**Какие бывают прерывания?**

1. системные вызовы (программные прерывания)
2. исключения (деление на ноль, и т.д.)
3. аппаратные прерывания

**Сколько раз возникает тик int 8h?**

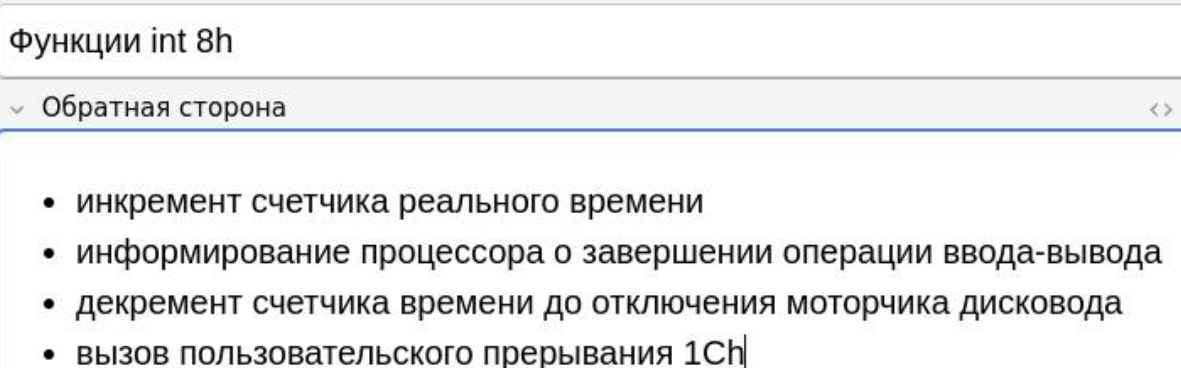
~18,2 раза в секунду

**Почему тик int 8h возникает именно такое число раз?**

Так как ~18,2 раз/сек - частота тактового генератора

**Как называются 4 push и 4 pop в начале и конце?**

Сохранение/восстаноление аппаратного контекста



**Почему перед косвенным вызовом обработчика 1Ch мы сохраняем флаги?**

Потому что при завершении выполнения обработчика 1Ch выполняется IRET, которая восстанавливает регистр флагов.  
При вызове обработчика с помощью INT эта команда сохраняет регистр флагов, а при косвенном вызове он не сохраняется, поэтому для корректной работы IRET необходимо отдельно сохранять регистр флагов.

**Что за прерывание 1Ch? Что внутри?**

Пользовательское прерывание. Внутри после инициализации системы только IRET (по сути - заглушка)

**Зачем после вызова 1Ch вызывать subroutine?**

Затем, чтобы снова сбросить IF, так как из-за IRET в 1Ch он мог испортиться

**Кто может заблокировать/разблокировать процесс?**

Только ядро (системные вызовы)

**Как называется действие по типу отключения моторчика?**

Отложенное действие

**На что можно повесить отложенные действия?**

Только на таймер. На независимое действие

**Для чего сохранять и восстанавливать аппаратный контекст?**

Для того, чтобы прерванная программа могла продолжить свое выполнение

**Что делает subroutine?**

Запрещает маскируемые прерывания

|  |
| --- |
| Прерывание - особая ситуация, когда выполнение текущей программы приостанавливается и управление передаётся программе-обработчику возникшего прерывания.  Прерывание можно рассматривать как некоторое особое событие в системе, требующее моментальной реакции. Практически все системы ввода/вывода в компьютере работают с использованием прерываний. Например: нажатие клавиши, щелчок мышкой - аппаратура вырабатывает прерывания. В ответ на них система, соответственно, считывает код нажатой клавиши или запоминает координаты курсора мыши.  Виды прерываний:  ● аппаратные (асинхронные) - события от внешних устройств;  ● внутренние (синхронные) - события в самом процессоре, например, деление на ноль;  ● программные - вызванные командой int.  Внешние прерывания, в зависимости от возможности запрета, делятся на:  ● маскируемые — прерывания, которые можно запрещать установкой соответствующего флага;  ● немаскируемые (англ. Non-maskable interrupt, NMI) — обрабатываются всегда, независимо от запретов на другие прерывания |

**Почему именно перед AND в subroutine стоит LOCK?**

AND – 2 обращения к памяти и потому что IF это крайне важный флаг (флаг прерываний 0-не обрабатывает прерывания)

Потому, что AND представляет собой два действия: чтение аргументов из памяти и запись результата в память, а LOCK блокирует локальную шину данных, по которой приходят маски прерываний, при этом единственное прерывание, которое может прервать -- int 8h, происходит борьба с этим

**Что такое маскируемые и немаскируемые прерывания?**

Маскируемые - прерывания, которые вызываются по маске. Их  можно запретить

Маскируемые прерывания — это прерывания, которые можно запрещать установкой соответствующих битов в регистре маскирования прерываний.

**По какой шине приходят маски прерываний?**

По шине данных

**Как получили адрес обработчика прерывания?**

Таблица векторов прерываний (IDT) в DOS расположена по нулевому адресу. Зная номер прерывания, можно получить смещение в этой таблице, по которому расположен дальний (far) адрес обработчика прерывания. Размер дальнего адреса - 4 байта (2 на сегмент и 2 на смещение). Например, для 8 прерывания смещение = 8 \* 4 = 32. Надо иметь в виду, что в x86 используется обратный порядок записи байтов (little endian), поэтому если таблица векторов прерываний получена в виде массива байтов, то для получения шестнадцатеричного адреса необходимо перевернуть расположенные по найденному смещению 4 байта.

**Что значит каждая из колонок листинга?**

(слева - направо) адреса (сегмент-смещение!), машинный код, команды ассемблера, комментарии.

**Зачем нужно прерывание 1Ch, если есть 8h?**

1Ch – программное прерывание, а 8h — аппаратное. Если изменять код аппаратного, оно будет выполняться долго, а так как в doc у int8h самый высокий приоритет, то все процессорное время будет занято выполнением кода int8h.

**Что это за адрес - 0040:0314h ?**

Область данных биос, там содержится копия флагов.

**Что делает префиксная команда lock?**

Указывает на то, что следующее за ней команда должна выполняться, как неделимая. Прервать выполнение след за lock команды нельзя. Технически это осуществляется блокировкой локальной шины памяти. На время выполнения команды блокируется локальная шина памяти.

**Почему lock перед and?**

Связано с особенностью работы and, операция объёмная и требует много времени.