

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»  
Филиал  
«Минский радиотехнический колледж»

Учебная дисциплина «Встраиваемые микропроцессорные системы»

по выполнению лабораторной работы  
«Программирование на языке Ассемблер. Отладчик программ  
языка Ассемблер»

Минск  
2017

»

## 1. :

Освоение инструментальных средств создания и отладки программ на языке ассемблера. Знакомство с процессом трансляции и компоновки программ на ассемблере. Разработка и отладка программ на языке ассемблера.

## 2.

Выполнить ассемблирование и компоновку программы. Исследовать программу с помощью отладчика.

## 3.

Техническое задание, ПК.

## 4.

## 4

Полный цикл создания программы на Ассемблере можно представить в виде последовательности четырех этапов (рисунок 4.1):

- подготовка исходного текста программы и оформление его в виде текстового файла (одного или нескольких) с помощью текстового редактора в формате DOS – с расширением ASM.

- ассемблирование программы с применением ассемблера Tasm, результатом чего будет объектный файл с расширением OBJ. Если программа состоит из нескольких файлов (модулей), они ассемблируются независимо. Если при ассемблировании обнаруживаются ошибки, то объектный файл не создается, а выдается сообщение об ошибках. По устранении ошибок ассемблирование нужно повторить. Объектный файл (двоично-кодированное представление программы) не может исполняться, так как не содержит информации о загрузке сегментов программы в память компьютера.

- компоновка программы выполняется компоновщиком (редактором связей) Turbo Linker и заключается в доработке объектного файла до исполняемой формы с назначением стартового адреса программы. При компоновке программы из нескольких частей компоновщик объединяет объектные модули в один исполняемый файл. Исполняемый файл обычно имеет расширение EXE.

- отладка программы с использованием отладчика AFDPRO, программы отладчика Turbo Debugger (td.exe). Этот этап не всегда нужен и требуется при обнаружении в программе неявных семантических и алгоритмических ошибок, не обнаруживаемых ассемблером, либо при задаче исследования работы про-

граммы.



Рисунок 4.1 – Этапы создания ассемблерной программы

Исходный модуль программы создается в любом текстовом редакторе, например, в «Блокноте» и сохраняется в виде файла с именем, присвоенным по правилам MS DOS, с обязательным расширением *asm*.

Для получения исполняемого модуля, который можно запустить на выполнение, требуется последовательно выполнить этапы трансляции и компоновки. Для этого используются программы, входящие в состав пакета Ассемблера Turbo Assembler (TASM) фирмы Borland.

Трансляция производится с помощью компилятора Турбо Ассемблера, который является исполняемой программой *tasm.exe*, работающей в режиме командной строки. Он вызывается командой DOS:

*tasm /z/zi/n <имя файла><имя файла><имя файла>*,

где */z* – ключ, разрешающий вывод на экран строк исходного текста программы, в которых Ассемблер обнаружил ошибки;

*/zi* – ключ, управляющий включением в результирующий файл полных сведений о номерах строк и именах исходного модуля;

*/n* – ключ, который исключает из листинга информацию о символических обозначениях в программе.

Следующие далее имена трех файлов – исходного (*\*.asm*), объектного (*\*.obj*) и листинга (*\*.lst*) можно использовать без расширений.

Если исходный модуль не содержит ошибок, то на экран выводится со-

общение об успешной трансляции, а в текущем каталоге появятся новые файлы – объектный (.obj) и листинга (.lst).

Компоновка объектных модулей с библиотечными модулями производится вызовом компоновщика tlink.exe из командной строки:

```
tlink/v <имя файла>
```

Ключ /v передает в загрузочный файл информацию, используемую при отладке программ. Следующее далее имя файла обозначает имя объектного модуля. Расширение в этом имени можно не указывать.

В случае успешного окончания компоновки в текущем каталоге появляется исполняемый файл – загрузочный модуль .exe, и файл карты сборки .map. Загрузочный модуль может быть запущен на выполнение командой DOS.

Для отладки создаваемых программ используется программа отладчик.

```
TITLE "Демонстрационная программа HELLO.ASM"
```

```
IDEAL
```

```
MODEL small
```

```
STACK 256
```

```
DATASEG
```

```
Prompt DB 'Это время после полудня? (Да/Нет - Y/N)$'
```

```
GoodMorning DB 13,10,'Доброе утро!',13,10,''
```

```
GoodAfternoon DB 13,10,'Здравствуйте!',13,10,''
```

```
NonSense DB 13,10,'Что-что?',13,10,''
```

```
CODESEG
```

```
Start: MOV AX,@DATA ;Установка в DS адреса
```

```
MOV DS,AX ; сегмента данных
```

```
MOV DX,OFFSET Prompt ;Сообщение-запрос
```

```
MOV AH,9 ;Подготовка вывода сообщения
```

```
INT 21H ;Функция вывода на экран
```

```
MOV AH,1 ;Подготовка ввода символа
```

```
INT 21H ;Функция ввода символа с клавиатуры
```

```
CMP AL,'Y' ;Буква Y?
```

```
JZ Afternoon ;Да, время после полудня
```

```
CMP AL,'N' ;Буква N?
```

```
JZ Morning ;Нет, время до полудня
```

```
MOV DX,OFFSET NonSense ;Указание на "Что-что?"
```

```
JMP SHORT Disp
```

```
Afternoon:
```

```
MOV DX,OFFSET GoodAfternoon ;Указание на "Здравствуйте"
```

```
JMP SHORT Disp
```

```
Morning:
```

```
MOV DX,OFFSET GoodMorning ;Указание на "Доброе утро"
```

Disp: MOV AH,9	;Подготовка вывода на экран
INT 21H	;Функция вывода сообщения на экран
Exit: MOV AX,4C00H	;Подготовка выхода из программы
INT 21H	;Функция выхода - останов программы
END Start	;Конец программы / точка входа

Рассмотрим ассемблирование программы на примере простой интерактивной программы Hello.asm. Для ассемблирования файла Hello.asm в командной строке нужно набрать TASM HELLO.ASM и нажать клавишу ввода. Поскольку особое имя для объектного файла не задано, будет создан объектный файл с тем же программным именем HELLO.OBJ. Расширение имени файла ASM вводить не требуется – TASM принимает его по умолчанию. На экране появится следующее:

```
Turbo Assembler Version 2.0 Copyright (C) 1988,1990 Borland International
```

```
Assembling file: Hello.asm
```

```
Error messages: None
```

```
Warning messages: None
```

```
Passes 1
```

```
Remaining memory: ***K
```

(На месте трех звездочек будет выдан объем свободной оперативной памяти DOS.)

Сообщение о том, что ассемблирование завершено без ошибок и предупреждений нет.

Замечания:

- Предупреждение – не ошибка, но его игнорирование может привести к неприятностям в дальнейшей работе с программой, поэтому лучше своевременно отреагировать на него.

- Если использовать другое имя программы, сообщения на экране соответственно изменятся.

Для компоновки программы нужно ввести в командную строку TLINK HELLO.OBJ. Здесь расширение имени OBJ тоже не обязательно. По завершении компоновки будет сформирован файл HELLO.EXE с выводом на экран сообщения

```
Turbo Linker Version 3.0 Copyright (C) 1987,1990 Borland International
```

Теперь программу HELLO.EXE можно запустить. Результатом исполнения будет вывод на экран сообщения:

```
Это время после полудня? (Да/Нет - Y/N)
```

Курсор будет мерцать после последнего символа в ожидании ответа. Введите букву Y. Программа ответит:

```
Здравствуйте!
```

Если будет введена буква N, программа ответит:

Доброе утро!

Если будет введена какая-то другая буква, программа ответит:

Что-что?

Внимание! Буквы Y и N нужно вводить в верхнем регистре - как прописные. Иначе ответом всегда будет "Что-что?".

В ходе ассемблирования или компоновки можно выбирать различные особенности их исполнения, что задается ключами в командной строке Tasm или Tlink. Для вывода списка ключей командной строки нужно набрать просто TASM либо TLINK и нажать ввод.

Ключи записываются в виде одной или нескольких букв. Для задания ключа нужно набрать косую черту (или дефис) и нужную букву между командой TASM либо TLINK и именем программы, которая ассемблируется либо компоуется. Например, для ассемблирования программы HELLO.ASM и получения файла с распечаткой (листингом), где содержится описание хода ассемблирования, нужно ввести команду:

TASM /L HELLO.

Команды и ключи можно набирать прописными и строчными буквами.

Исполните эту команду и затем рассмотрите файл HELLO.LST. В распечатке каждая строка начинается с номера, затем следуют байты объектного кода и собственно строка программы. Кроме того, TASM выводит в этом файле таблицу идентификаторов, где содержится информация о метках и сегментах, включая значение и тип каждой метки и атрибуты каждого сегмента. При ассемблировании программы можно использовать в одной командной строке несколько ключей, разделяя их косыми чертами.

Некоторые примеры для команды TASM:

TASM /L /C HELLO – команда дополняет файл распечатки таблицей перекрестных ссылок, указывающей, где определена каждая метка и где на нее есть ссылка;

TASM /L /N HELLO – исключает таблицу идентификаторов из распечатки;

TASM /ML HELLO – включает различение прописных и строчных символов в идентификаторах;

TASM /ZI HELLO – команда добавляет в HELLO.OBJ информацию, необходимую для использования отладчика.

Для команды TLINK:

TLINK/M HELLO – ключ /M приводит к созданию файла отображения или файла загрузки HELLO.MAP, где перечисляются имена, адреса загрузки и размеры всех сегментов, входящих в программу;

TLINK /X /V HELLO – команда позволяет загрузить HELLO.EXE в отладчик, запрещая создание (ключ /X) файла отображения.

TLINK /T <FILE\_NAME.OBJ> – ключ /T обеспечивает создание файла типа COM.

Внимание: программа типа COM требует особой организации, и, если

этот ключ использовать некстати, результирующая программа либо не будет создана, либо будет неработоспособна.

Отладчик можно использовать и когда ключи, указанные как полезные для него, не указывались. Но при надобности отладки их лучше использовать.

### Debugger AFDPror - AFDPROK.EXE

Tasm умеет ассемблировать синтаксически правильные программы, но «не понимает», что именно эти программы делают. Часто программа работает не так, как должна бы работать. В такой ситуации может помочь программа-отладчик, предназначенная для поиска и исправления логических (семантических) ошибок. Подобно всем отладчикам, AFDPror может работать в режиме супервизора, беря на себя управление программой в режиме пошагового (командного) исполнения кода программы. При этом можно изменять значения операндов в памяти, значения регистров и флагов.

Горячие клавиши – все в строке-подсказке (обеспечивают переход по окнам и пр.)

Команды – набираются в строке "CMD >" и нажимается клавиша ввода (Enter):

QUIT – выйти из программы

PD addr, length, filename - параметры addr и length суть шестнадцатеричные

адрес (000 или 100 без буквы H в конце) и длина дисассемблируемого кода в байтах (до ffff без буквы H в конце)

filename – имя файла, в который будет сохранен результат дисассемблирования (например, list.txt)

G start\_addr, break\_addr – где start\_addr есть адрес перехода на код тестируемой программы, а break\_addr – адрес первого останова.

Остальные команды можно найти, вызвав окно помощи F4. Прервать выполнение программы можно нажатием клавиш Ctrl-Esc.

Для ознакомления с работой отладчика при изучении языка ассемблера исследуем программу HELLO под его управлением. Снова выполним ассемблирование и компоновку программы с ключами, добавляющими отладочную информацию в файлы OBJ и EXE:

```
TASM /ZI HELLO.ASM
```

```
TLINK /V HELLO.OBJ
```

```
AFDPROR HELLO.EXE
```

После выполнения последней команды на экране появится заставка отладчика. При нажатии любой клавиши оно сменится рабочим окном отладчика.

## 4

Самые частые ошибки при написании первых программ. Обнаружение этих ошибок отмечается при ассемблировании (с указанием номера строки):

- искажено имя команды или директивы;
- опущено двоеточие ':' после имени метки;

- неправильно записан пользовательский идентификатор;
- в качестве оператора указан неопределенный ранее идентификатор;
- использовано имя служебного идентификатора в качестве пользовательского;

- значение инициализируемой константы превышает допустимую величину, например:

DB 400 ;400 > 255, т. е. максимального значения в формате BYTE

- поставлена запятая в конце списка элементов при множественной инициализации, например:

DW 1,3,5,400, ;лишняя запятая в конце списка

- нет круглых скобок при операторе DUP, например:

DB 4 DUP 7 ;надо "4 DUP (7)"

- ошибка в написании имени модели памяти, или не указана модель памяти при использовании упрощенных директив;

- опущена одна или обе квадратные скобки [], заключающие адресное выражение в режиме Ideal, например:

MOV AX, [BX+SI ;содержимое по адресу BX+SI отправить в AX

- не совпадают типы операндов команды, например:

MOV AL,[VAR\_W]

(ошибка – ранее переменная VAR\_W определена в формате WORD)

MOV AX,[VAR\_B]

(ошибка – ранее переменная VAR\_B определена в формате BYTE)

- требуется явно указать тип операнда, например:

MOV [BX],1

(ошибка – надо явно указать тип ячейки памяти: [BYTE BX] или [WORD BX])

- операнд в текущей инструкции не может быть ссылкой на адрес памяти:

MOV [OPER\_1],[OPER\_2]

(команда MOV не может передавать данные из одной ячейки памяти в другую);

- недопустимый режим адресации, например:

ADD [DX+SI],AX

(регистр DX нельзя использовать в косвенной адресации)

- недопустимая команда для выбранного типа процессора (по умолчанию используется i8086);

- адрес назначения в команде условного перехода вне допустимого для процессора диапазона – не принадлежит интервалу (-128, +127);

- непарное использование директивы ENDP, например:

PROC ADDITION

.....

ENDP SUBTRACTION

(ошибка – должно быть указано имя процедуры ADDITION). Аналогичные ошибки могут появиться при использовании других парных директив

"SEGMENT <NAME>...ENDS <NAME>", "MACRO <NAME>...ENDM



<NAME>" и пр.

- в конце программы отсутствует директива END.

## 5.

1. Выполнить ассемблирование и компоновку программы HELLO.ASM с ключами, предполагающими использование отладчика AFDPror. Исследовать программу HELLO.EXE с помощью отладчика.
2. Выполнить ассемблирование и компоновку программных файлов с ключами, предполагающими получение распечатки и отладочной информации.
3. Создать копию файла программы с измененными названиями – для экспериментов по ассемблированию (получение распечатки: TASM/L <FILE.ASM>) с добавленными синтаксическими ошибками (достаточно 5 – 7 ошибок). При введении ошибок в правильно составленные программы можно руководствоваться перечнем типичных ошибок, данным выше. Ошибки вместе с замечаниями на них Turbo Assembler выводит в распечатке.

*Лабораторная работа № \_\_\_\_\_*

*Номер учебной группы \_\_\_\_\_*

*Фамилия, инициалы учащегося \_\_\_\_\_*

*Дата выполнения работы \_\_\_\_\_*

*Тема работы: \_\_\_\_\_*

*Цель работы: \_\_\_\_\_*

*Оснащение работы: \_\_\_\_\_*

*Результат выполнения работы: \_\_\_\_\_*

## 7.

1. Запишите команду для ассемблирования файла F\_1.ASM с учетом формирования распечатки и возможности использования отладчика.
2. Запишите команду для компоновки файла F\_1.OBJ, разрешающую использование отладчика и запрещающего создание файла отображения.
3. В чем состоит основная разница между исполняемыми файлами программ типа EXE и COM?
4. Запишите инструкции, определяющие начало и конец программного кода.
5. В чем состоит различие между ошибками и предупреждениями, создаваемыми при ассемблировании? Опишите нужные действия при обнаружении ошибки.
6. Опишите основные форматы команд процессоров.
7. Какие из приведенных ниже команд записаны с ошибками и каковы эти ошибки? Считайте все идентификаторы переменными-словами в сегменте данных:

```
MOV BP,AL  
MOV [OP_1+BX+DI+12],AX  
MOV [OP_1],[OP_2]  
MOV AX,[OP_3+DX]  
MOV CS,AX  
MOV [BX+SI],2  
LEA BX,OP_2  
MOV BX,OFFSET [OP_2]
```

Финогенов, К. Г. Основы языка Ассемблера [Текст] / К. Г. Финогенов. – М.: Радио и связь, 2000.

Финогенов, К. Г. Использование языка Ассемблера [Текст]: учеб. пособие для вузов / К.Г. Финогенов. – М.: Горячая линия Телеком, 2004.

Юров, В. И. Assembler [Текст]: учеб. пособие для вузов / В. И. Юров. 2-е изд. – СПб.: Питер, 2007.