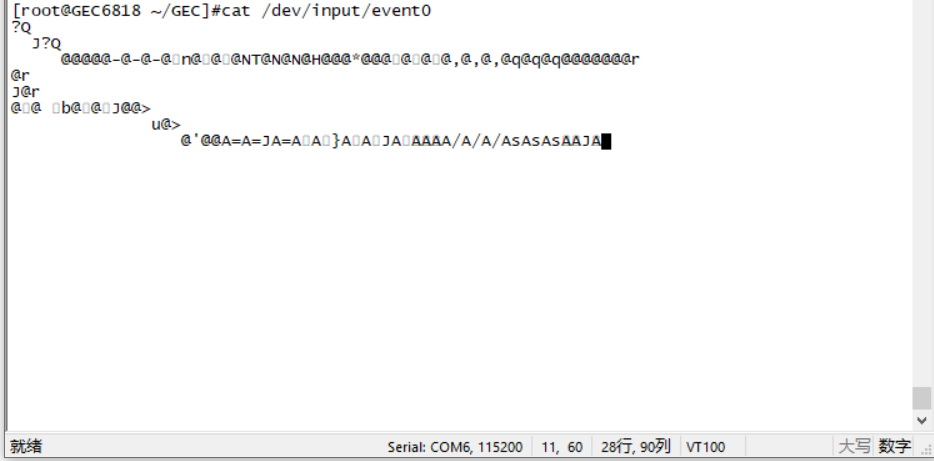
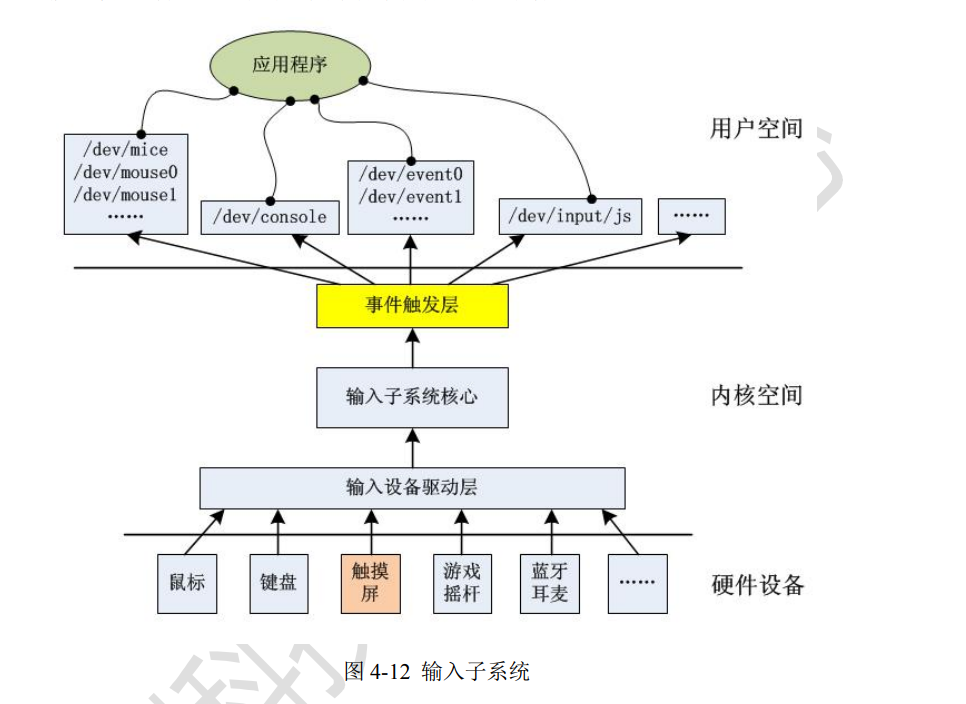
测试触摸屏的好坏：cat /dev/input/event0



如上图所示，去触摸屏幕显示乱码即代表触摸屏是好的。

**一、输入子系统**



输入设备驱动层：用于其特定硬件的驱动程序，向上提供接口、

输入子系统核心：将设备驱动层发来的数据进行整合，触发某一事件

事件触发层：通过用户空间读取相应的设备的节点来获取某一设备的动作

关注两个头文件：

gedit /usr/include/linux/input.h

struct input\_event {

struct timeval time; // 时间戳，输入设备此次操作发生的时间，精确到us级别

\_\_u16 type; // 输入事件的类型

\_\_u16 code;

// 输入事件类型的编码

\_\_s32 value;

// 输入事件类型码编码的值

};

gedit /usr/include/linux/input-event-codes.h

type:

EV\_KEY(0x01):按压事件(键盘，鼠标，触摸屏)

EV\_REL(0x02):相对事件（鼠标的滚轮）

EV\_ABS(0x03):绝对事件（触摸屏的坐标）

code：

ABS\_X：触摸屏的x轴坐标

ABS\_Y：触摸屏y轴坐标

BTN\_TOUCH：屏幕是松开还是按下，松开0，按下1

value：

你先去判断它是什么类型，再去判断它的编码，最后获取value值

#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

#include <sys/mman.h>

#include <linux/input.h>

int main(int argc, char const \*argv[])

{

    // 打开触摸屏

    int touch\_fd = open("/dev/input/event0", O\_RDWR);

    if (touch\_fd < 0)

    {

        printf("触摸屏打开失败\n");

        return -1;

    }

    // 读取触摸屏信息

    struct input\_event info;

    int x ,y;

    while (1)

    {

        read(touch\_fd, &info, sizeof(info)); // 阻塞，等待输入设备发生变化

        if (info.type == EV\_ABS)    // 判断是不是绝对事件

        {

            if (info.code == ABS\_X) // 是不是x轴

            {

                x = info.value;

                // printf("x = %d\n", info.value); // 蓝色屏幕

                //printf("x = %d\n", info.value\*800/1024);    // 黑色屏幕

            }

            if (info.code == ABS\_Y) // 是不是x轴

            {

                y = info.value;

                // printf("y = %d\n", info.value); // 蓝色屏幕

                //printf("y = %d\n", info.value\*480/600); //黑色屏幕

            }

        }

        if(info.type == EV\_KEY && info.code == BTN\_TOUCH && info.value == 1)    // 按下退出

            break;

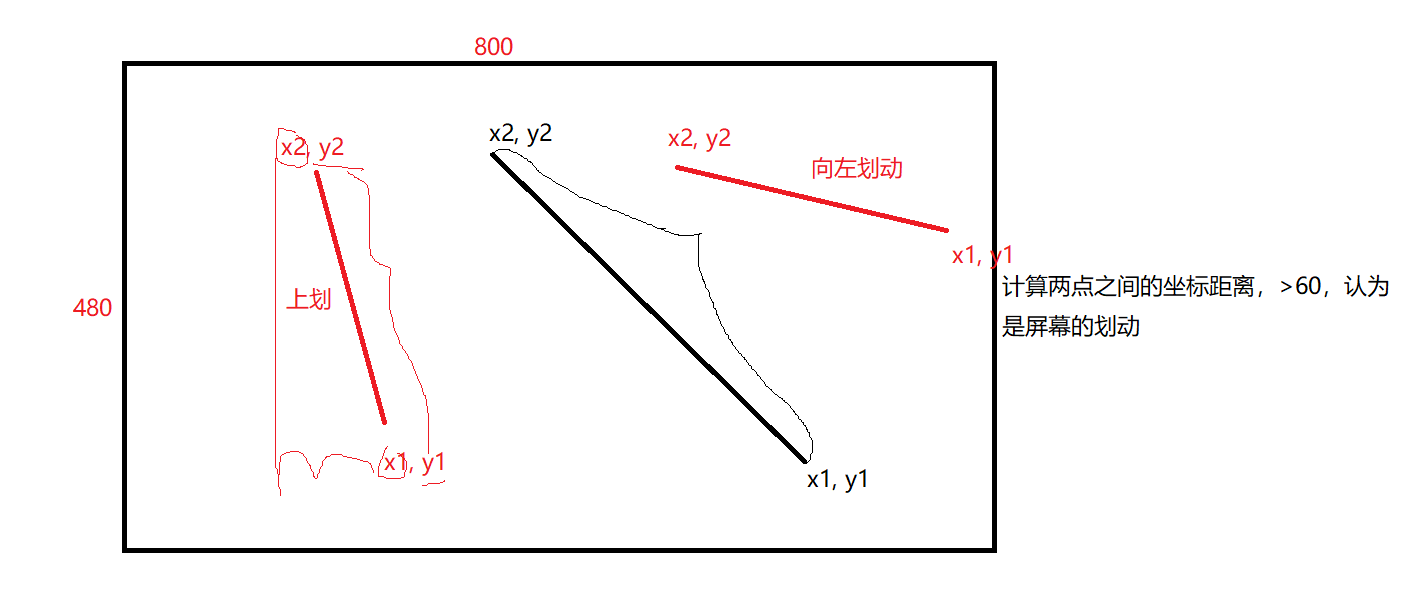
    }

    printf("x = %d, y = %d\n", x, y);

    return 0;

}

**二、屏幕的划动**



#include <stdio.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

#include <sys/mman.h>

#include <linux/input.h>

int get\_xy(int touch\_fd, int \*x, int \*y)

{

    // 读取触摸屏信息

    struct input\_event info;

    int temp\_x, temp\_y, x1, x2, y1, y2, var\_x, var\_y;

    while (1)

    {

        read(touch\_fd, &info, sizeof(info)); // 阻塞，等待输入设备发生变化

        if (info.type == EV\_ABS)    // 判断是不是绝对事件

        {

            if (info.code == ABS\_X) // 是不是x轴

            {

               temp\_x = info.value;

                // printf("x = %d\n", info.value); // 蓝色屏幕

                //printf("x = %d\n", info.value\*800/1024);    // 黑色屏幕

            }

            if (info.code == ABS\_Y) // 是不是x轴

            {

                temp\_y = info.value;

                // printf("y = %d\n", info.value); // 蓝色屏幕

                //printf("y = %d\n", info.value\*480/600); //黑色屏幕

            }

        }

        // 判断手指是按下还是松开，按下时获取第一组坐标，松开的时获取第二组坐标

        if(info.type == EV\_KEY && info.code == BTN\_TOUCH)

        {

            if (info.value == 1)  // 按下，获取第一组坐标

            {

                x1 = temp\_x;

                y1 = temp\_y;

            }

            else if(info.value == 0)    // 松开，获取第二组坐标

            {

                x2 = temp\_x;

                y2 = temp\_y;

                break;

            }

        }

    }

    // 计算第一组坐标与最后一组坐标的差值

    var\_x = x2 - x1;

    var\_y = y2 - y1;

    // 判断划动的距离，如果距离大于60就设置为划动，否则就视为点击

    if ((x2 - x1)\*(x2 - x1) + (y2 - y1)\*(y2 - y1) >= 60\*60) // 划动

    {

        if (var\_y < 0 && abs(var\_y) > abs(var\_x)) //abs取绝对值 , 上划

            return 1;

        else if(var\_y > 0 && abs(var\_y) > abs(var\_x)) // 向下划动

            return 2;

        else if(var\_x < 0 && abs(var\_y) < abs(var\_x)) // 左划

            return 3;

        else if(var\_x > 0 && abs(var\_y) < abs(var\_x)) // 右划

            return 4;

    }

    else // 点击

    {

        \*x = x2;

        \*y = y2;

        return 0;

    }

}

int main(int argc, char const \*argv[])

{

    // 打开触摸屏

    int touch\_fd = open("/dev/input/event0", O\_RDWR);

    if (touch\_fd < 0)

    {

        printf("触摸屏打开失败\n");

        return -1;

    }

    int x, y;

    while (1)

    {

        int p = get\_xy(touch\_fd, &x, &y);

        if (p == 1)

            printf("向上划动\n");

        else if (p == 2)

            printf("向下划动\n");

        else if (p == 3)

            printf("向左划动\n");

        else if (p == 4)

            printf("向右划动\n");

        else

            printf("x = %d, y = %d\n", x, y);

    }

    return 0;

}

作业：结合触摸屏与显示屏实现一个电子相册(百叶窗)