# Вопросы по защищенному режиму:

1. Какую программу вы написали?  
   Написали управляющую программу с 0 уровнем привилегий.
2. Что пришлось создать в этой программе?  
   Две системные таблицы.
3. Где явно установлен соответствующий уровень привилегий?  
   2 бита (DPL) в первом атрибуте дескриптора сегмента.
4. Какие сегменты описали в глобальной таблице и почему, для чего?
   * 16-разрядный сегмент кода для реального режима;
   * 16-разрядный сегмент данных, для хранения данных (строки для вывода, например); (!\*!Мы выполняем нашу программу под управлением DOS значит нам не нужны 16и разрядные сегменты => удалить дескриптор для 16 разрядного)
   * 32-разрядный сегмент кода для защищенного режима;
   * 32-разрядный сегмент данных, чтобы посчитать доступную память;
   * 32-разрядный сегмент стека.
5. Охарактеризуйте дескриптор сегмента дополнительной памяти, который вы описали.  
   ?.
6. Почему ваша таблица дескрипторов прерываний имеет такую структуру?  
   ?.
7. Что вы написали для исключений?  
   IDT, обработки для исключений (заглушки) и обработчики для прерываний клавиатуры и системного таймера.
8. Как адресуются аппаратные прерывания в защищенном режиме? (самый "ударный" вопрос)  
   см. рисунок из семинара 2.
9. Когда вызывается ваш обработчик от клавиатуры / системного таймера?  
   Для клавиатуры: вызывается при нажатии или отжатии кнопки на клавиатуре.  
   Для таймера: 18.2 раз в секунду.
10. Какие действия необходимо выполнить для корректного возвращения компьютера в реальный режим?  
    По коду:
    * Заперт MI;
    * \*far jmp;
    * Обнулить флаг pe в CR0;
    * Обновить сегментные регистры;
    * Перенастроить базовый вектор контроллера прерываний;
    * Восстановить маски контроллеров прерываний;
    * Восстановить состояние IDTR реального режима;
    * Закрыть линию А20 (Не обязательно?);
    * Разрешить MI и NMI.
11. Что такое теневые регистры? Для чего они включены в процессор? Какую информацию содержат?  
    «Теневые регистры находятся в процессоре. Нужны чтоб исключить постоянные обращения к физической памяти. Тк мы обращаемся к физ. памяти на каждой команде, а то и несколько раз, особенно если адресация косвенная.» (отвечать, наверное, лучше по-другому)  
    Из семинара 3: Задача теневых регистров – хранить информацию о текущем сегменте в самом процессоре, чтобы не обращаться в памяти (затратное действие). (чтобы избежать обращение к таблице дескрипторов)
12. Линия А20. Что произойдет с памятью, если при переходе в защищенный режим забудем открыть линию А20?  
    В реальном режиме линия A20 закрыта. Когда компьютер переводится из реального режима в защищённый, она должна быть принудительно обнулена. Иначе адреса с 1 в 20ой линии будут недоступны.
13. Что произойдет, если при возвращении в реальный режим забудем закрыть линию А20?  
    Если мы в реальном режиме откроем линию A20, нам станет доступно еще 64 Кб памяти (HMA)

**ТРЕБУЕТСЯ ПОЛНОЕ ПОНИМАНИЕ**

*/\**

*Привилегии != приоритеты  
Привилегии - 4 кольца защиты  
Приоритеты - приоритет процессов, потоков, который назначается и может быть пересчитан.   
Процессы выстраиваются в очередь в соответствии с приоритетом.*

*\*/*

### Что нужно сделать для перевода из р-р в з-р (Проверено)

Описать GDT:

* + Нулевой (обязательный)
  + 16-разрядный сегмент кода для реального режима;
  + 16-разрядный сегмент данных, для хранения данных (строки для вывода, например);
  + 32-разрядный сегмент кода для защищенного режима;
  + 32-разрядный сегмент данных, чтобы посчитать доступную память;
  + 32-разрядный сегмент стека.

### Какие задачи выполняет GDT и IDT (Проверено)

GDT используется для адресации программы. (Проверено)Таблица глобальных дескрипторов содержит дескрипторы сегментов физической памяти.

IDT – через шлюзы осуществляется доступ к обработчикам прерываний и исключений. (шлюз – дескриптор обработчика прерывания) (НЕ ВЕРНО ЫЫЫ)

IDT – Чтобы адресовать обработчики прерываний в защищённом режиме (Проверено)

### Почему при переходе в з-р нужно перепрограммировать контроллер прерываний на новый базовый вектор? (Проверено)

Поскольку в з-р векторы 0…31 зарезервированы за исключениями, базовые **дескрипторы (не векторы!)** контроллеров прерываний должны быть расположены за исключениями в диапазоне 32…248.