K-OS I/O

Kai Chen

CS134c Spring 2003

Overview

- **Low Level I/O (Text Mode)**
  - **Writing to video memory 0x6B000**
  - **Manipulating Cursor**
- **High Level I/O**
  - **printk(*fmt, …)**

Text Mode Basics

- **Video Memory**
  - **Video screen is memory mapped to 0x6B000**
  - **Screen divided into 25 rows and 80 columns**
  - **Each char is represented by 2 bytes:**
    - **ASCII value**
    - **Attributes (fg/bg color)**
- **Cursor Manipulation**
  - **Send command to CRT_CTRL register (0x03D4)**
  - **Read/write high/low address of cursor from/to CRT_DATA register (0x03D5)**
  - **Translating address to position on screen:**
    - **addr = row*80 + col**

High Level I/O

- **printk(char *fmt, …)**
  - **figure out the pointer to the first argument after *fmt. (*args)**
  - **Call vprintk(fmt, args)**
- **vprintk(char *fmt, void *args)**
  - **%d : signed decimal integer**
  - **%x : unsigned hexadecimal integer**
  - **%s : string**
  - **%c : char**

Printk(char *fmt, …)

- **Linux version:**
  - **va_start, va_end, va_list macros**
- **K-OS version:**

printk:

```
PUSH    EBP
MOV     EBP, ESP
MOV     EAX, EBP            ; EBP points to EBP
ADD     EAX, 12             ; EBP+4 points to return addr
PUSH    EAX                 ; EBP+8 points to the first arg *fmt
PUSH    dword [EBP+8]       ; EBP+12 points to the second arg
CALL    vprintk
MOV     ESP, EBP
POP     EBP
RET
```

Programming the Keyboard

- **Communicating with the KBD Controller**
  - **KBD_CTRL_REG: 0x64**
  - **KBD_STAT_REG: 0x64**
  - **KBD_DATA_REG: 0x60**
  - **A LOT of commands, boring details omitted**
- **Keyboard ISR**
  - **Read in scancode from keyboard controller**
  - **Translate scancode to ASCII (or extended ASCII)**
    - **2 keymap tables, one normal, one with SHIFT pressed**
  - **Write to keyboard buffer (not yet implemented)**
  - **Wake up a process awaiting keystroke (no process yet)**
  - **Other actions for special keys (e.g. Ctrl+Alt+Del)**