ATA read/write sectors

From OSDev Wiki

Since interrupts (like INT 13h) can't be called easily in protected mode or long mode, direct disk access through ports might be the only solution. The below code is example of subroutines to read and write disk sectors directly from the first hard disk (80h) in long mode using CHS and LBA.

Contents

ATA read sectors

Read in CHS mode

Accessing disk using CHS (cylinder,head,sector) indexes seem to be kinda old type but is the base for LBA access. The following NASM long-mode subroutine reads CH sectors to memory address indicated by RDI register.

```
;=============================================================================
; ATA read sectors (CHS mode)
; Max head index is 15, giving 16 possible heads
; Max cylinder index can be a very large number (up to 65535)
; Sector is usually always 1-63, sector 0 reserved, max 255 sectors/track
; If using 63 sectors/track, max disk size = 31.5GB
; If using 255 sectors/track, max disk size = 127.5GB
; See OSDev forum links in bottom of [http://wiki.osdev.org/ATA]
;
; @param EBX The CHS values; 2 bytes, 1 byte (BH), 1 byte (BL) accordingl
; @param CH The number of sectors to read
; @param RDI The address of buffer to put data obtained from disk
;
; @return None
;=============================================================================
ata_chs_read:   pushfq
                push rax
                push rbx
                push rcx
                push rdx
```

```
                push rdi

                mov rdx,1f6h            ;port to send drive & head number
                mov al,bh               ;head index in BH
                and al,00001111b        ;head is only 4 bits long
                or  al,10100000b        ;default 1010b in high nibble
                out dx,al

                mov rdx,1f2h            ;Sector count port
                mov al,ch               ;Read CH sectors
                out dx,al

                mov rdx,1f3h            ;Sector number port
                mov al,bl               ;BL is sector index
                out dx,al

                mov rdx,1f4h            ;Cylinder low port
                mov eax,ebx             ;byte 2 in ebx, just above BH
                mov cl,16
                shr eax,cl              ;shift down to AL
                out dx,al

                mov rdx,1f5h            ;Cylinder high port
                mov eax,ebx             ;byte 3 in ebx, just above byte 2
                mov cl,24
                shr eax,cl              ;shift down to AL
                out dx,al

                mov rdx,1f7h            ;Command port
                mov al,20h              ;Read with retry.
                out dx,al

.still_going:   in al,dx
                test al,8               ;the sector buffer requires servi
                jz .still_going         ;until the sector buffer is ready

                mov rax,512/2           ;to read 256 words = 1 sector
                xor bx,bx
                mov bl,ch               ;read CH sectors
                mul bx
                mov rcx,rax             ;RCX is counter for INSW
                mov rdx,1f0h            ;Data port, in and out
                rep insw                ;in to [RDI]

                pop rdi
                pop rdx
                pop rcx
                pop rbx
                pop rax
                popfq
                ret
```

Read in LBA mode

This page or section is a stub. You can help the wiki by *accurately* contributing (http://wiki.osdev.org/index.php?title=ATA_read/write_sectors&action=edit) to it.

```
;=============================================================================
; ATA read sectors (LBA mode)
;
; @param EAX Logical Block Address of sector
; @param CL  Number of sectors to read
; @param RDI The address of buffer to put data obtained from disk
;
; @return None
;=============================================================================
ata_lba_read:
               pushfq
               and rax, 0x0FFFFFFF
               push rax
               push rbx
               push rcx
               push rdx
               push rdi

               mov rbx, rax         ; Save LBA in RBX

               mov edx, 0x01F6      ; Port to send drive and bit 24 - 27
               shr eax, 24          ; Get bit 24 - 27 in al
               or al, 11100000b     ; Set bit 6 in al for LBA mode
               out dx, al

               mov edx, 0x01F2      ; Port to send number of sectors
               mov al, cl           ; Get number of sectors from CL
               out dx, al

               mov edx, 0x1F3       ; Port to send bit 0 - 7 of LBA
               mov eax, ebx         ; Get LBA from EBX
               out dx, al

               mov edx, 0x1F4       ; Port to send bit 8 - 15 of LBA
               mov eax, ebx         ; Get LBA from EBX
               shr eax, 8           ; Get bit 8 - 15 in AL
               out dx, al
```

```
                mov edx, 0x1F5        ; Port to send bit 16 - 23 of LBA
                mov eax, ebx          ; Get LBA from EBX
                shr eax, 16           ; Get bit 16 - 23 in AL
                out dx, al

                mov edx, 0x1F7        ; Command port
                mov al, 0x20          ; Read with retry.
                out dx, al

.still_going:   in al, dx
                test al, 8            ; the sector buffer requires servicir
                jz .still_going       ; until the sector buffer is ready.

                mov rax, 256          ; to read 256 words = 1 sector
                xor bx, bx
                mov bl, cl            ; read CL sectors
                mul bx
                mov rcx, rax          ; RCX is counter for INSW
                mov rdx, 0x1F0        ; Data port, in and out
                rep insw              ; in to [RDI]

                pop rdi
                pop rdx
                pop rcx
                pop rbx
                pop rax
                popfq
                ret
```

ATA write sectors

A write is mostly equivalent to performing a read operation. The only changes needed are a change in command (0x30 for chs write), and the direction of the data, which is written (rep outsw from *SI) to the data port rather than read (rep insw to *DI).

See also

- Read/write disk sectors by Dex (http://forum.osdev.org/viewtopic.php?t=12268)
- ATA PIO mode
- ATA

Retrieved from "http://wiki.osdev.org/index.php?title=ATA_read/write_sectors&oldid=13525"
Categories: Stubs | ATA

- This page was last modified on 10 June 2012, at 23:29.
- This page has been accessed 11,092 times.