

CRONIS



Low Level Computing Fundamentals

Лекция 01. Двоичная система счисления:

- Системы счисления
- Из десятичной в двоичную и обратно
- Сложение двоичных чисел
- Вычитание двоичных чисел
- Ограниченность разрядности процессора
- Десятичные отрицательные и положительные числа
- Арифметика в регистрах
- Арифметика в двоичных регистрах
- Границы целых чисел в компьютере
- Двоичное представление плавающей точки
- Сравнение чисел
- Двоичное представление плавающей точки в компьютере
- Решение задач

Лекция 02. Создание компьютеров. Машина Тьюринга:

- Определения компьютера:
 - Математическая функция
 - Проблема взаимодействия с компьютером
 - Инструкция, параметр и аргумент
 - Программа и программирование
 - Создания языка программирования
 - Информация и данные
- Состояние, регистры и разрядность процессора
 - Восприятие устройств человеком
 - Пример использования инструкции
 - Регистр
 - Инструкции
 - Лучший способ записи инструкции
- Слово и бит
- Массив
- Принцип выполнения программы и машина Тьюринга
- Оперативная память
- Причина кратности оперативной памяти степени 2
- Байт
 - Двоично-десятичный код
 - Набор символов
 - Появление байта
- Невозможность внедрения ASCII
- Размер регистров и байтовая адресация оперативной памяти
 - Проблема создания новых машин
 - Создание новой оперативной памяти
- Размер байта и набор символов EBCDIC
 - Выбор размера байта
 - Замечание о размере байта
- Удобство шестнадцатеричной системы
- Оперативная память и шестнадцатеричная система
- Термин архитектура. Семейство машин IBM System / 360
- Успех IBM System / 360
- Микропроцессор Intel

Лекция 03. Процессор, типы, битовые операции:

- Договоренности о процессоре
- Регистры в современных процессорах
 - Особенность записи данных в регистры
 - Фундаментальное различие чисел
- Регистры и типы в языках
 - Выбор размера регистра для данных
 - Регистры и типы
 - Фундаментальное различие типов
- Компиляция и машинный код
- Исполнение программы процессором
- Порядок хранения байт в памяти
 - Причины использования порядка от младшего к старшему
- Хранение массива в оперативной памяти
- Термины “операция” и “оператор”
- Кэш процессора
 - Скорость оперативной памяти
 - Линия кэша
 - Скорость кэша
 - Оптимальный размер типа
 - Виды операций
 - Операции и целые типы
 - Приоритеты операций
- Максимальный размер массива
 - C/C++ и Python
 - C#
 - Java
 - JavaScript
- Логические операции
- Битовые операции
 - Битовые операции и целые числа
 - Сокращенная запись двоичных чисел
 - Битовый сдвиг
 - Арифметический сдвиг
 - Логический сдвиг
- Умножение и деление сдвигами
 - Арифметика сдвигов
 - Формула умножения сдвигом положительных чисел
 - Формула деления сдвигом положительных чисел
 - Сдвиг нуля
 - Формула умножения сдвигом отрицательных чисел
 - Формула деления сдвигом отрицательных чисел
- Причина кратности регистров степени 2
- Битовые сдвиги и архитектура x86
- Младшие биты и деление по модулю
 - Особенность взятия младших бит
 - За цикленность деления по модулю
 - Связь взятия младших бит и деления по модулю
- Причина нецелостности работы сдвигов
 - Наследие
 - Теории почему оставили 5 бит

Задачи:

- Скользящее окно:
 - Поиск максимальной подстроки
 - Поиск подстроки с заменного одного символа
 - Поиск максимальной суммы в одномерном массиве
 - Поиск максимальной суммы в двумерном массиве
- Математика:
 - Нахождение предыдущего ближайшего меньшего числа
 - Нахождение следующего ближайшего большего числа
- Битовые операции:
 - Арифметика через битовые операции (6 задач)
 - Смена четных и нечетных бит местами
 - Найти уникальное число среди дубликатов
 - Поменять местами переменные без третьей переменной

Лекция 04. Битовые сдвиги в C++, C#, Java, JavaScript, Python:

- Ограничения сдвигов в C++, C#, Java, JavaScript, Python
- Работа и рекомендуемые размеры сдвигов в C++, C#, Java, JavaScript, Python
- Эмуляция логического сдвига в C++, C#, Java, JavaScript, Python
- Получение 0 сдвигом в C++, C#, Java, JavaScript, Python
- Умножение и деление сдвигами в C++, C#, Java, JavaScript, Python
- Когда использовать знаковые и беззнаковые типы, а также логический сдвиг в C++, C#, Java, JavaScript, Python

Лекция 05. Битовые манипуляции:

- Договоренности про битовые манипуляции
 - Восприятие целых чисел как набора бит
 - Псевдокод битовых манипуляций
 - Размер типа при битовых манипуляциях
- Перевод псевдокода на C++ и C#
- Перевод псевдокода на Java и JavaScript
- Перевод псевдокода на Python
- Получение младшего или старшего бита
- Обходы двоичных данных в C++, C#, Java, JavaScript, Python
- Получение бита в индексе i
- Проверка, что 1 установлена в индексе i
- Инверсия бита в индексе i
- Запись 1 по индексу i
- Запись 0 по индексу i
- Заполнение 0 младших i бит
- Заполнение 1 младших i бит
- Заполнение 0 старших i бит
- Заполнение 1 старших i бит

Задачи:

- Сколько надо поменять бит чтобы из a получить b
- Нахождение последовательности единиц
- Создать последовательность единиц от i до j
- Вставить определенные биты от i до j
- Записать последовательность бит в одномерный массив
- Математика:
 - Нахождение двоичного предыдущего ближайшего меньшего числа
 - Нахождение двоичного следующего ближайшего большего числа

Лекция 06. Битовый вектор:

- Сжатие аргументов функции
- Битовый вектор
 - Проблема **boolean** массива
 - Использование **int** массива как набора бит
 - Битовый вектор
 - Потери памяти в битовом векторе
- Формулы битового вектора
 - Формулы доступа к биту в массиве
 - Формула получения размера массива
- Лучший размер типа массива и кол-во бит для хранения в битовом векторе
 - Свойство целых неотрицательных чисел
 - Связь кэша и скорости работы битового вектора
 - Кол-во чтений из памяти в кэш при **X** кратном **512**
 - Кол-во чтений из памяти в кэш при **X** не кратном **512**
 - Связь кэша и скорости работы битового вектора
 - Влияние размера типа массива на потери памяти в битовом векторе
 - Связь скорости языков и кол-ва хранимых бит
- Разница в памяти битового вектора и **boolean** массива
- Возможности **boolean** массива и битового вектора
- Реализация функции выделения памяти для битового вектора
 - Физический размер битового вектора
 - Выделение памяти для битового вектора
- Реализация функции проверки порядкового индекса бита
- Формулы доступа к порядковому индексу бита
- Реализация функций установки, очистки и получения **boolean** значения
- Полный код и пример использования битового вектора
- Причина хранения битового вектора в знаковом типе
- Глубокая оптимизация битового вектора в C#, Java и JavaScript
- Стандарты размерностей величин
 - SI
 - IEC
 - JEDEC

Задачи:

- Напечатать все дубликаты используя 4 KiB памяти
- Найти пропущенное **int** число среди 4,000,000,000 чисел, используя 1 GiB
- Преобразование 2D координат в 1D
- Нарисовать линию на черно-белом экране
- Найти пропущенное число, используя 128 KiB