

## **КОНТРОЛЬНАЯ РАБОТА ПО ДИСЦИПЛИНЕ «ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ»**

Цель работы – изучить назначение и применение СОМ-файлов и перевод ассемблерных программ в формат СОМ-файлов.

### **3.1. Основные сведения**

В прошлой работе было рассмотрено написание, ассемблирование и выполнение программы в EXE-формате. Компоновщик TLINK автоматически генерирует особый формат для EXE-файлов, в котором присутствует специальный начальный блок (заголовок) размером не менее 512 байт.

Для выполнения программы можно также создавать СОМ-файлы. В операционной системе для преобразования EXE-файлов в СОМ-файлы используется программа EXE2BIN.COM. Получение файла в формате СОМ возможно на этапе компоновки применением ключа /t, например: `tlink.exe /t exasm3.obj`.

Существуют различия между программой, транслированной как EXE-файл, и программой, транслированной как СОМ-файл.

**Размер программы.** Программа в формате EXE может иметь любой размер, в то время как СОМ-файл ограничен размером одного сегмента и не превышает 64К. Размер СОМ-файла всегда меньше, чем размер соответствующего EXE-файла, так как в СОМ-файле отсутствует 512-байтовый заголовок EXE-файла.

**Сегмент стека.** В EXE-программе определяется сегмент стека, а в СОМ-программе стек автоматически указывает на конец сегмента кода. Таким образом, при создании ассемблерной программы, которая будет преобразована в СОМ-файл, стек должен быть опущен.

**Сегмент данных.** В EXE-программе сегмент данных обычно определяется как сегмент в отдельной области памяти. В СОМ-программе все сегментные регистры указывают на один общий сегмент.

**Инициализация.** В EXE-программе выполняется инициализация регистра DS. Когда СОМ-программа начинает работать, все сегментные регистры содержат адрес префикса программного сегмента (PSP) – 256-байтовый (100H) блок, который резервируется операционной системой

непосредственно перед COM- или EXE- программой в памяти. Так как адресация начинается со смещения 100H от начала PSP, то в программе после оператора SEGMENT используется директива ORG 100H.

**Обработка.** Для программы в EXE- и COM- форматах выполняются ассемблирование для получения OBJ-файла и компоновка для получения EXE- файла. Если программа создается для выполнения как EXE- файл, то ее уже можно выполнить. Если же программа создается для выполнения как COM-файл, то компоновщиком будет выдано сообщение: «Warning: No STACK Segment», т. е. предупреждение, что сегмент стека не определен. Это сообщение можно игнорировать, так как определение стека в программе не предполагалось. Для преобразования EXE-файла в COM-файл используется программа EXE2BIN. Предположим, что EXE2BIN имеется на диске C, а скомпонованный файл по имени CALC.EXE - на диске D. Введите:

```
EXE2BIN D:CALC,D:CALC.COM.
```

Так как первый операнд всегда предполагает EXE-файл, то можно не кодировать тип EXE. После того как преобразование будет выполнено, можно удалить OBJ- и EXE-файлы.

**Пример COM-программы.** Программа EXCOM1, приведенная на рис. 3.1, аналогична программе, рассмотренной в лабораторной работе 2 (см. рис. 2.3), но изменена согласно требованиям COM-формата. Обратите внимание на следующие изменения:

стек и сегмент данных отсутствуют;

оператор ASSUME указывает ассемблеру на необходимость установить относительные адреса с начала сегмента кодов. Регистр CS также содержит этот адрес, являющийся к тому же адресом префикса программного сегмента (PSP). Директива ORG служит для резервирования 100H байт от начального адреса под PSP;

директива ORG 100H устанавливает относительный адрес для начала выполнения программы. Программный загрузчик устанавливает этот адрес в командном указателе;

команда JMP служит для обхода данных, определенных в программе.

```
CODESG    SEGMENT
           ASSUME CS:CODESG,DS:CODESG,SS:CODESG,ES:CODESG
           ORG      100H
BEGIN:    JMP      MAIN
```

;------

FLDA	DW	250
FLDB	DW	125
FLDC	DW	?

-----

MAIN	PROC	NEAR
	MOV	AX,FLDA
	ADD	AX,FLDB
	MOV	FLDC,AX
	RET	
MAIN	ENDP	
CODESG	ENDS	
	END	BEGIN

Рис. 3.1. Пример COM-программы

Для обработки и выполнения этой программы необходимо выполнить следующие шаги:

```
TASM [имя файла.asm]
TLINK [имя файла.obj]
EXE2BIN [имя файла.exe], [имя файла.com]
DEL [имя файла.obj], [имя файла.exe],
```

или

```
TASM [имя файла.asm]
TLINK /t [имя файла.obj].
```

Размеры EXE- и COM-программ – 788 и 20 байт соответственно.

Учитывая такую эффективность COM-файлов, рекомендуется все небольшие программы создавать в COM-формате.

**Стек для COM-программы.** Для COM-файла операционная система автоматически определяет стек и устанавливает одинаковый общий сегментный адрес во всех четырех сегментных регистрах. Если для программы размер сегмента в 64 килобайта достаточен, то операционная система устанавливает в регистр SP адрес конца сегмента - FFFEH. Это будет вершина стека. Если 64-килобайтный сегмент не имеет достаточно места для стека, то операционная система устанавливает стек в конце доступной оперативной памяти. В обоих случаях операционная система записывает в стек нулевое слово.

Возможность использования стека зависит от размера программы и ограниченности памяти. С помощью команды DIR можно определить размер файла и вычислить необходимое пространство для стека.

Особенность отладки. Несоблюдение хотя бы одного требования COM-формата может послужить причиной неправильной работы программы. Если EXE2BIN обнаруживает ошибку, то выдается сообщение о невозможности преобразования файла без указания конкретной причины. Необходимо проверить в этом случае директивы SEGMENT, ASSUME и END. Если пропущена директива ORG 100H, то на данные в префиксе программного сегмента будут установлены неправильные ссылки с непредсказуемым результатом при выполнении.

### **3.2. Порядок выполнения работы.**

1. Составьте ассемблерную программу выполняющую:
  - a) определение однобайтовых элементов (директива DB) по имени FLDA, содержащего 28H, и по имени FLDB, содержащего порядковый номер фамилии студента в журнале;
  - b) определение двухбайтового элемента (директива DW) по имени FLDC, не имеющего значения;
  - c) пересылку содержимого поля FLDA в регистр AL и сдвига содержимого этого регистра на один бит влево;
  - d) умножение содержимого регистра AL на значение в поле FLDB;
  - e) пересылку результата из регистра AX в поле FLDC.
2. Измените эту программу для COM-формата, перетранслируйте ее и выполните под управлением отладчика DEBUG.