Лекция №7. Многозадачном режиме И страничный

ОРГАНИЗАЦИЯ ПАМЯТИ

Организация многозадачного режима.

Переключение задач.

Использование таблиц локальных дескрипторов позволяет разнести физические адресные пространства отдельных задач без права их обращение к "чужой" памяти. система привилегий, которая входит в архитектуру процессора, предотвращает несанкционированные вызовам задачами защищенных процедур.

В простых случаях параллельное выполнение нескольких задач осуществляется на одном уровне привилегий.

Защита адресных пространств или процедур осуществляется путем тщательного написание программ комплекса.

Организация многозадачного режима основывается на следующих аппаратных и программных средствах:

- сегмент состояния задачи (TSS-Task State Segment)
- дескриптор сегмента состояния задачи;
- регистр задачи (Task Register, TR)
- дескриптор шлюза задачи (Task gate).

Сегмент состояния задачи (TSS) - это поле данных. Оно может быть включено в состав сегмента данных, или образовывать отдельный сегмент небольшого размера. каждая задача, которая участвует в процессе переключения, должна иметь свой TSS.

Поскольку TSS представляется процессору отдельным сегментом, ему должен отвечать дескриптор. TSS описывается системным дескриптором, который может находиться только в GDT.

Формат системного дескриптору (рис. 1) почти одинаковый с форматом дескриптору памяти. Отсутствует бит замалчивания D і код дескриптора - 0, а не 1.

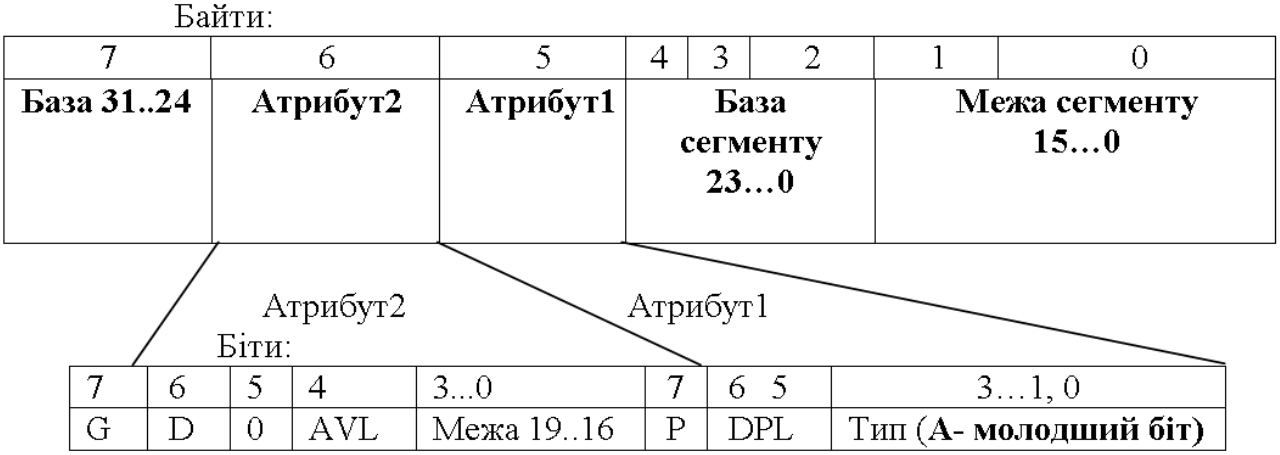

Рис. 1 Формат системного дескриптору

Табл. 1 Тип системного дескриптору может принимать несколько значений

Тип	назначение дескриптору
0	Не определен 1
	Свободный сегмент состояния задачи (TSS) 80286 2
	LDT 3
	Занят сегмент состояния задачи (TSS) 80286 8
	Не определен 9
	Свободный сегмент состояния задачи (TSS) 80386 ... Ah
	Не определен Bh
	Занят сегмент состояния задачи (TSS) 80386 ...
Dh	не определен

В зависимости от порядкового номера дескриптору TSS в таблице дескрипторов, ему соответствует тот или иной селектор. Селектор TSS активной задачи должен быть загружен в регистр задачи TR. Для главной задачи это загрузка осуществляется программно с помощью команды ltr (load task register). При переключении на новую задачу программа передает процессору селектор нового TSS, и перегрузки регистра TR процессор осуществляет в ходе переключения задач.

Переключение на новую задачу осуществляется командой дальнего вызова call dword ptr или дальнего перехода jmp dword ptr. В качестве аргумента этих команд указывается двосливне поле, в первом слове которого записывается селектор нужного TSS. Существует и другой способ переключения - не через селектор TSS, а через шлюз задачи (Дескриптор шлюза с кодом типа 5).

В этом случае селектор нужного TSS указывается в поле для селектора в шлюзе задачи. Переключение через шлюз задачи имеет то преимущество, что его можно

осуществить аппаратным прерыванием (шлюз задачи можно разместить в таблице дескрипторов прерываний).

В процессе переключения задач процессор сохраняет контекст текущей задачи в TSS и загружает новый контекст с TSS новой задачи (включая селектор и относительную адрес точки входа в задачу). Контекст задачи: EAX, EBX ..., SS: ESP, CS: EIP (адрес наступноі команды).

Задача, на которую осуществляется переключение, должна завершаться командой iret, которая обрабатывается процессором не так, как iret обычного обработчика прерывания. В случае переключения задач процессор по команде iret выполняет обратное перемещения контекстов. Контекст завершенной задачи сохраняется в ее TSS, а в регистры процессора загружается с TSS входящей задачи сохранен там контекст, соответствующий момента переключения на другую задачу. Таким образом, TSS можно рассматривать как функциональный аналог стеке. Но хранения в стеке и восстановления из стека осуществляется с помощью последовательных команд процессора, а хранение и восстановление контекстов с помощью TSS осуществляется процессором аппаратно в процессе переключения задач.

Поля TSS входящей задачи заполняются процессором при первом переключении на новую задачу. Программист может не заботиться об их содержание. Поля TSS задачи, на которую происходит переключение, содержащие адрес точки входа и входной содержание сегментных регистров и регистров общего назначения. Эти поля должны быть заполнены во входной задачи перед переключением на новую.

Содержание TSS. (Размер TSS минимум 104 байта). В начале TSS расположен 16-битное поле связи, которое используется при переключении задач. Для входящей задачи его содержание не имеет значения.

При переключении на новую задачу процессор заносит в поле связи TSS новой задачи селектор TSS входящей задачи. В результате создается связный список вложенных задач (рис. 2).

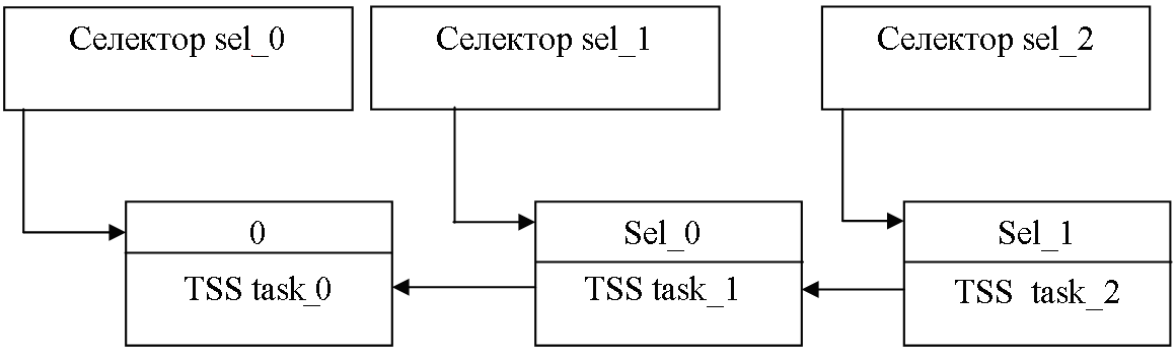

Рис. 2 Связный список вложенных задач

Табл. 4.2 Формат сегмента состояния задачи (TSS)

Смещение от базы TSS 0			
	связь		0h
ESP0			04h - кадр стека уровня привілеів 0
0	SS0		08h
ESP1			0ch - кадр стека уровня привілеів 1
0	SS1		10h
ESP2			14h - кадр стека уровня привілеів 2
0	SS2		18h
CR3			1ch
EIP			20h
EFLAGS			24h
EAX			28h
ECX			2ch
EDX			30h
EBX			34h
ESP			38h
EBP			3ch
ESI			40h
EDI			44h
0	ES		48h
0	CS		4ch
0	SS		50h
0	DS		54h
0	FS		58h
0	GS		5ch
0	LDT		60h
Адрес карты в / в	000 ... 000	T	64h
Карта в / в (если она есть)			68h

Команда iret, которой завершается каждая вложенная задача, выполняет обратное переключения задач. В ходе этой операции процессор извлекает из поля связи TSS текущей задачи селектор TSS предыдущей (по порядку вложенности), восстанавливает из TSS ее контекст и передает ей управление.

По адресам 04h, 0Ch, 14h относительно базы TSS располагаются кадры стеков уровней привилегий 0,1,2. Содержание этих полей загружается в регистры SS и ESP, если при переключении задачи происходит изменение уровней привилегий.

По адресу 1Ch содержится поле регистра управления CR3. Он содержит базовую адрес каталога страницы и используется, если включен страничное преобразования. Наличие в TSS этого поля позволяет иметь для каждой задачи свой каталог страниц и свои таблицы отображения виртуальных страниц программы на физические адреса памяти.

Двосливне поле TSS по адресу 20h служит для сохранения значения указателя команд EIP. В TSS входящей задачи это поле процессор заполняет адресу возвращения. В TSS задачи, на которую происходит переключение, поле для EIP должно быть заполнено программно смещением точки входа в задачу. При возвращении в старую задачу командой iret процессор записывает в поле EIP задачи, завершившейся адрес команды, которая идет по iret, как адрес возврата. То есть адрес уже за пределами задачи. Если планируется повторное переключение на эту задачу, то перед каждым переключением необходимо восстанавливать в ее TSS адрес точки входа.

По адресу 24h в TSS сохраняется текущее содержимое регистра флагов EFLAGS. Это позволяет осуществлять переключение задач в любой точке задачи без потери ее работоспособности.

Часть TSS с адресами 28h -47h отведена для сохранения регистров общего назначения. При переключении с входящей задачи на новую в этих полях TSS входящей задачи сохраняется текущее содержимое регистров, при обратном переключении командой iret восстанавливается их содержимое из этих полей. Заполнив ранее поля регистров в TSS новой задачи, можно передать ей входные параметры.

Младшие половины шести двосливних полей, начиная с адреса 48h отведенные под содержимое сегментных регистров.

Если новая задача использует LDT, ее селектор нужно занести в TSS по адресу 60h.

Бит 0 слова по адресу 64h используется для отладки задач, переключаются. Если в TSS новой задачи установлен этот бит, то сразу же после переключения генерируется исключение отладки с номером 1. Другие биты слова по адресу 64h в TSS должны равняться 0.

Слово по адресу 66h содержит смещение битовой карты в \ у, которая, если она есть, располагается по следующим адресам и используется для защиты портов компьютеру от НСД. Каждый бит этой карты соответствует одному порту (64 Кбитів или 8 Кбайтів).

Если бит, закрепленный за некоторым портом, равен 1, то при обращении к порту задачей с недостаточно высоким уровнем привилегий генерируется исключение общей защиты. Если он = 0, то задача с любым уровнем привилегий может обращаться к порту.

В бите 14 регистра флагов NT (Nested Task, вложенная задача) устанавливается режим выполнения команды iret. Команда iret анализирует состояние флага NT и если он сброшен, осуществляет обычное возвращения из программы обработки прерывания (через стек). Если флаг NT установлен, то iret инициирует обратное переключение задач через селектор в TSS.

После загрузки компьютера флаг NT установлен. Но любое программное, аппаратное прерывание или исключение сбрасывает этот флаг. после завершения обработчика NT снова установлен (iret восстанавливает входной содержание регистра флагов).

При переключении задач через шлюз задачи или через TSS, процессор сохраняет в TSS текущей задачи слово флагов и устанавливает в регистре флагов бит NT.

Дескрипторы-псевдонимы.

Во многих случаях, работая в защищенном режиме, нужно расширить права программы при обращении к сегментам. В тех случаях, когда требуются характеристики, не предусмотренные среди стандартных типов дескрипторов, расширение прав доступа к сегменту осуществляется с помощью описания для того же самого сегмента нового дескриптору с другими правами доступа. Например, если нужно не только выполнять строки сегмента команд, но и читать их, то дескриптору этого сегмента нужно присвоить тип 5. Если же нужно сегмент еще и модифицировать, то его нужно описать двумя дескрипторами: с типом 5 и типом 1.

Такой дополнительный дескриптор, расширяющий права доступа к сегменту, называется дескриптором-псевдонимом.

Процессор запрещает обращение к сегментам состояния задач с целью их чтения или модификации. Селектор дескрипторов TSS можно использовать только для переключения задач. Скачать такой селектор в сегментный регистр нельзя. для того, чтобы в защищенном режиме выполнить настройку сегмента состояния задачи, для него необходимо создать дескриптор-псевдоним с атрибутом разрешения читання-записи. Загрузив в какой-нибудь сегментный регистр селектор этого дескриптору, можно обращаться к этому сегменту как к обычному сегмента данных.