

Лекция №4. РАБОТА С ПАМЯТЬЮ В РЕАЛЬНОМ РЕЖИМЕ И ОСОБЕННОСТИ защищенного режима

Распределение памяти в реальном режиме x86 процессора.

Каждому блоку памяти предшествует 16 байт блок управления памятью (МСВ), что определяет владельца блока, размер блока, и, если блок содержит программу, имя программы. Блок, владельца, свободен и может быть назначен программе. МСВ образует цепочку, определяет структуру памяти. если установлено местонахождение главного МСВ, можно найти блок, расположенный в более старших адресах памяти, добавив к сегментного адреса этого МСВ длину блока памяти, следующего за МСВ и еще 1 (1 добавляется том, что длины блоков памяти выражаются в пунктах и сам МСВ занимает 1 пункт).

Каждый МСВ содержит 4 поля. Байт со смещением 0 (байт сигнатуры) определяет, является этот МСВ последним в цепочке. Размер 4Dh (N) означает, что за этим МСВ следует другой; 5Ah (Z) - блок последний в цепочке. Слово со смещением 1 определяет владельца блока памяти, следующего за МСВ. Оно содержит сегментный адрес программы, для которого выделен этот блок памяти. Размер 0 означает, что блок на данный момент не выделен. Слово со смещением 3 устанавливает в байтах длину последнего блока памяти. Если блок памяти содержит программу, ее имя появляется в МСВ со смещением 8. Имя сохраняется в формате ASCII и заканчивается нулем, если его длина менее 8 символов. Когда в Config.sys включена команда DOS = UMB, система создает цепочка МСВ в верхней памяти.

По умолчанию эта цепочка не связан с цепочкой МСВ в обычной памяти. Блоки памяти, резервируются для операционной системы, имеют значение 08h в поле идентификатора владельца. Байты 8 и 9 в соответствующих МСВ содержат буквы SC и SD, что означает системный код и системные данные. Блоки, обозначенные SD, подразделяются на меньшие блоки. Этим меньшим блокам предшествуют свои суб-МСВ, формат которых идентичен формату действительных МСВ, но вместо M и Z байт сигнатуры содержит B, D, F, L, S, X. Каждая из этих букв идентифицирует содержимое блока памяти, следующего за Суб-МСВ. Буква B устанавливает, что блок содержит дисковые буфера, связанные с командой BUFFERS; D - указывает, что блок содержит драйвер устройству; F -

блок содержит системную таблицу файлов; L - блок хранит данные логического типа, S - блок содержит внутренние стеки системы; X - кэширует данные MCB.

Программам почти не приходится управлять блоками памяти на таком низком уровне. Чтобы облегчить для программ работу с памятью, система предоставляет такие системные функции:

1. 48h выделяет блок памяти.
2. 49h освобождает выделенный блок памяти.
3. 4Ah изменяет размер выделенного блока памяти.
4. 5800h и 5801h получают и устанавливают стратегию выделения памяти.
5. 5802h и 5803h получают и устанавливают состояние "привязки" верхней памяти.

Выделение блоков верхней памяти.

По умолчанию все блоки памяти, выделяемых функцией 48h, берутся из обычной памяти. Если система содержит свободные блоки верхней памяти, то используя функцию 48h можно выделить UMB. Сначала программа должна связать цепочки MCB в верхней и обычной памяти. Этот процесс называется привязкой верхней памяти. Затем программа должна изменить стратегию выделения памяти. Эти два действия могут быть выполнены с помощью функций 5803h и 5801h.

функция 5803h . Управляет связью между обычной и верхней памятью.

Вход:

AX - код функции

BX - код связи: 1 - верхняя и обычная память связываются,

0 - связь отменяется.

Выход: Флаг переноса сброшен - вызов завершен успешно; флаг переноса установлен - произошла ошибка. Функция завершается неудачно, если происходит попытка установить связь с верхней памятью, в то время как UMB нет. Например, если EMM386. EXE не загружена или загружен без ключей RAM или NOEMS. Она также потерпит неудачу, если UMB есть, в Config. sys отсутствует команда DOS = UMB.

Функция 5801h. Устанавливает стратегию, система должна использовать

при поиске свободных блоков в процессе удовлетворения запроса выделения памяти.

Вход :

- AX - код функции;
- BX - код стратегии выделения памяти (табл. 1).

Табл. 1 Коды стратегий

<u>КОД</u>	стратегия	описание
00h	Нижний-первый / поиск от начала	Поиск ведется сначала в обычной памяти, затем в верхней. Возвращается адрес первого найденного блока, объем которого больше или равен размеру передаваемого ф. 48h.
01h	Нижний-первый / оптимальный Поиск	Поиск сначала в обычной памяти, затем в верхней. Возвращается адрес блока, объем которого <u>более всего близок к размеру, переданного ф. 48h.</u>
02h	Нижний-первый / поиск от конца.	Поиск сначала в обычной памяти, затем в верхней. Возвращается адрес последнего блока, объем которого больше или равен размеру, переданного ф. 48h.
40h	Едва верхней / поиск от начала	Поиск только в верхней памяти. Возвращается адрес первого найденного блока, объем которого больше или равен размеру передаваемого ф. 48h.
41h	Только верхний / оптимальный Поиск	Поиск только в верхней памяти. Возвращается адрес блока, объем которого близок к размеру, переданного ф. 48h.
42h	Только верхний / поиск от конца	Поиск только в верхней памяти.
80h	Верхний-первый / <u>поиск от начала</u>	Поиск сначала в верхней, затем в обычной памяти.
81h	Верхний-первый / оптимальный Поиск	Сначала в верхней, затем в обычной памяти.

После того, как связь с верхней памятью и стратегия выделения памяти установлена, функция 48h может быть использована для назначения свободных UMB.

Освобождение блоков верхней памяти.

Программа освобождает UMB тем же средством, что и блок обычной памяти: вызывая функцию 49h с сегментной адресу блока в регистре ES. Чтобы UMB мог быть освобожден, должна быть установлена связь с верхней памятью. В противном случае функция 49h вернет установлен флаг переноса и UMB останется выделенным.

Хранения и восстановления стратегии выделения памяти и состояния связи.

Всякий раз, когда программа изменяет состояние связи с верхней памятью, ей стоит сначала сохранить текущее состояние связи и восстановить его по завершении. точно также, перед тем, как изменить стратегию системы выделения памяти, программе стоит сохранить текущую стратегию и затем восстановить ее перед завершением. для определения внутри программы текущей стратегии выделения памяти и состояния связи системы предоставляет функции 5800h и 5802h. Функция 5800h возвращает код текущей стратегии в AX.

Аналогично, функция 5802h возвращает 0 AL, если связь с верхней памятью не установлен; в противном случае она возвращает 1. Восстановление состояния связи с верхней памятью выполняется функцией 5803

Пример назначения UMB ...

```
mov ax, 5800h; сохраняем эту
int 21h          ; стратегию
mov strategy, ax; выделение
mov ax, 5804h; храним текущую
int 21h          ; привязку верхней памяти
mov byte ptr linkstate, al
mov ax, 5803h; установления связи с верхней
mov bx, 1        ; памятью
int 21h
mov ax, 5801h; переключаемся
mov bx, 40h, на стратегию
int 21h; "только верн. / поиск с нуля"
mov ah, 48h; выделяет блок памяти
mov bx, 400h; объемом 16к
int 21h
jc error
mov addr, ax; сохр. сегм. адрес
mov ax, 5803h; восстановление состояния
mov bx, linkstate; связи с верх. п.
int 21h
mov ax, 5801h; восстановления стратегии
mov bx, strategy; выделения.
int 21h
...
```

Написана таким образом программа оставляет нетронутыми внутренние флаги, контролируют метод выделения памяти, используемой системой.

Управление памятью.

карты памяти . Память ПК состоит из множества интегральных микросхем (Чипов). Для программ неважно физическое расположение чипов, но важно, в каком логическом пространстве расположена память ПК. Этот логический пространство называется адресным пространством памяти.

Карта памяти - это диаграмма распределения хранимой информации. На ней также указаны места, что в принципе могут использоваться для хранения данных, хотя и не предназначены для этого.

Средства представления карты памяти:

- Наименьшая адрес - вверху (по аналогии с книгой).
- Наименьшая адрес - снизу, так как программы размещаются от нижних адресов

памяти до верхних.

Память в реальном режиме.

Под памятью в реальном режиме понимается область памяти с адресами от 0 до 1 Мбайта (1024 Кб). IBM приняла несколько решений об использовании первого мегабайта пространства, адресуется: первые 5/8 первого мегабайта используются для операционной системы и для любой другой программы, выполняется. Это логические страницы от 0 до 9. Страницы A и B зарезервированы для видео-RAM. Страницы C и D были зарезервированы для дальнейшего применения. Страницы E и F - для системной ROM (постоянной памяти). Эти решения привели к появлению барьера в 640 Кбайт.

Реальный режим называется так потому, что является операционным режимом семейства процессоров 80x86 в момент подачи на них напряжения или после перезагрузки системы.

В реальном режиме нет различия между линейными и физическими адресами.

(Линейная адрес = сегмент * 16 + смещение). Максимальная физическая адрес = 1 Мб.