

Мультизадачность – способность ЭВМ выполнять несколько задач одновременно.

В общем определении под задачей понимается последовательность взаимосвязанных действий, ведущих к достижению некоторой цели.

В узком смысле задача определяется как совокупность кода и данных, которым назначен сегмент состояния задачи (Task State Segment - TSS). Имеется однозначное соответствие между сегментами TSS и задачами.

Контекст задачи – состояние процессора на момент завершения текущего кванта времени решения задачи.

Переключение контекста – действия процессора по сохранению состояния одной задачи и восстановлению состояния другой задачи.

Для поддержки мультизадачности используются:

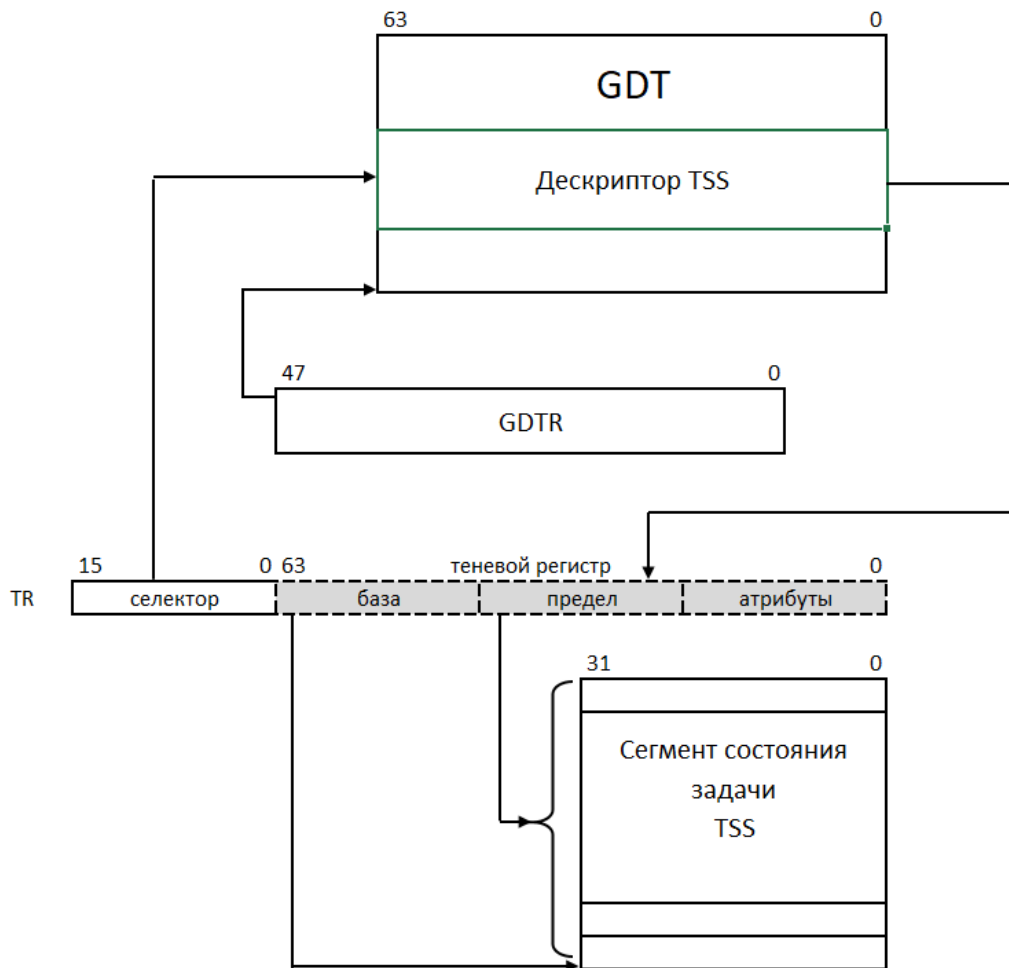
1. Регистр задачи (Task Register – TR)
2. Сегмент состояния задачи (Task State Segment)
3. Дескриптор TSS
4. Дескриптор шлюза задачи;

Регистр TR.

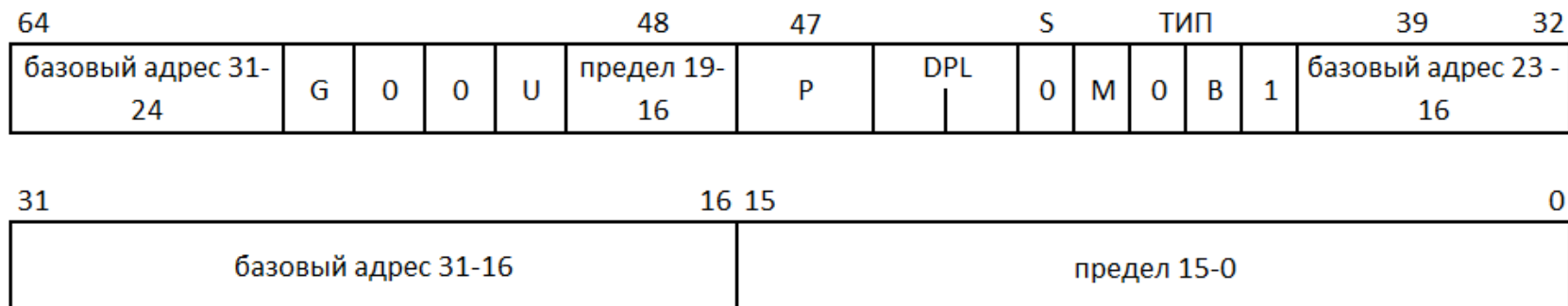
TR (Task Register) Определяет косвенно через GDT (Global Descriptor Table, глобальная таблица дескрипторов) местоположение сегмента состояния задачи TSS (Task Segment State).

Каждая задача имеет свой TSS.

TSS предназначен для сохранения контекста задачи.



Формат дескриптора TSS (может находиться только в GDT)



S = 0 – системный объект.

M – тип процессора (0 – 16-ти разрядный для 286 процессора, 1 – 32-х разрядный для 386 процессора).

B – бит занятости (отмечает выполняющуюся задачу или вложенную).

P – бит присутствия.

DPL – допустимый уровень привилегий программ, которые могут обращаться к задаче, определяемой данным TSS.

G – бит гранулярности.

Базовый предел 31-0 – определяет начало TSS.

Предел 19-0 – определяет размер TSS.

Когда бит гранулярности G сброшен, предел задается 20-битным значением в байтах и варьируется от 00000 до FFFFFh (1Мбайт). Когда бит гранулярности G установлен, процессор сдвигает 20-битное значение предела на 12 бит влево. В этом случае предел сегмента задается в 4Кбайтных страницах и варьируется от FFFh (предел равен 0) до FFFFFFFFh (предел равен FFFFFh).

Типы дескрипторов TSS:

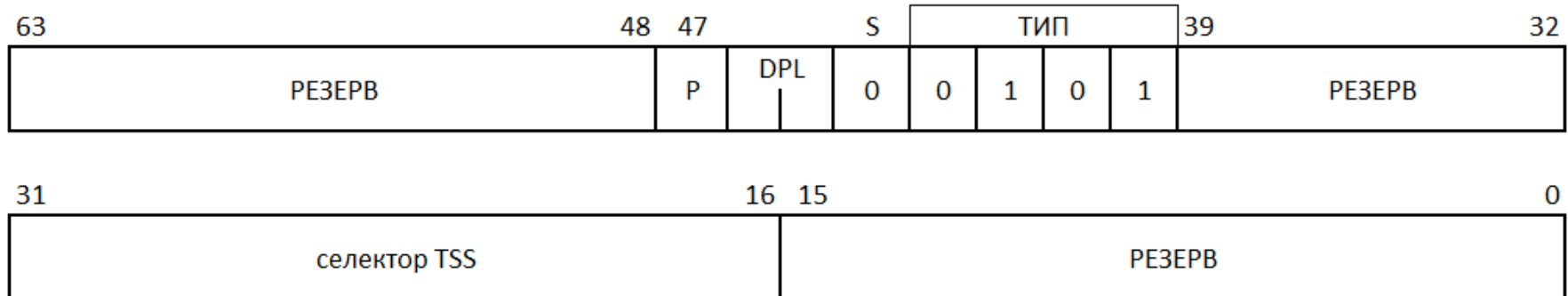
0001B = 1h – свободный TSS 286 процессора.

0011B = 3h – занятый TSS 286 процессора.

1001B = 9h – свободный TSS 386 процессора.

1011B = Bh – занятый TSS 386 процессора.

Формат шлюза задачи.

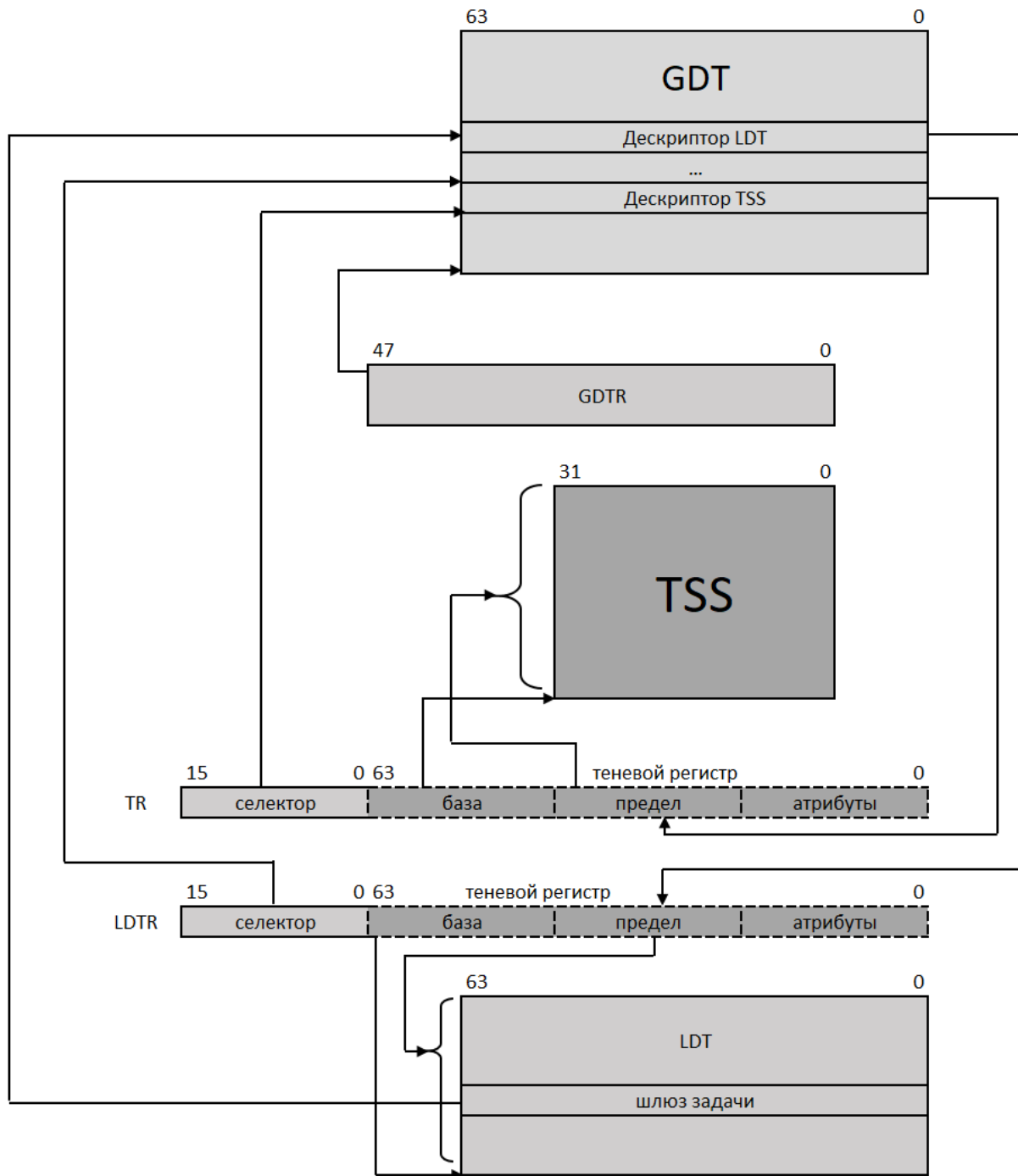


Действует как посредник в процессе переключения задачи.

Для чего нужны шлюзы задачи?


Дескриптор TSS может находиться только в GDT (Global Descriptor Table). Доступ к нему имеют только привилегированные задачи.

Если для данного TSS определить несколько шлюзов задачи и разместить их в различных LDT (Local Descriptor Table), то доступ к задаче получают те задачи, которые содержат шлюз задачи в своей LDT.



Формат 32-х битного сегмента TSS.

	31	16	15	0	
база двоичной карты расширения ввода/вывода	системно-зависимая часть				69h
	→		0	T	64h
	0	LDTR			60h
	0	GS			5ch
	0	FS			58h
	0	DS			54h
	0	SS			50h
	0	CS			4ch
	0	ES			48h
EDI				44h	
ESI				40h	
EBP				3ch	
ESP				38h	
EBX				34h	
EDX				30h	
ECX				2ch	
EAX				28h	
EFLAGS				24h	
EIP				20h	
CR3				1ch	
0		SS2		18h	
ESP2				14h	
0		SS1		10h	
ESP1				ch	
0		SS0		8h	
ESP0				4h	
0		обратная связь		0h	

 - статическая часть TSS

Поле обратной связи содержит селектор TSS задачи предка.
Т – бит ловушки.

Системно-зависимая часть TSS.

Может содержать полное состояние FPU, список открытых файлов, двоичную карту ввода/вывода и т.д. Системно-зависимая часть – это часть TSS сохраняется и восстанавливается программно.

Двоичная карта ввода/вывода.

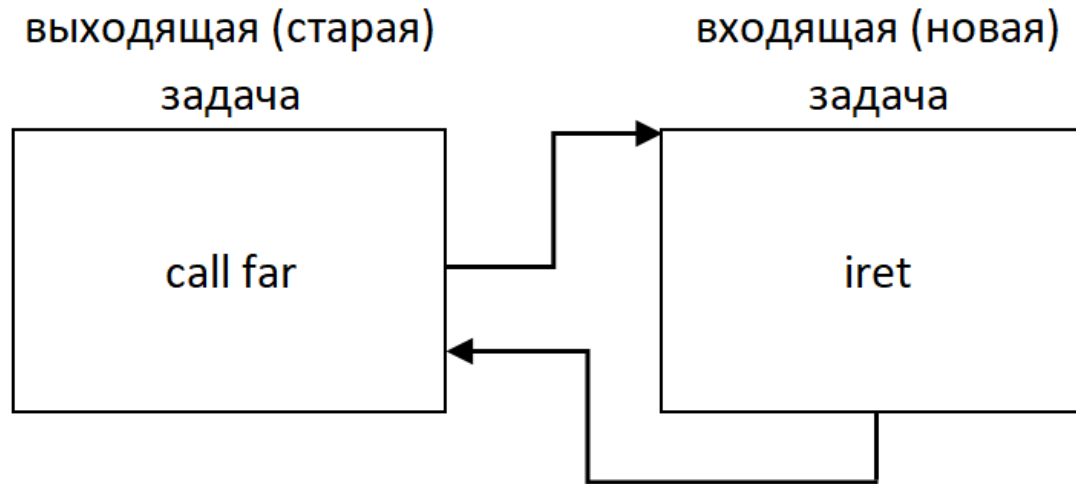
Каждому порту ввода/вывода из пространства 64к, соответствует один бит карты ввода/вывода. Сброшенный бит карты ввода/вывода разрешает обращение задачи к соответствующему порту ввода/вывода. Установленный бит – запрещает.

Если база двоичной карты ввода/вывода равна нулю, то карта отсутствует. В этом случае текущей задаче разрешается обращение к портам ввода/вывода, если $CR4 \leq IOPL$ в EFLAGS.

31	16 15	8 7	0
...	FFh	FFh	11000000B
FFh	FFh	FFh	FFh

Разрешено обращение к портам 32, 33, 34, 35, 36, 37.

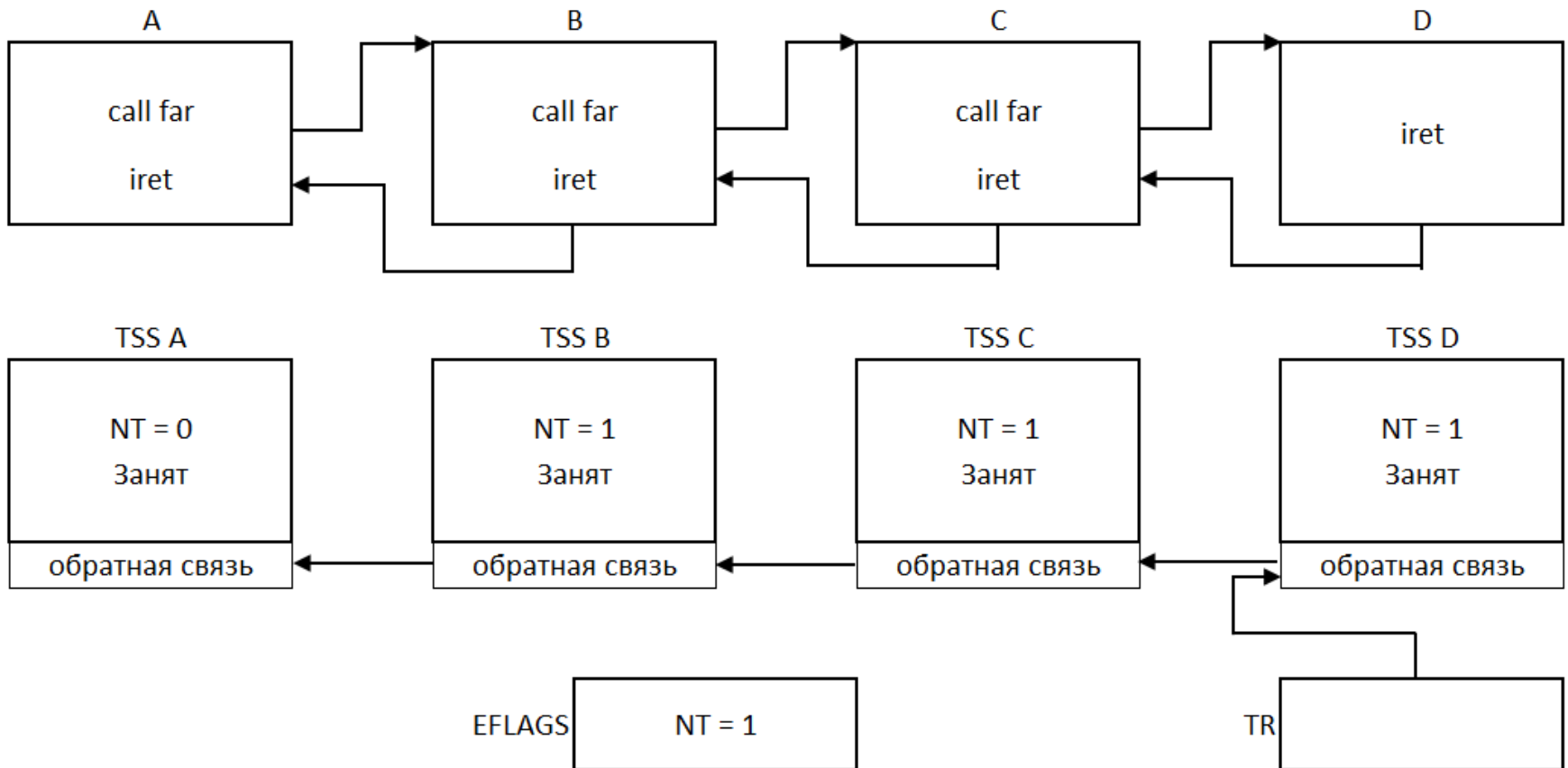
Переключение задачи.



- Выходящая задача выполняет команду CALL FAR или JUMP FAR и селектор выбирает дескриптор TSS;
- Выходящая задача выполняет команду CALL FAR или JUMP FAR и селектор выбирает шлюз задачи;
- Входящая задача выполняет команду IRET и бит NT = 1 в регистре EFLAGS;
- Возникло аппаратное или программное прерывание и соответствующий элемент IDT (Interrupt Descriptor Table) содержит шлюз задачи.

Задачи изолированы друг от друга. Входящая не получает от выходящей никаких параметров и не возвращает результат.

Вложенные задачи.



NT (Nested Task) в EFLAGS – указывает, что текущая задача является вложенной. По команде `iret` происходит переключение на задачу, на которую указывает поле обратной связи в TSS.

Глубина вложений задач не ограничена.

Действия процессора при переключении задач.

1. Если переключение задач не вызвано аппаратным прерыванием, особым случаем или командой `iret`, производится контроль привилегий дескриптора.
2. Проверяется бит присутствия и предел дескриптора TSS выходящей задачи. При положительном исходе сохраняется контекст выходящей задачи. На TSS указывает TR.
3. Проверяется бит присутствия и предел дескриптора TSS входящей задачи. При положительном исходе восстанавливается контекст входящей задачи. Если меняется CR3, очищается буфер TLB. Регистры процессора загружаются без загрузки теневых регистров.
4. Устанавливается связь с выходящей задачей:
 - если `JUMP FAR`, то дескриптор TSS входящей не отмечается занятым, а дескриптор TSS входящей задачи отмечается как занятый;
 - если прерывание или `CALL FAR`, то дескрипторы TSS входящей задачи отмечаются как занятые, бит NT в EFLAGS устанавливается и поле обратной связи TSS входящей задачи помещается селектор выходящей задачи;
 - если `IRET`, то выходящая задача отмечается как незанятая.
5. В регистр TR загружается селектор TSS новой задачи.
6. Бит TS (Task Switched) в CR0 устанавливается, и текущий уровень привилегий берется из поля RPL селектора CS в сегменте TSS.
7. Загружается теневой регистр для LDTR, если LDTR содержит правильный селектор.
8. В теневые регистры CS, SS, DS, ES, FS и GS загружаются необходимые дескрипторы. Все дескрипторы проверяются по правилу привилегий и должны быть присутствующими.
9. Если в TSS входящей задачи бит ловушки T = 1 фиксируется особый случай отладки.
10. Начинается выполнение новой задачи посредством выборки команды, адресуемой CS:EIP.

