EXE и COM файлы. Выполнение программ и распределение памяти

После трансляции и редактирования программы сохраняются на диске как выполнимые файлы. Одна из основных задач ОС – организовать их выполнение.

Все программы рассматриваются как внешние команды ОС. Командная строка для выполнения программы имеет формат:

[устройство] [путь] имя [параметры командной строки]

Устройство и путь указывают место расположения программного файла, если они пропущены, то поиск осуществляется по имени файла в текущем каталоге. Программные файлы бывают двух типов: с расширением EXE и COM. Они различаются как по структуре, так и по способу их загрузки и выполнения. Фалы с расширением BAT также рассматриваются как внешние команды ОС.

При загрузке программы в ОП резидентный загрузчик командного процессора (COMMAND.COM) использует системную функцию 4Bh (EXEC). Принято называть в этом случае программу процессом. Для загружаемых программ COMMAND.COM является родительским процессом. Он поддерживает область памяти, называемую окружением. Окружение состоит из ASCII цепочек следующего формата:

Имя переменной = переменная Например: SET ABC = XYZ12

Каждая цепочка завершается байтом 00h, последняя цепочка окружения – двумя такими байтами. Команда SET управляет окружением, позволяя добавлять или устранять переменные или изменять их. При активизации COMMAND в окружение записывается переменная COMSPEC, значение которой – полная файловая спецификация командного процессора, например: COMSPEC = C:\ COMMAND.COM. Последние команды PATH и PROMPT также включены в окружение. При загрузке программы на выполнение она получает копию окружения родительского процесса. Адрес окружения записывается в одном из полей PSP - префикса сегмента программы (смещение 2Ch от начала PSP). Окружение каждого процесса, активированного COMMAND.COM статично и в процессе выполнения не изменяется.

EXE файлы

Создаются редактором связей LINK из obj – файлов и состоят из двух частей:

- **Префикс (заголовок);**
- **Выполнимый модуль.**

Префикс состоит из управляющей информации и таблицы перемещаемых символов. Во время выполнения программы адресные константы должны иметь абсолютные адреса. Преобразование относительных адресов в абсолютные выполняется системным загрузчиком, который использует информацию из таблицы перемещаемых символов в которой описано местоположение адресных констант в выполнимом модуле. Управляющая информация в префиксе состоит из полей длиной слово и содержит следующую информацию:

Поле	Смещение	Содержание
1	**00h**	**4D5Ah – идентификатор EXE – файла**
2	**02h**	**Длина файла по модулю 512 (остаток от деления длины на 512)**
3	**04h**	**Длина файла в блоках по 512 байт**
4	**06h**	**Количество элементов в таблице перемещаемых символов**
…	**….**	**….**
8	**0Eh**	**Смещение SS в выполнимом модуле (в параграфах)**
9	**10h**	**Значение SP при получении управления выполнимым модулем**
		…
11	**14h**	**Значение IP при получении управления выполнимым модулем**
12	**16h**	**Смещение CS и выполнимом модуле (в параграфах)**
13	**18h**	**Начало таблицы перемещаемых символов (смещение от начала файла в байтах)**

Загрузка программы (EXE – файла)

1. **Управляющая информация из префикса (заголовка) считывается в рабочую область ОС.**
2. **Определяется размер выполнимого модуля в зависимости от полей 2, 3, 5, 6.**
3. **Находится первый свободный блок памяти, размер которого достаточен.**
4. **Строится PSP – префикс сегмента программы. Сегмент после PSP называется стартовым сегментом. Выполнимый модуль загружается с начала этого сегмента.**
5. **Обрабатывается таблица перемещаемых символов с целью настройки адресных констант в соответствии с адресом загрузки (стартовым сегментом).**
6. **Регистрам DS, ES, FS, GS присваивается сегментный адрес PSP, SS и SP присваиваются соответствующие значения из полей 8 и 9 префикса и к SS добавляется адрес стартового сегмента. CS присваивается сумма поля 12 префикса и адреса стартового сегмента. IP загружается из поля 11 префикса EXE файла.**
7. **На этом этапе выполнимый модуль настроен в соответствии с адресом загрузки и управление передаётся по адресу CS:IP.**

Префикс сегмента программы PSP

PSP является важнейшим информационным блоком для каждого активного процесса. Когда программа загружается на выполнение, ей выделяется вся необходимая память от самого младшего свободного байта. Эта область называется сегментом программы. Первые 256 байт отводятся под PSP, куда ОС помещает разнородную информацию (системную и для программы). Сама программа загружается непосредственно после PSP. Основные поля PSP:

Поле	Смещение (16)	Дли-на	Содержимое
1	0	2	Инструкция INT 20h (завершение программы)
2	2	2	Размер памяти в блоках по 16 байт (в параграфах)
…	…	…	…..
9	2C	2	Адрес области окружения (сегментный, смещение 0)
…			
13	55	7	Префикс первого FCB
14	5C	9	Первый FCB
…			
17	80	1	Длина параметров командной строки
18	81	127	Параметры командной строки

COM - файлы

Не имеют префикса (заголовка). Они строятся так, чтобы не содержать адресных констант, зависимых от адреса загрузки программы в память. COM – файлы состоят из одного сегмента, в котором определены коды, данные и стек. Их размер не превышает 64Кб. COM файл представляет собой точную копию программы в двоичном виде, в каком её нужно загрузить в память. Поэтому загрузка сводится к определению свободного блока памяти, построению PSP и размещение всего файла в область после PSP. В MASM при построении COM файлов используется модель памяти TINY. При загрузке регистры получают следующие значения:

CS, DS, SS, ES, FS, GS – адрес PSP;

IP = 100h, SP = FFFEh

Примечание. При загрузке COM и EXE файлов программе выделяется ещё один блок памяти в который помещается копия текущего окружения COMMAND.

Обобщенная структура com. файла

```
prog1      segment   para ‘code’
           assume  cs: prog1, ds: prog1,  ss: prog1,  es: prog1
           org   100h
  start:       jmp  M1
           A      dw  ?
           B      db  ‘Пример  com.’
    old_N   dd   ?
           …
M1:   mov   al,  ES: [80h]
           …
           …
           int  20h          ;     mov  ax,   4c00h       int  21h
Stack1        dw   100 dup(?)
prog1     ends
           end  start
```