Диспетчер исключений.
Экспортируется как **KiUserExceptionDispatcher()**.
При возникновении исключения ядро формирует в стеке фрейм(**EXCEPTION_POINTERS**, **EXCEPTION_RECORD** и **CONTEXT)** и передаёт управление на эту точку.

- Сохраняется полный контекст потока,(**CONTEXT.ContextFlags = 0x1003F**).

- Исключение не разворачивается в юзермоде и процесс завершается в случае:
  1. Не валидный стек(не доступен для записи), адресуемый регистром **Esp**.
  2. Селектор стека(содержимое регистра **Ss**) отличен от **KGDT_R3_DATA or RPL_MASK = 0x23**.

- Стек выравнивается на границу 4-х байт.

- При входе в диспетчер исключений восстанавливаются в нормальные значения все сегментные регистры:
  Cs = KGDT_R3_CODE or RPL_MASK(0x1B)
  Ds = KGDT_R3_DATA or RPL_MASK(0x23)
  Es = KGDT_R3_DATA or RPL_MASK(0x23)
  Fs = KGDT_R3_TEB or RPL_MASK(0x3B)
  Gs = 0

- Сохраняются регистры общего назначения:
  Eax, Ecx, Edx, Ebx, Esi, Edi, Ebp

- Сохраняются флажки:
  ID A R N IOP O D I T S Z A P C
  0.. X 0 0 X 0 0 0 X X X X X 1 X X X 0 X 0 X 1 X
  Если возникает исключение **#DB(STATUS_SINGLE_STEP)**, то в контексте потока находящемся в стеке и текущем сбрасывается **TF(Trap Flag)**.
  Для иного исключения, отличного от **#DB(STATUS_SINGLE_STEP)** на момент возникновения которого был взведён **TF**, вход в диспетчер исключений выполняется с взведённым **TF**. После чего генерируется трассировочное исключение**(#DB)** и **TF** сбрасывается.
  Сохраняется поле **IOPL**.

- Сохраняются отладочные регистры **Dr0 % Dr3**.
  Сохраняется регистр управления **Dr7**, сбрасываются биты **11, 12, 14, 15**, взводится бит **10**.
  Сохраняется регистр состояния **Dr6**:
  T S D B3 ~~ B0
  1.. X X X 0 1 1 1 1 1 1 1 1 X X X X
  Так как регистр управления не сбрасывается, возможно зацикливание диспетчера исключений, а далее переполнение стека, изза чего ядро завершает процесс.

- В контексте потока регистр **Dr6** содержит информацию об трассировочном исключении. Исли одновременно происходит два останова, например аппаратный и пошаговый, будет взведено два флажка в **Dr6 - BS и D#**.

- В контексте потока находящимся в стеке и в текущем сохраняются значения отладочных регистров, если на момент возникновения исключения регистр **Dr7** отличен от нуля, иначе все они сбрасываются. Тоесть пока не были изменены отладочные регистры вручную, при пошаговом останове все регистры сброшены(в частности **Dr6**, в котором бит **BS** также сброшен).

- Если значение регистра **Dr7** отлично от нуля, хардварные регистры не сбрасываются и сохраняются. Так при пошаговом останове контекст(текущий и в стеке) будет содержать взведённый бит **BS** в регистре состояния **Dr6**.

- Возврат потока на диспетчер исключений из ядра выполняется посредством инструкции **IRetd**, изза чего пошаговый останов может быть сгенерирован только после исполнения первой инструкции диспетчера исключений. Изза этого аппаратный останов на первой инструкции диспетчера не сработает.

- При обработке **#BP(STATUS_BREAKPOINT)** выполняется декремент регистра **Eip** в контексте. Изза этого при генерации исключения двухбайтовой инструкцией **Int 3(0x03CD)** в контексте регистр **Eip** будет указывать на середину инструкции.

Апрель 2009, virustech.org