**Tutorial 19**

El tutorial 19 hace referencia a como poder controlar el teclado y poder leer la información del controlador de teclado. Cada vez que una tecla es presionada en el teclado, el controlador de teclado dispara el IRQ1, como se definió en una tabla pasada en otro tutorial. El sistema operativo se encarga de leer una vez que se dispara el IRQ1, si el teclado devolvió un scan code.

|  |  |  |
| --- | --- | --- |
| **Keyboard Controller Ports** | | |
| Port | Read/Write | Descripton |
| Keyboard Encoder | | |
| 0x60 | Read | Read Input Buffer |
| 0x60 | Write | Send Command |
| Onboard Keyboard Controller | | |
| 0x64 | Read | Status Register |
| 0x64 | Write | Send Command |

El teclado se compone exactamente de dos microcontroladores. Uno esta en el teclado y el el Keyboard encoder, y se encarga de enviar al controlador de teclado en la motherboard cada uno de los tecleos realizados con su scan code. El controlador de teclado en el motherboard se encarga de leer cada uno de los tecelos del keyboard encoder y dispara un IRQ1 cada vez que esto pase. El keyboard encoder también se encarga de controlar los LEDs del teclado y reportar la presencia del teclado en un sistema. Los comandos aceptados por el microcontrolador de teclado son los siguientes:

|  |  |  |
| --- | --- | --- |
| **Command Listing** | | |
| Command | Descripton |  |
| Common Commands | | |
| 0x20 | Read command byte |  |
| 0x60 | Write command byte |  |
| 0xAA | Self Test |  |
| 0xAB | Interface Test |  |
| 0xAD | Disable Keyboard |  |
| 0xAE | Enable Keyboard |  |
| 0xC0 | Read Input Port |  |
| 0xD0 | Read Output Port |  |
| 0xD1 | Write Output Port |  |
| 0xE0 | Read Test Inputs |  |
| 0xFE | System Reset |  |
| 0xA7 | Disable Mousr Port |  |
| 0xA8 | Enable Mouse Port |  |
| 0xA9 | Test Mouse Port |  |
| 0xD4 | Write To Mouse |  |
| Non Standard Commands | | |
| 0x00-0x1F | Read Controller RAM |  |
| 0x20-0x3F | Read Controller RAM |  |
| 0x40-0x5F | Write Controller RAM |  |
| 0x60-0x7F | Write Controller RAM |  |
| 0x90-0x93 | Synaptics Multiplexer Prefix |  |
| 0x90-0x9F | Write port 13-Port 10 |  |
| 0xA0 | Read Copyright |  |
| 0xA1 | Read Firmware Version |  |
| 0xA2 | Change Speed |  |
| 0xA3 | Change Speed |  |
| 0xA4 | Check if password is installed |  |
| 0xA5 | Load Password |  |
| 0xA6 | Check Password |  |
| 0xAC | Disagnostic Dump |  |
| 0xAF | Read Keyboard Version |  |
| 0xB0-0xB5 | Reset Controller Line |  |
| 0xB8-0xBD | Set Controller Line |  |
| 0xC1 | Continuous input port poll, low |  |
| 0xC2 | Continuous input port poll, high |  |
| 0xC8 | Unblock Controller lines P22 and P23 |  |
| 0xC9 | Block Controller lines P22 and P23 |  |
| 0xCA | Read Controller Mode |  |
| 0xCB | Write Controller Mode |  |
| 0xD2 | Write Output Buffer |  |
| 0xD3 | Write Mouse Output Buffer |  |
| 0xDD | Disable A20 address line |  |
| 0xDF | Enable A20 address line |  |
| 0xF0-0xFF | Pulse output bit |  |

Básicamente, lo que se tiene que programar es un IRQ Handler que cada vez que un IRQ1 es disparado, este lea de los puertos del controlador de teclado el ultimo scan code, interpretar el scan code, y luego desplegar el carácter en pantalla o pasarlo a nuestro programa siendo ejecutado en el sistema.