

Машинно-зависимые языки программирования

Лабораторная работа №2

“EXE-файлы. Сегменты. Простейший ввод-вывод средствами DOS”

Справочная информация

Логическая структура памяти. Сегменты

Любая программа состоит из одного или нескольких сегментов (блоков памяти размером до 64 КБ). Сегменты могут быть следующих типов: кода, данных, стека. За адрес начала сегмента отвечают сегментные регистры: для кода - CS, для стека - SS, для данных - DS и дополнительные ES, FS, HS.

Составление программ на ассемблере

Как и на других языках программирования, программа на ассемблере может состоять из нескольких файлов - модулей. При компиляции (трансляции) каждый модуль превращается в объектный файл, далее при компоновке объектные файлы соединяются в единый исполняемый модуль.

Директивы ассемблера - ключевые слова в тексте программы на языке ассемблера, влияющие на процесс ассемблирования или свойства выходного файла.

Модули обычно состоят из описания сегментов будущей программы с помощью директивы SEGMENT.

Пример:

```
имя SEGMENT [READONLY] выравнивание тип разряд 'класс'
...
имя ENDS
```

Параметры:

- Выравнивание - расположение начала сегмента с адреса, кратного какому-либо значению. Варианты: BYTE, WORD (2 байта), DWORD (4 байта), **PARA (16 байт, по умолчанию)**, PAGE (256 байт).
- Тип: PUBLIC (сегменты с одним именем объединятся в один); STACK (для стека); COMMON (сегменты будут “наложены” друг на друга по одним и тем же адресам памяти); AT <начало> - расположение по фиксированному физическому адресу, параметр - сегментная часть этого адреса; PRIVATE - вариант по умолчанию.
- Класс - метка, позволяющая объединить сегменты (расположить в памяти друг за другом).

Выделение памяти

Директивы выделения памяти: DB (байт), DW (слово), DD (двойное слово).

Описание строки программы

метка команда/директива операнды ; комментарий

Любое поле может быть опущено.

Метка в коде заканчивается двоеточием и обозначают ссылку на команду, расположенную за ней.

```
mov cx, 5
label1:
    add ax, bx
    loop label1
```

Метка в описании данных является ссылкой на переменную, расположенную после неё. Метка не является директивой выделения памяти (см. л/р 2).

метка label тип

Допустимые типы: BYTE, WORD, DWORD, FWORD, QWORD, TBYTE (для данных), NEAR, FAR (для указателей на команды).

Команда организации цикла

Команда LOOP <метка> - команда организации цикла со счётчиком. Уменьшает регистр CX на 1 и выполняет переход на метку, если новое значение регистра не равно нулю.

Директива ASSUME

ASSUME регистр:имя сегмента

Является инструкцией компилятору, указывающей, какой сегментный регистр с каким сегментом будет связан во время работы программы. Используется для контроля правильности обращения к переменным и автоматического определения сегментного префикса в машинных командах работы с памятью.

EXE-файлы

Исполняемые файлы с расширением EXE, используемые в ОС DOS и Windows (и некоторых других).

Может быть нескольких форматов:

- MZ - 16-битный формат, основной для DOS;
- NE - 16-битный формат старых версий Windows;
- LE, LX - формат OS/2;
- PE - 32- и 64-битный формат современных Windows (начиная с Windows 95).

MZ — стандартный формат 16-битных исполняемых файлов с расширением .EXE для DOS. Назван так по сигнатуре — ASCII-символам MZ (4D 5A) в первых двух байтах. Эта сигнатура — инициалы Марка Збиковски, одного из создателей MS-DOS.

Формат был разработан как замена устаревшему формату .COM. Исполняемые файлы MZ включают метаданные, могут иметь размер больше 64 Кбайт и использовать несколько сегментов памяти различного типа (кода, данных и стека), точка входа в программу также может быть в любом месте (в файлах .COM выполнение команд всегда начинается непосредственно с начала файла). Метод загрузки исполняемого файла определяется по сигнатуре: при её наличии обрабатывается MZ-заголовок, при отсутствии файл запускается как .COM — независимо от расширения файла (например, в последних версиях MS-DOS интерпретатор командной строки COMMAND.COM на самом деле является EXE-файлом).

Исполняемые файлы более поздних форматов для Windows начинаются с MZ-заглушки. Обычно заглушка, добавляемая компиляторами, выводит сообщение наподобие «This program cannot be run in DOS mode» («Эту программу невозможно запустить в режиме DOS»).

([https://ru.wikipedia.org/wiki/MZ_\(%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82\)](https://ru.wikipedia.org/wiki/MZ_(%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82)))

EXE-файлы состоят из заголовка и собственно загружаемой части (тела файла).
Заголовок MS-DOS (размер 40H байт):

| Адрес | Тип | Имя | Описание |
|-------|------|--------------------|---|
| 00h | word | Сигнатура | Магическая сигнатура DOS-файла - два символа "MZ" |
| 02h | word | Extra bytes | Количество байт на последней странице файла |
| 04h | word | Pages | Количество страниц в файле |
| 06h | word | Relocation items | Количество релокейшенов |
| 08h | word | Header size | Размер заголовка в параграфах |
| 0Ah | word | Minimum allocation | Мин. выделение памяти в параграфах |
| 0Ch | word | Maximum allocation | Макс. выделение памяти в параграфах |
| 0Eh | word | Initial SS | Начальное (относительное) значение регистра SS |
| 10h | word | Initial SP | Начальное значение регистра SP |
| 12h | word | Checksum | Контрольная сумма |

| | | | |
|-----|-------|---------------------|--|
| 14h | word | Initial IP | Начальное значение регистра IP |
| 16h | word | Initial CS | Начальное (относительное) значение регистра CS |
| 18h | word | Relocation table | Адрес на релокейшны и программу-заглушку |
| 1Ah | word | Overlay | Количество оверлеев |
| 1Ch | word | Overlay information | Зарезервировано |
| 24h | word | OEMIdentifier | Для OEMInfo |
| 26h | word | OEMInfo | Информация о программе |
| 28h | word | Res1[10] | Зарезервировано |
| 3Ch | dword | PEHeaderAddr | Адрес в файле заголовка PE |

(<http://mzc.narod.ru/Creating/Step008.htm>, <https://wiki.osdev.org/MZ>)

Практическое задание

1. Набрать программу из файла ASM1_AFD.pdf, скомпилировать и сформировать исполняемый файл с помощью команд `masm+link` или `ml`.
2. Проанализировать получившиеся файлы.
3. Запустить программу, убедиться в работоспособности.
4. Запустить программу в отладчике, пронаблюдать изменение сегментного регистра DS и изучить содержимое сегментов кода и данных.
5. Дописать исходный текст программы так, чтобы строка выводилась на экран 3 раза.