Interrupts

Прерывания в операционных системах

Таир Сабыргалиев

Как работают операционные системы?

- Поддержка в железе
  - Управление (разграничение) памяти
  - Многозадачность, многопроцессорность
  - **Обработка исключений и прерываний**
  - Управление кэшами процессора
  - Работа с устройствами (диски, сети, …) и их питанием
  - Отладка и мониторинг производительности
- Набор алгоритмов для производительности и безопасности
  - Управление процессами (запуск, учет времени, памяти, приоретизация, …)
  - Файловые системы
  - Сети (все что выше L1/2?)
  - Учет пользователей
  - и тд.

Прерывания

- Механизм, обеспечивающий возможность обработки
  - событий от устройств:
    - таймер, клавиатура, мышь, сеть, диски(?)
  - программных исключений
    - деление на ноль, обращение к невыделенному участку памяти (pagefault) и другие
  - системных вызовов: все, что нельзя делать без разрешения операционной системы, чтение/запись файлов, общение по сети
    - сегодня системные вызовы уже используют более быстрый механизм: syscall/sysret или sysenter/sysexit

Прерывания от железа

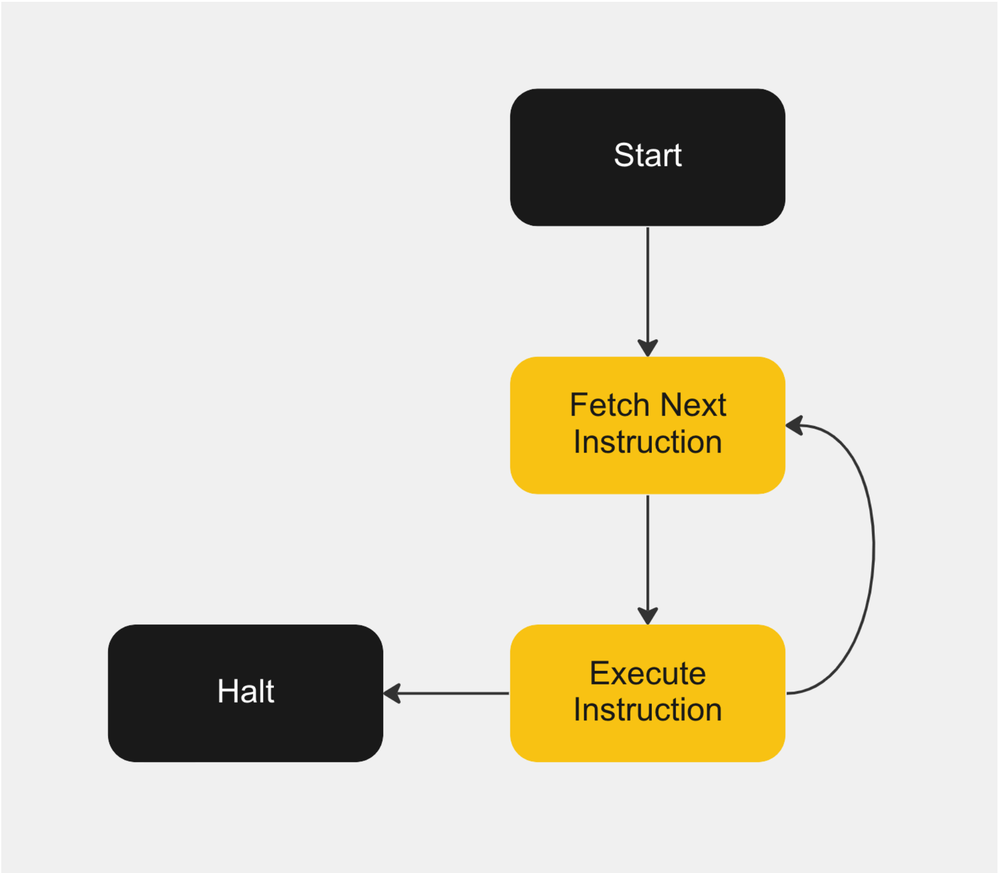
- Клавиатура/мышь: нажали на клавишу, кнопку мыши
- Сеть: поступил пакет и требует обработки
- Диски: поступил блок данных, которые у диска запросили ранее
- Таймер: нужно решить продолжать текущую задачу или переключиться на другую

Программные прерывания

- Системный вызов
  - Альтернатива: SYSENTER/SYSEXIT
- Деление на ноль
- Page fault
  - Minor: физический адрес еще недоступен данной задаче, нужно настроить
  - Major: также требуется часть физической памяти подгрузить с диска

Процессоры без прерываний

- Процессору дается задача
    1. Процессор загружает из памяти следующую инструкцию
    2. Исполняет инструкцию
    3. Возвращаемся к п. 1 или заканчиваем задачу

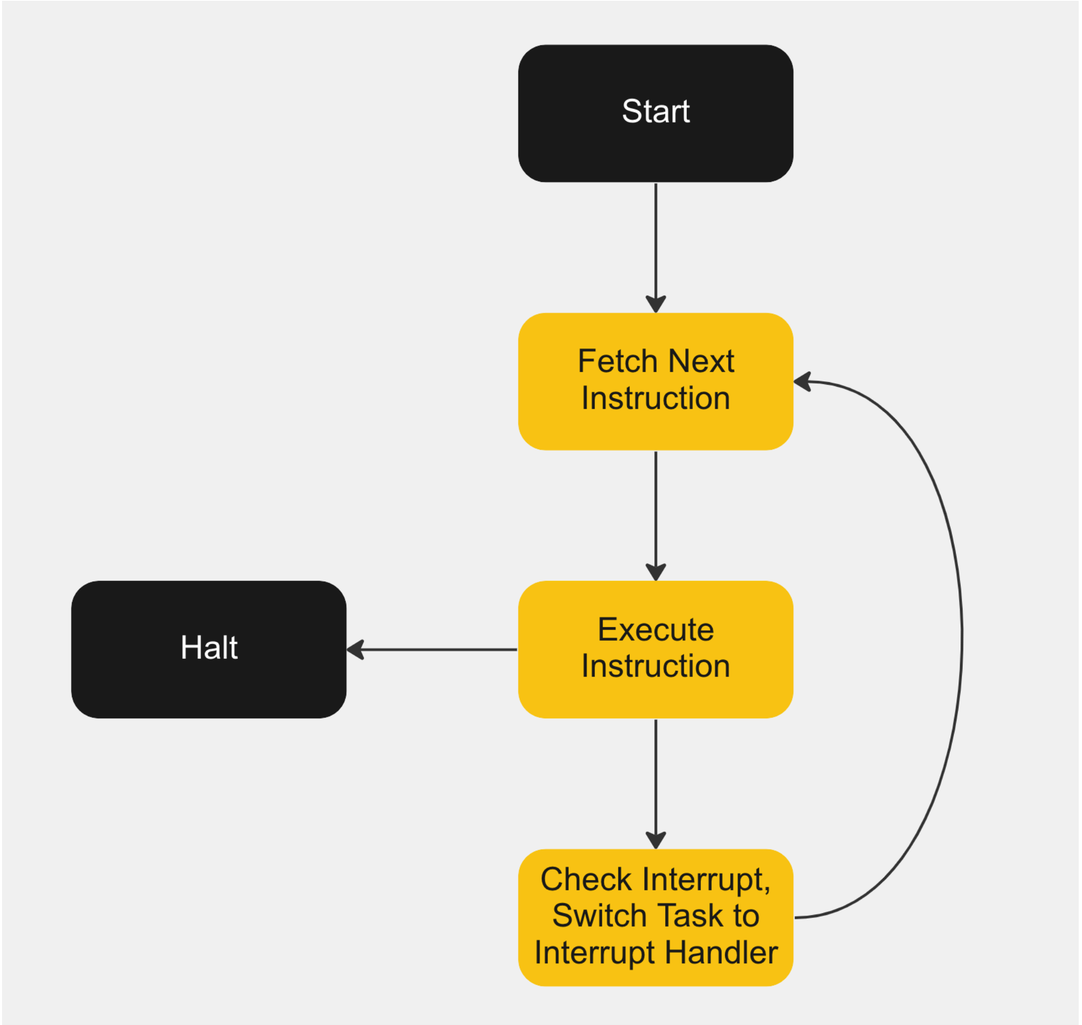


Процессоры без прерываний

- Процессору дается задача
    1. Процессор загружает из памяти следующую инструкцию
    2. Исполняет инструкцию
    3. Возвращаемся к п. 1 или заканчиваем задачу

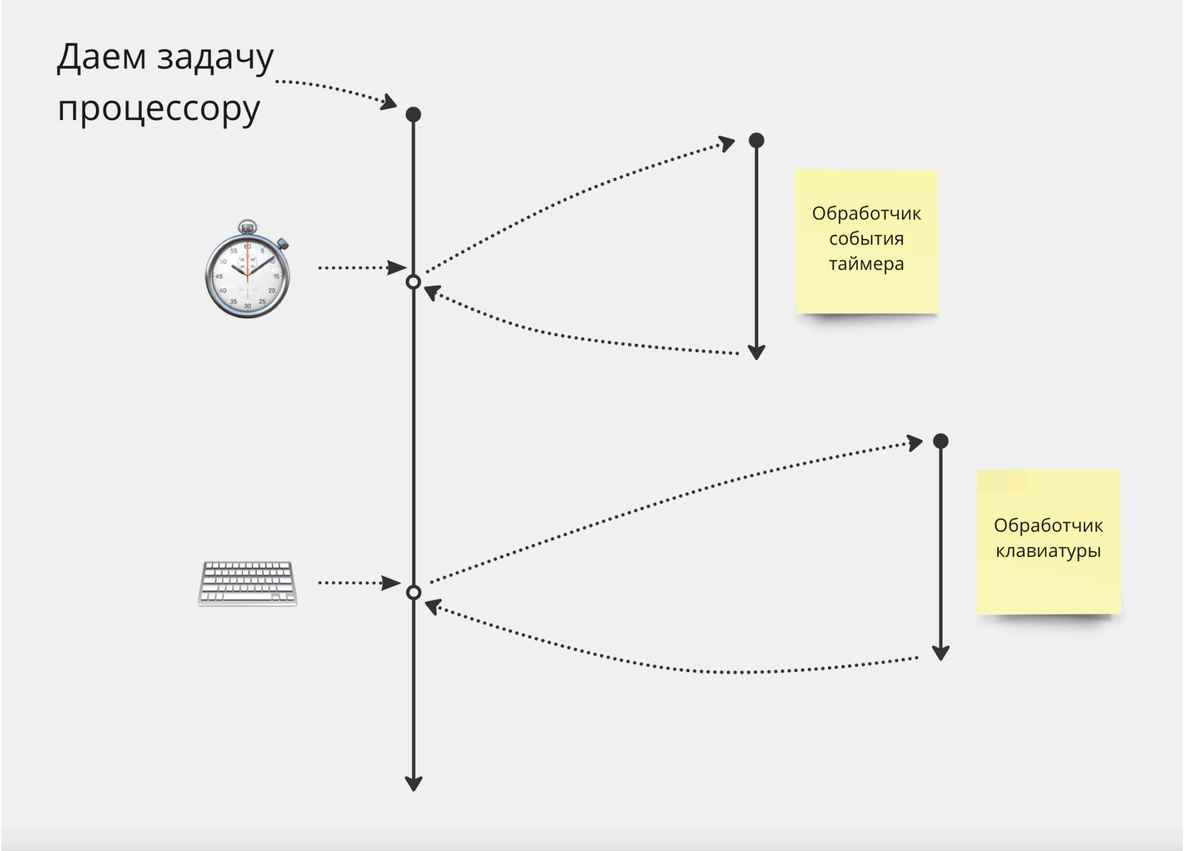
Процессоры с прерываниями

- Процессору дается задача
  1. Загружаем из памяти следующую инструкцию
  2. Исполняем ее
  3. Если появилось прерывание, переключаем задачу на обработчик прерывания и идем в п.1



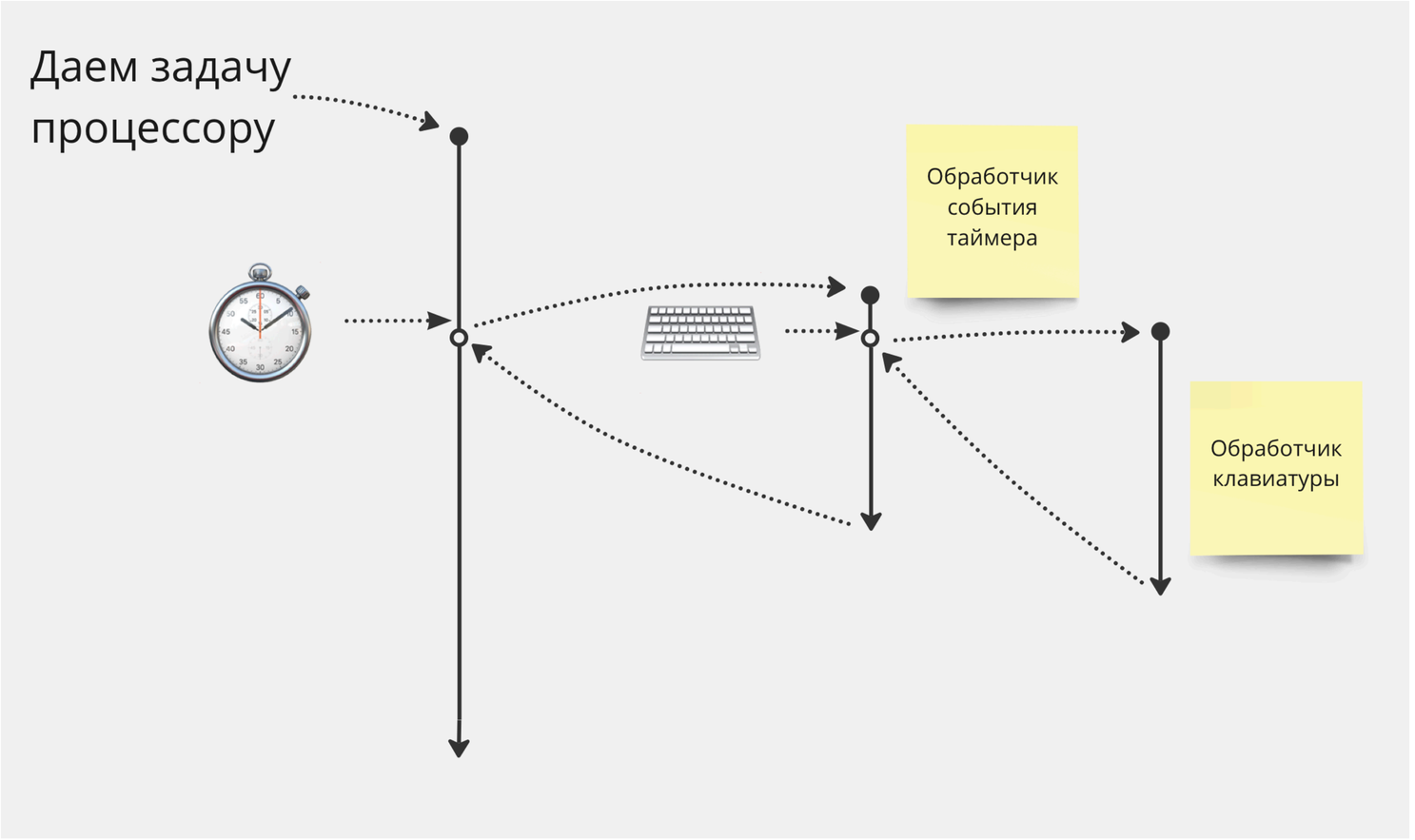
Процессоры с прерываниями

- Процессору дается задача
    1. Загружаем из памяти следующую инструкцию
    2. Исполняем ее
    3. Если появилось прерывание, переключаем задачу на обработчик прерывания и идем в п.1



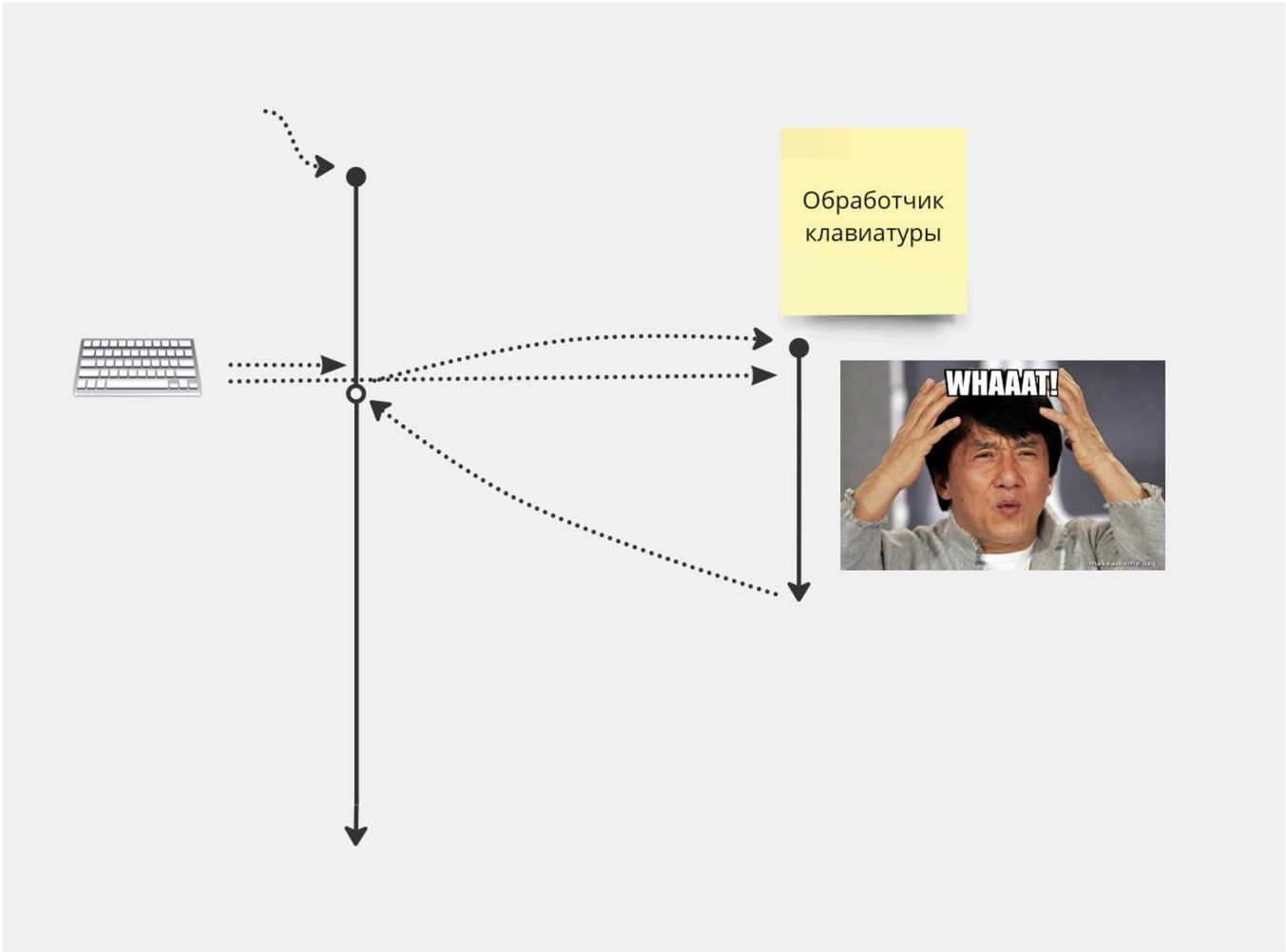
Процессоры с прерываниями

- А что делать с вложенными прерываниями?



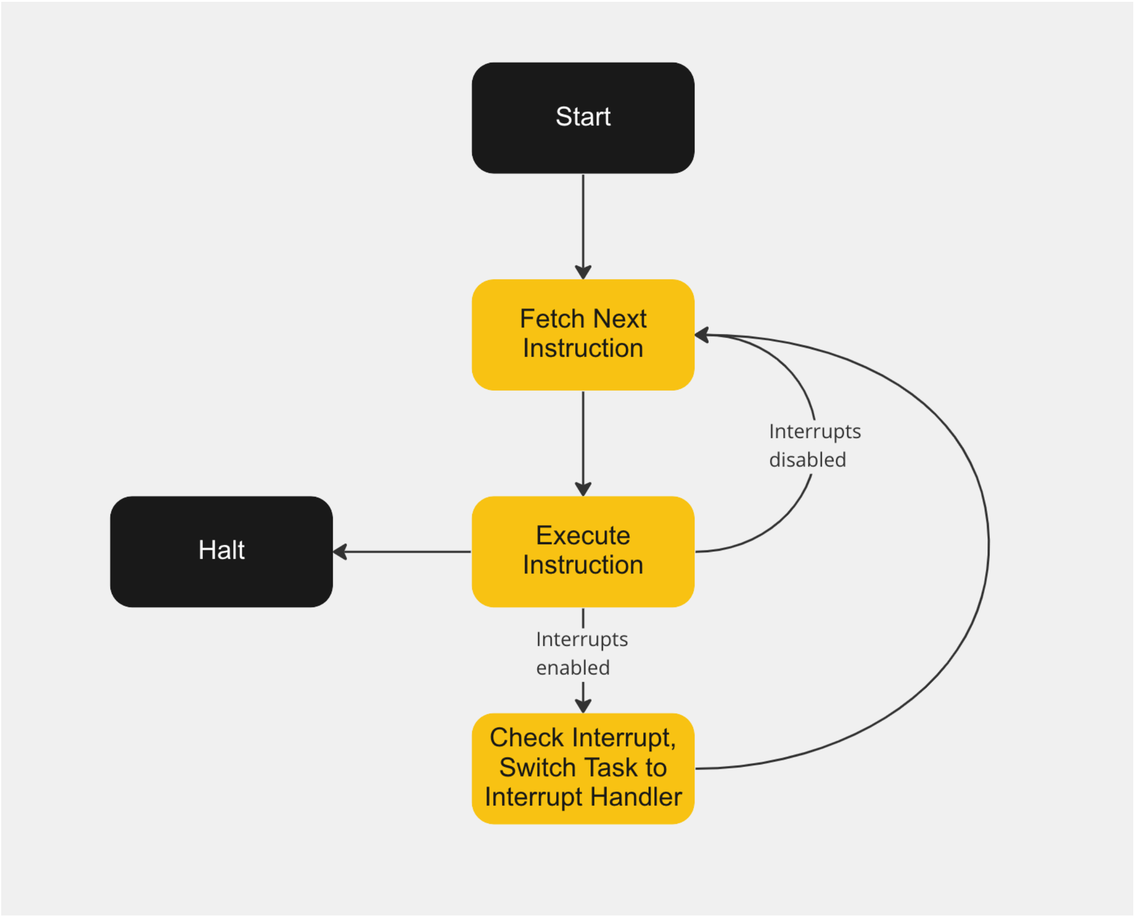
Процессоры с прерываниями

- А что делать с вложенными прерываниями?
  - Можно их выключить и отложить обработку
    - CLI — disable interrupts
    - STI — enable interrupts
  - Некоторые прерывания не выключаются, например сигнал об отказе оборудования, памяти и тп.



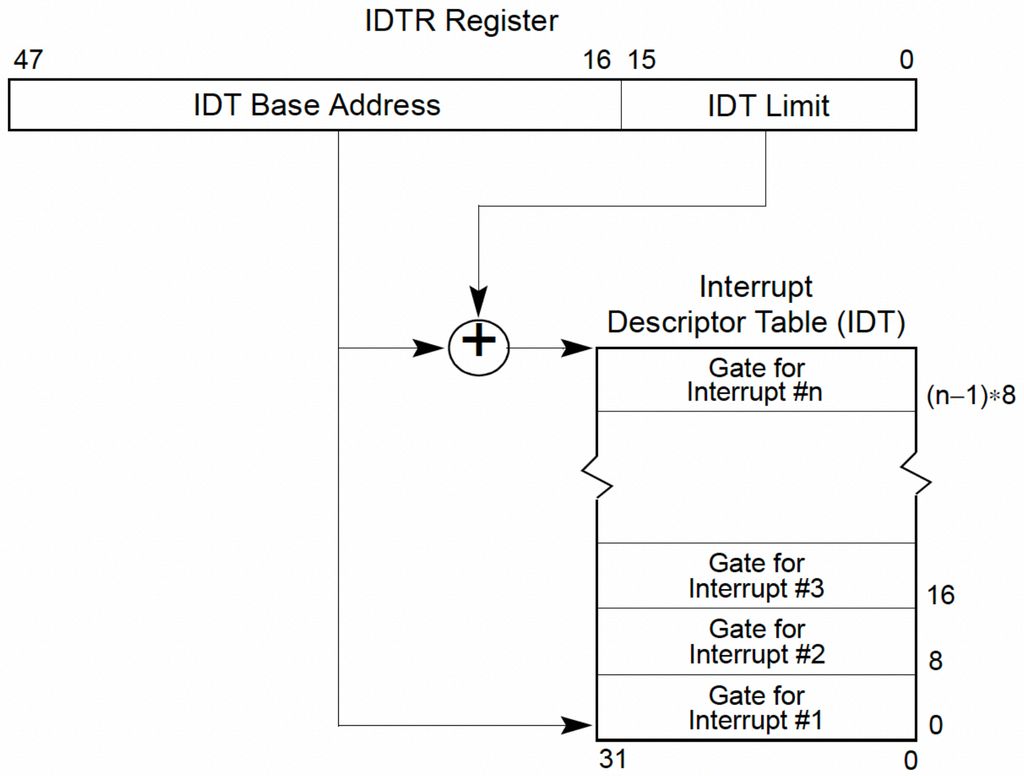
Процессоры с прерываниями

- Процессору дается задача
    1. Загружаем из памяти следующую инструкцию
    2. Исполняем ее
    3. Если прерывания ВЫКЛЮЧЕНЫ идем к п. 1
    4. Иначе, если ЕСТЬ прерывание, переключаем задачу на обработчик прерывания и идем к п. 1



Обработчики прерываний

- Включается компьютер, загружается базовая система ввода-вывода (BIOS), ее задача: загрузить полноценную ОС из диска или откуда-то еще, например по сети и передать ей управление.
- Полноценная ОС запускается в режиме супервизора (Ring 0), что позволяет ей настраивать некоторые части процессора.
  - Настраиваем частоту таймера
  - В памяти выделяем таблицу с адресами обработчиков прерываний и адрес этой таблицы кладем в специальные регистры процессора.
  - Включаем обработку прерываний
  - Настраиваем кучу других вещей в железе и софте
- Далее ОС передает процессор задаче в режиме пользователя (Ring 3), например процессу *init* в Линуксе
- При возникновении прерывания, процессор приостанавливает задачу, передает управление нужному обработчику
- При выходе из обработчика процессор возвращает задачу, которая была прервана ранее



Полезные ссылки

- wiki.osdev.org
- Intel® 64 and IA-32 Architectures Software Developer’s Manual