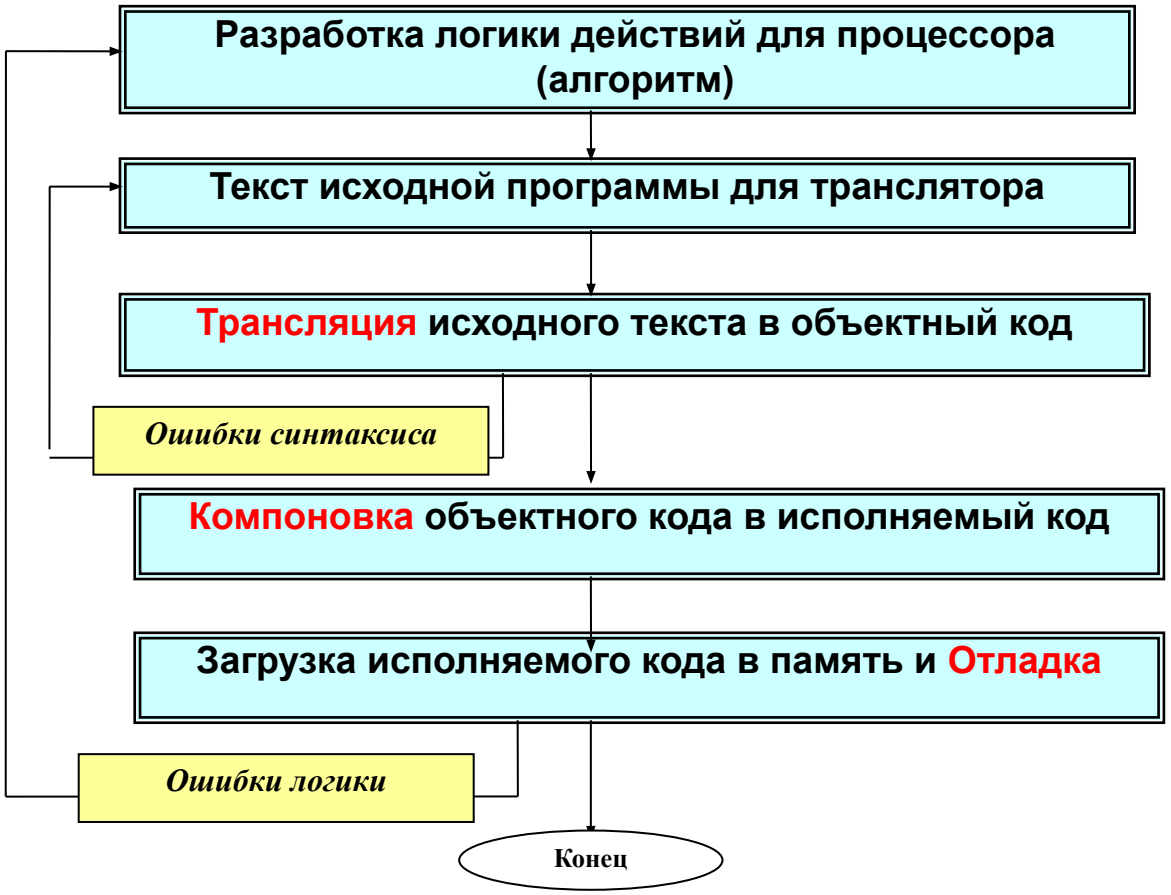
Жизненный цикл разработки низкоуровневой программы



Текстовый файл исходной программы

- Файл с текстом исходной программы должен иметь расширение .asm
- Каждая строка текста должна заканчиваться Enter
- *Стандартный текстовый редактор.* Для правильного представления русскоязычных комментариев редактор должен поддерживать кодировку OEM (кодовая страница 866)
- При наборе строк программы на языке ассемблера стремиться к большей читабельности, т.е. структурировать строки

Исходная программа. Файл metod.asm

```
; Получить 2-х байтную сумму двух однобайтных кодов в регистре AX
    .386
    dseg  segment use16
        a     db  34
        b     db  75h
    dseg  ends
    cseg  segment use16
    assume ds:dseg, cs:cseg
    m1:mov cx, dseg
         mov ds, cx
    ; расширим байты до 2-х байтных в регистрах и сложим
         movzx ax, ds:a
         movzx bx, ds:b
         add   ax, bx
    ; завершение исполнения
         mov ah, 4ch
         int 21h
    cseg ends
         end  m1
```

Трансляция

- Задача транслятора - преобразовать символически записанные команды процессора и данные в машинные коды
- Машинные коды он записывает в файл с расширением .obj - «объектный файл»
- Листинг трансляции (файл с расширением .lst)- это текстовый протокол результатов трансляции, включая сообщения о синтаксических ошибках

Протокол трансляции исходного текста - metod.lst

```
1   ; metod.asm - Получить        2-х байтную сумму двух однобайтных кодов
2                          .386
3   0000                   dseg  segment use16
4   0000 22                      a          db  34
5   0001 75                      b          db  75h
6   0002                   dseg  ends
7   0000                   cseg  segment use16
8                          assume   ds:dseg, cs:cseg
9   0000 B9 0000 s         m1:       mov cx, dseg
10  0003 8E D9                       mov ds, cx
11                         ; расширим байты до 2-х байтных и сложим
12  0005 0F B6 06 0000 r             movzx ax, ds:a
13  000A 0F B6 1E 0001 r             movzx bx, ds:b
14  000F 03 C3                       add   ax, bx
15                         ; завершение исполнения
16  0011 B4 4C                       mov ah, 4ch
17  0013 CD 21                       int 21h
18  0015                   cseg ends
19                         end  m1
```

16-разрядные внутрисегментные адреса

Формат протокола трансляции

- Первая колонка - номера строк исходного текста.
- Вторая колонка - показывает *внутрисегментные адреса начала размещения команд и данных*. Для 16-разрядной программы внутрисегментный адрес - четыре hex – цифры.
- Третья колонка - *машинные коды команд и данных* в hex-виде.

Символ S рядом с машинным кодом говорит лишь о том, что команда содержит указатель адреса сегмента памяти.

Символ R после машинного кода команды говорит о том, что машинный код содержит внутрисегментный адрес операнда

- Четвертая колонка показывает транслируемую символическую строку из исходного текста программы

Компоновка

- Компоновщик создает из машинного кода (.obj) «исполняемый код» для операционной системы. исполняемый код дополнен технической информацией о программных сегментах, необходимой ОС при загрузке кодов из программных сегментов в память
- Исполняемый код сохраняется в файле с расширением .exe или .com.

Отладка

Для отладки исполняемого кода после загрузки его в память используются программы-отладчики.

Они позволяют:

- управлять процессором во время исполнения команд программы:
  - исполнять команды с остановкой после каждой
  - исполнять команды до определенного адреса
  - возвращать процессор к уже исполненной команде и другое
- в моменты приостановки процессора просматривать или изменять текущее содержимое байтов памяти и регистров процессора
- и другое

Инструментальные средства

Классический пакет Borland Turbo Assembler (TASM) включает в себя необходимые утилиты для трансляции, компоновки и отладки 16-разрядных программ для процессоров x86:

- Tasm.exe – транслятор
- Tlink.exe – компоновщик
- Td.exe - отладчик

Файловые менеджеры

- Программы TASM являются консольными утилитами, то есть требуют режима командной строки.
- Самой удобной средой для работы с этими утилитами является не чистый консольный режим, а среда файловых менеджеров (FAR, Total Commander, NC, VC и др.)
- В интерфейс файловых менеджеров интегрированы:

- консольная среда командного режима,
- удобный панельный интерфейс для работы с дисками, каталогами и файлами,
- встроенный текстовый редактор (с кодовой страницей 866)

Транслятор TASM

- Запуск транслятора Tasm.exe из командной строки:

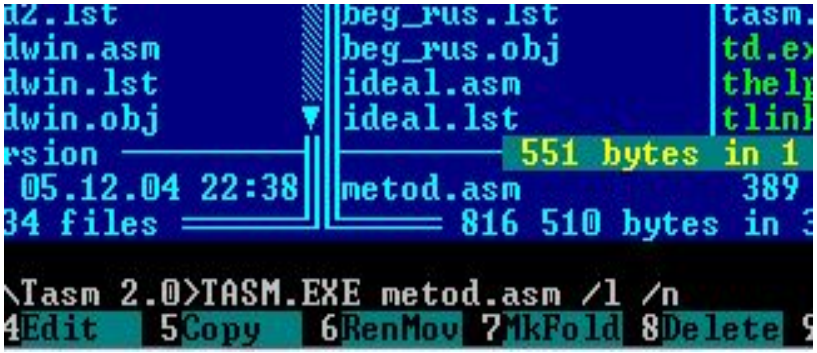
  〉 tasm.exe metod.asm /l /n

Объектный файл metod.obj по умолчанию будет создан в каталоге, где находится транслятор.

Ключи трансляции:

/l - создать файл-протокол трансляции

/n – не создавать в конце протокола список символических имен (для экономии бумаги при печати)



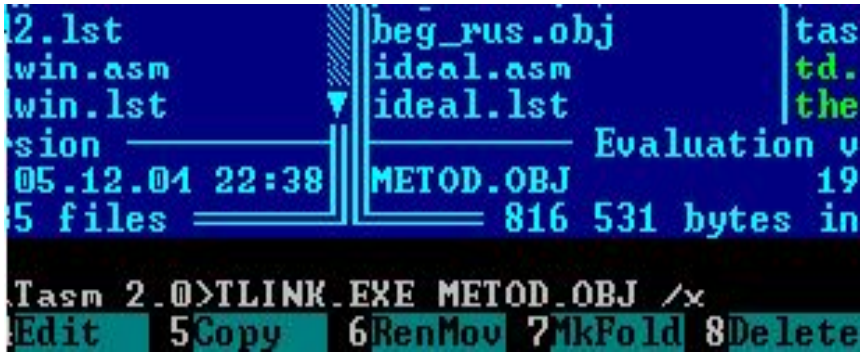
Компоновщик TLINK

- Запуск компоновщика **Tlink.exe** из командной строки:

  〉 tlink.exe metod.obj /x

Исполняемый файл metod.exe создается в каталоге, где был объектный модуль.

Ключ: **/x** - не создавать файл карты связи (.map)



Отладчик Td.exe

- Загрузка отладчика из командной строки вместе с исполняемой программой:

〉 td.exe metod.exe