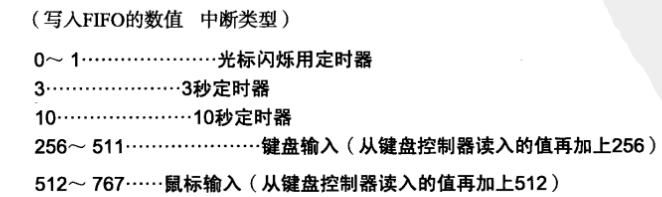
把画背景框，显示字符，刷新整合到一个函数中

void putfonts8\_asc\_sht(struct SHEET \*sht, int x, int y, int c, int b, char \*s, int l); // 画背景框，显示字符，刷新

把定时器，鼠标，键盘归纳到一个FIFO缓冲区管理



将结构体中的中断类型判断改为int，修改四个处理函数

struct FIFO32 // 缓冲

{

    int \*buf;

    int p, q, size, free, flags;

};

void fifo32\_init(struct FIFO32 \*fifo, int size, int \*buf); // 初始化FIFO缓冲区

int fifo32\_put(struct FIFO32 \*fifo, int data);             // 向FIFO传送数据并保存

int fifo32\_get(struct FIFO32 \*fifo);                       // 从FIFO取得一个数据

int fifo32\_status(struct FIFO32 \*fifo);                    // 报告一下积攒是数据量,查看缓冲区状态

键盘修改

extern struct FIFO32 \*keyfifo;

void inthandler21(int \*esp);                        // 来自PS/2键盘的中断

void init\_keyboard(struct FIFO32 \*fifo, int data0); // 初始化键盘控制电路

鼠标修改

extern struct FIFO32 \*mousefifo;                       // 鼠标输入缓冲

void inthandler2c(int \*esp);                // 来自PS/2鼠标的中断

void enable\_mouse(struct FIFO32 \*fifo, int data0, struct MOUSE\_DEC \*mdec); // 激活鼠标

定时器及加快中断处理修改

struct TIMER // 超时信息

{

    struct TIMER \*next;

    unsigned int timeout, flags;

    struct FIFO32 \*fifo;

    int data;

};

struct TIMERCTL

{

    unsigned int count, next;

    struct TIMER \*t0;

    struct TIMER timers0[MAX\_TIMER];

};

void init\_pit(void);                                                 // 初始化PIT

void timer\_init(struct TIMER \*timer, struct FIFO32 \*fifo, int data); // 定时器初始化

void timer\_settime(struct TIMER \*timer, unsigned int timeout);       // 设定定时器

void inthandler20(int \*esp);

测试，我的快太多了吧

