Как использовать User Accessible MMIO в KolibriOS

1. В файле kernel/bus/pci/pci32.asm раскомментировать такую строчку в начале:

**mmio\_pci\_addr = 0x400 ; set actual PCI address here to activate user-MMIO**

в данном случае 0x400 обозначает «шина 4 устройство 0 функция 0»

Вообще mmio\_pci\_addr двухбайтовый и имеет следующий формат:

**Назначение битов: bbbbbbbb dddddfff (bus-device-function). mmio\_pci\_addr - 16 bits or 4 bytes. 0x400 means bus4 device0 func0**

*Замечание:* Если fasm ругается на использование константы PG\_UW т.е говорит что ее не нашел, но напишите вместо нее 0x007 .

1. После этого программа PCIDEV также будет показывать информацию об mmio\_pci\_addr красным шрифтом.
2. Становятся доступны подфункции 11, 12, 13 системной функции 62.

Они не документированы в вики. Поэтому я, почитав исходники pci32.asm понял как их использовать т.е в какие регистры что ложить, и потом составил такие обертки для их вызова из Си:

static inline dword pci\_mmio\_init (int bus, int dev, int func)

{

 dword val;

 word dev\_addr = (bus << 8) | (dev << 3) | func; // is it correct? Seems to be so

 \_\_asm\_\_ \_\_volatile\_\_(

 "int $0x40"

 :"=a"(val)

 :"a"(62),"b"(11),"c"(dev\_addr));

 return val;

}

static inline dword pci\_mmio\_map (int bar\_index, int block\_size, int offset\_in\_mmio\_block)

{

 dword val;

 word bx = (bar\_index << 8) | 12;

 \_\_asm\_\_ \_\_volatile\_\_(

 "int $0x40"

 :"=a"(val)

 :"a"(62),"b"(bx),"c"(block\_size),"d"(offset\_in\_mmio\_block));

 return val;

}

static inline dword pci\_mmio\_unmap (int lin\_addr) // returns 1 if success, 0 if fail

{

 dword val;

 \_\_asm\_\_ \_\_volatile\_\_(

 "int $0x40"

 :"=a"(val)

 :"a"(62),"b"(13),"c"(lin\_addr));

 return val;

}