатель (const int* p) и постоянный указатель (int* const p). Передача в функцию переменных по указателю. Передача массивов в функции.

9. 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, strcmp. Функции sprintf и sscanf, использование этих функций для конвертации числа в строку и наоборот. Аргументы командной строки, argc и argv.

10. Структуры

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

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. Реализация вызова функций с помощью сегмента памяти стек. Стековый кадр, что в нём хранится. Адрес возврата. 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. Динамический массив, его создание с помощью функции malloc. Динамический двумерный массив, его создание с помощью функции malloc. Хранение двумерного массива в виде одномерного массива. Утечки памяти. Основы работы с valgrind.

13. Сегмент памяти Данные

Сегменты памяти data и rodata. 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?

14. 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.

15. Файлы

Открытие и закрытие файла, функции fopen и fclose. Режимы открытия файла "w" и "r". Бинарный и текстовый режимы открытия файла, в чём их отличие. Как хранится перенос строки на операционных системах Linux и Windows. Запись/чтение с помощью функций fprintf и fscanf. Запись/чтение по одному байту с помощью функций fputc и fgetc. Что возвращает функция fgetc? Запись из памяти в файл и чтение из файла в память с помощью функций fwrite и fread. Функции fseek и ftell. Как узнать размер открытого файла.

16. Память

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. Функции memset, memcpy, memmove из стандартной библиотеки. Что такое алиасинг? Strict Aliasing Rule. Неопределённое поведение при алиасинге. Просмотр байт объекта с помощью указателя типа unsigned char*.

Материалы для подготовки:

1. Керниган Ритчи Язык программирования С
lib.mipt.ru/book/266005/
2. Лекторий МФТИ:
lectoriy.mipt.ru/course/ComputerTechnology-Informatics-14L#lectures
3. Язык С и структуры данных в Йелле.
www.cs.yale.edu/homes/aspnes/classes/223/notes.html
4. Кормен, Лейзерсон, Ривест. Алгоритмы: построение и анализ.
lib.mipt.ru/book/16736/