



1.9. Ассемблер. Расчёт времени выполнения команды. Форматы команд, правила записи, поля, трансляция команд ассемблера в машинный код;

Лабораторные работы:

2.1. Позиционные системы счисления;

2.2. Ревизия памяти ПК и его системных ресурсов;

2.3. Работа с оперативной памятью ПК;

2.4. Программа в машинном коде. Шаблон программы на Ассемблере. Отладчик.

Практические занятия:

3.1. Ассемблер. Ручная трансляция команды цикла, расчёт смещения. Пример программы и ручной трансляции команд ассемблера в машинный код в среде языка Pascal;

3.2. Язык Ассемблер (TASM), его функции. Пример программы. Определение сегментов: стека, данных, кодового;

3.3. Ассемблер. Полное определение стека, его глубина. Правила организации стека, обращения к элементам стека и размещения его в памяти, направление его роста. Предекрементный-постинкрементный, LIFO стек. Шаблон программы;

3.4. Ассемблер. Написание текста программы. Цикл устранения синтаксических ошибок. Получение объектного кода. Редактор связей и его назначение. Получение исполняемого файла;

3.5. Ассемблер. Полно экранный отладчик (TD). Отладка, использование ресурсов отладчика: пошаговое выполнение, точки разрывов, окна содержимого регистров, стека, областей памяти. Псевдооператоры определения данных. Примеры. Команды цикла: организация вложенных циклов с использованием стека;

3.6. Ассемблер. Разрядная сетка. Представление числа в ПК. Целые и дробные числа без знака. Числа со знаком. Обратный и дополнительный код. Числовая ось;

3.7. Ассемблер. Бит переноса и признак переполнения. Формальные признаки потери точности расчёта. Примеры программ;

3.8. Ассемблер. Побайтная арифметика. Операции с числами повышенной точности. Пример программы. Арифметические команды.

Представление чисел. Формат. Числа со знаком. Коды. Двоично-десятичный код. Упакованные числа. Сложение. Перенос. Команды сложения. Формат. Флаги. Примеры. Вычитание. Заем. Команды вычитания. Формат. Флаги. Примеры.

Расчётно-графическая работа:

- индивидуальные задания на РГР назначаются из примерного нижеприведённого перечня:

4.1. Каков алгоритм (общий) перевода чисел из одной позиционной системы счисления (ПСС) в другую?

4.2. В чём преимущества «машинного» алгоритм перевода чисел (простым перебором) из одной позиционной системы счисления в другую и в чём его недостатки?

4.3. В чём состоит алгоритм смены содержимого счётчика команд процессора ПК?

4.4. Каковы формальные правила работы в ПСС при умножении (делении) на число, равное основанию. Пример для двоичной системы?

4.5. Привести алгоритм распознавания чётного (нечётного) целого числа в ПСС с основанием 2.

4.6. Составить алгоритм умножения числа в ПСС (двоичной, восьмеричной, шестнадцатеричной) на множитель, кратный основанию?

4.7. Составить алгоритм деления числа в ПСС (двоичной, восьмеричной, шестнадцатеричной) на число, кратное основанию?

4.8. Предложить алгоритмы смены знака числа в ПСС с основанием 2.

- 4.9. Как составить алгоритм циклического сдвига информации в пяти байтах?
- 4.10. Какой алгоритм поднятия флага переполнения при арифметических действиях процессора?
- 4.11. Привести алгоритм («общий») в виде блок-схемы перевода чисел из одной позиционной системы счисления (ПСС) в другую;
- 4.12. Показать преимущества и недостатки «машинного» алгоритм перевода чисел (простым перебором) из одной позиционной системы счисления в другую.
- 4.13. Дать иллюстрацию алгоритма смены содержимого счётчика команд процессора ПК;
- 4.14. На примерах показать формальные правила работы в ПСС при умножении (делении) на число, равное основанию. Дать примеры для двоичной системы счисления;
- 4.15. Привести алгоритм распознавания чётного (нечётного) целого числа в ПСС с основанием 2.
- 4.16. Составить алгоритм умножения числа в ПСС (двоичной, восьмеричной, шестнадцатеричной) на множитель, кратный основанию.
- 4.17. Составить алгоритм деления числа в ПСС (двоичной, восьмеричной, шестнадцатеричной) на число, кратное основанию.
- 4.18. Предложить алгоритмы смены знака числа в ПСС с основанием 2.
- 4.19. Составить универсальный алгоритм циклического сдвига информации в нескольких байтах.
- 4.20. Сформулировать обобщённый алгоритм поднятия флага переполнения при арифметических действиях процессора. Показать работу флага на примерах операций со знаковыми числами.

Соответствующий, проработанный дополнительный пункт задания включается в соответствующий отчёт по лабораторной работе. Чаще всего эти пункты РГР включают программный код или набор структурных схем. Работа должна быть оформлена со всеми требованиями ЕСКД.

Год начала подготовки (по учебному плану)  
Образовательный стандарт (СУОС)

2026  
от 20.12.2023