Файл Теория

Специальные регистры и их роль: *eip* – счётчик адреса, *cs*, *ds*, *es*, *ss*, *fs*, *gs* – сегментные регистры, регистры *gdtr*, *ldtr*, *idtr* – указатели на таблицы, *tr* – регистр задачи.

eip - регистр что хранит адрес на текующую исполняемую команду, единственный способ достать значение через call

сегментные регистры (дексриптор - описатель): В архитектуре x86 сегментные регистры (CS, DS, ES, SS, FS, GS) используются для адресации памяти соблюдения уровней привелегий и защиты сегментов и для адресации. Они хранят селекторы сегментов — указатели на структуры данных в памяти (глобальную или локальную таблицу дескрипторов), которые описывают характеристики сегмента (базовый адрес, размер, права доступа и т.д.). Селектором называется 16-битовое значение, первые 13 бит из них являются индексом дескриптора в одной из двух таблиц дескрипторов (таблица – это массив дескрипторов). Ещё один бит-индикатор (Table-Indicator) указывает, в какой таблице дескрипторов (глобальной или локальной gdt ldt) находится дескриптор. Последние два бита задают уровень привилегий данного селектора. Дескриптор описывает какой-либо важный объект системы (сегмент, шлюз прерывания, задачу (процесс), локальную таблицу дескрипторов и т.д.).
Дескрипторы сегментов хранятся в специальных таблицах, каждый такой дескриптор описывает конкретный сегмент и содержит:

1. адрес начала этого сегмента в оперативной памяти,
2. максимальную длину (предел) сегмента,
3. права доступа (разрешено ли чтение, запись и выполнение команд),
4. уровень необходимых привилегий для работы с сегментом
5. и некоторые другие атрибуты.

у каждого сегментоного регистра есть теневой регистр для хранения дескриптора, для быстроты, но с ними работать напрямую нельзя, они теневые, дескрпитор сегмента можно достать используя индекс (первые 13 бит) и указатель на необходимую табличку ldtr gdtr (указатели на локальную и глобальную таблицы дескрпиторов)

1. CS (Code Segment / Сегмент кода): Содержит селектор сегмента, в котором находятся исполняемые инструкции программы. Указатель инструкций IP/EIP/RIP содержит смещение внутри этого сегмента. запись запрещена для пользователя. Сегмент изменяется инструкциями перехода (дальние CALL, JMP, RET, прерывания, исключения).
2. DS (Data Segment / Сегмент данных): Содержит селектор сегмента по умолчанию для большинства операций с данными (чтение/запись). Используется, когда в инструкции явно не указан другой сегментный регистр.
3. ES (Extra Segment / Дополнительный сегмент): Исторически был "дополнительным"сегментом данных. Часто используется как регистр-получатель в строковых операциях (MOVS, STOS, CMPS и т.д.) вместе с DS (источник) или для доступа к специальным структурам данных.
4. SS (Stack Segment / Сегмент стека): Содержит селектор сегмента, в котором находится стек программы. Указатель стека ESP содержит смещение (вершину стека) внутри этого сегмента. Разрешает чтение/запись, но не исполнение. Изменяется при переключении задач/потоков.
5. FS (Extra Segment 2 / Дополнительный сегмент 2): указывает на (Thread Information Block), содержащий указатель на структуру TEB (Thread Environment Block) с критической информацией о текущем потоке (исключения, стек, локальное хранилище потока (TLS), ID потока, PID процесса и т.д.).
6. GS (Extra Segment 3 / Дополнительный сегмент 3): GS указывает на TEB текущего потока (в 64-битном режиме роль FS и GS меняется по сравнению с 32-битным).
7. TR (Task register): регистр задачи содержит селектор сегмента TSS (task status segment) используются осью для переключения между счётом задач

В современном ПО (особенно в 32/64-битном защищенном режиме) DS, ES, SS часто указывают на один и тот дескриптор с полными правами на данные.
FS GS тесно связаны с курсом ОС достаточно просто сдополнительный сегментный регистр
TR - тоже про ОСИ но стоит помнить мы запускаем программу она порождает задачи задачи - потоки вот в TR - селектор сегмента в котором сохраняется все текущие данные чтобы позднее восстановить, после перехода. Уровень привилегий тоже не проблема курса архитектура ЭВМ мы никчёмные у нас 3=CPL мы в ядро не лезем

Каждый номер прерывания определяет для процессора так называемый дескриптор (описатель) процедуры-обработчика прерывания с данным номером. Все такие дескрипторы (каждый длиной по 8 байт) располагаются в специальной таблице (массиве) дескрипторов прерываний с именем IDT (Interrupt Descriptor Table), максимальная длина этой таблицы 2048 байт. По существу, номер прерывания является индексом в массиве IDT. На начало IDT указывает системный регистр IDTR, так что эта таблица может располагаться в любом месте памяти, а не обязательно с нулевого адреса, как было в младших моделях первого и второго поколений процессоров семейства Intel. Для многоядерных ЭВМ у каждого процессорного ядра свой регистр IDTR и, соответственно, своя IDT.