2014.11.02:

1.执行make menuconfig对配置进行修改时出现错误：

错误提示： HOSTCC scripts/basic/fixdep

HOSTCC scripts/kconfig/conf.o

\*\*\* Unable to find the ncurses libraries or the

\*\*\* required header files.

\*\*\* 'make menuconfig' requires the ncurses libraries.

\*\*\*

\*\*\* Install ncurses (ncurses-devel) and try again.

\*\*\*

make[1]: \*\*\* [scripts/kconfig/dochecklxdialog] 错误 1

make: \*\*\* [menuconfig] 错误 2

解决办法：yum install ncurses-devel

2014.11.04:

1.使用open函数时提示找不到fcnt1.h

解决办法：印刷错误，应该是fcntl.h，是英文字母l

2014.11.07:

1.已经搭建好win7以及和台式机器fedora之间以及和开发板的ftp,telnet登陆，但是未尝试Fedora和开发板之间的nfs挂载；乱码问题没解决；

2014.11.11:

1.常用：linux常用编码为UTF-8，而往往windows下常用ANSI格式，这里需要转换，请注意；

2014.11.13:

1.Fedora关闭防火墙：

重启后生效   
开启： chkconfig iptables on   
关闭： chkconfig iptables off   或者 /sbin/chkconfig --level 2345 iptables off  
  
 即时生效，重启后失效

**service 方式**开启： service iptables start   
关闭： service iptables stop

**iptables方式**

查看防火墙状态：  
/etc/init.d/iptables status

暂时关闭防火墙：  
/etc/init.d/iptables stop

重启iptables:  
/etc/init.d/iptables restart

2.ubuntu中如何关闭防火墙

sudo ufw disable 关闭防火墙

sudo ufw enable 开启防火墙

sudo ufw status 检查防火墙的状态

3.利用FlashTxp时发现登陆不上ubuntu，查看是否是选择“主动模式”、“被动模式”错误；

2014.11.15

1.两台linux机器之间利用ftp传输文件时需要若提示：

425 Can't open passive connection: Permission denied.

Passive mode refused.

解决方法：

passive mode off；即可；原因？

2.当在linux下编译好arm的可执行程序后，下载到arm板发现提示：

-sh: ./button: Permission denied

解决办法：传文件过程中文件属性发生修改，要把文件修改成可执行文件才可以被执行；

2014.11.18;

1.union数据结构;

公用内存，但只能取到一个变量；

**2. 如何在sourceinsight里面显示行号**

option->document option->editing options中，show line numbers

2014.11.20:

1. printk和printf的区别：

printk是在内核中运行的向控制台输出显示的函数，Linux内核首先在内核空间分配一个静态缓冲区，作为显示用的空间，然后调用sprintf，格式化显示字符串，最后调用tty\_write向终端进行信息的显示。

printk与printf的差异，是什么导致一个运行在内核态而另一个运行用户态？其实这两个函数的几乎是相同的，出现这种差异是因为tty\_write函数需要使用fs指向的被显示的字符串，而fs是专门用于存放用户态段选择符的，因此，在内核态时，为了配合tty\_write函数，printk会把fs修改为内核态数据段选择符ds中的值，这样才能正确指向内核的数据缓冲区，当然这个操作会先对fs进行压栈保存，调用tty\_write完毕后再出栈恢复。总结说来，printk与printf的差异是由fs造成的，所以差异也是围绕对fs的处理。

# 2. unix 中的EBUSY什么意思：

这是一种标准的错误值，表示 “设备或资源忙”错误。

2014.11.22:

1.编译驱动模块时可以只编译单个文件，如：

比如编译自带的驱动演示，qq2440\_hello.c，可以在配置了Makefile和Kconfig之后执行

make drivers/char/qq2440\_hello.o (注意这里是.o不是.c)

然后make drivers/char/qq2440\_hello.ko

编译就完成了，只需要几秒钟时间

参考链接:http://www.arm9home.net/read.php?tid-2670.html

2.编译nrf24l01驱动时出现警告：

warning: passing argument 2 of 'SPI\_Write\_Buf' discards qualifiers from pointer target type

note: expected 'uchar \*' but argument is of type 'const uchar \*'

解决方法：

主要是因为uchar SPI\_Write\_Buf(uchar reg, uchar \*pBuf, uchar uchars)定义的时候为uchar,而我真正在用这个函数的时候却实际赋值的是：uchar const，导致实际的不对应；改成一一对应即可；

2014.11.28:

1．编译第一个程序：

（1）编译模块：

现在Makefile里面修改添加上：obj-m +=NRF24L01.o;

在linux3.0.8下面执行make drivers/char/NRF24L01.o;生成NRF24L01.o文件；

接着，执行：make drivers/char/NRF24L01.ko; 生成NRF24L01.ko文件；

将ko文件利用ftp传入到开发板/lib/modules/3.0.8-FriendlyARM/;将权限改成777；

进入开发板/lib/modules/3.0.8-FriendlyARM/目录，在当前开发板上执行：insmod NRF24L01.ko

加载NRF24L01模块，再运行下列命令：

cat /proc/devices；

即可发现多出了“NRF24L01”这一行，找到它的主设备号后，接着运行：

mknod /dev/NRF24L01 c 241 0;(备注：241是驱动申请的主设备号，其它照常，c表示的是字符设备)

2014.11.29:

1.static定义的静态变量的作用是：

生命期延长，即相比自动变量而言，虽然局部函数已经失效，但是static定义的变量仍然存在，但是也不能引用，再次调用该函数时才能继续引用该静态变量；

2.关于如何卸载模块：

（1）使用命令：rmmod /proc/devices NRF24L01，应该是要加上/proc/devices才行；

对于杂项设备，需要这样子输入：rmmod mini210\_leds，就是把驱动后缀去掉，才能卸载，上一行的方法不行；

2014.11.30：

1.对于NRF24L01的理解：

（1）本机地址和接收地址，本机地址是指发送出去时对外宣称的本机地址，而不是要发送目标的目标地址；接收地址是指接收哪个地址的数据；

（2）在C语言的编程中，一个函数若带有返回值，则遇到return语句函数终止执行，已经退出了，所以不会执行函数后面的语句；

2014.12.1:

(1)在使用talnet时，发现串口能够输出很多信息，而利用talnet则不能显示很多信息，而且，在用talnet时竟然连plg主文件夹都显示不出原来的文件，只显示了NRF24L01.ko,和test两个文件，而用串口则显示正常，是网络的问题吗？之前都没遇到过；

2014.12.4:

1.驱动函数注册时有两种注册方式，各有特点：

参见：<http://blog.csdn.net/lanmanck/article/details/4713978>

杂项设备（misc device）

杂项设备也是在嵌入式系统中用得比较多的一种设备驱动。在 Linux 内核的include/linux目录下有Miscdevice.h文件，要把自己定义的misc device从设备定义在这里。其实是因为这些字符设备不符合预先确定的字符设备范畴，所有这些设备采用主编号10 ，一起归于misc device，其实misc\_register就是用主标号10调用register\_chrdev()的。

也就是说，misc设备其实也就是特殊的字符设备，可自动生成设备节点。

字符设备(char device)

使用register\_chrdev(LED\_MAJOR,DEVICE\_NAME,&dev\_fops)注册字符设备驱动程序时，如果有多个设 备使用该函数注册驱动程序，LED\_MAJOR不能相同，否则几个设备都无法注册(我已验证)。如果模块使用该方式注册并且 LED\_MAJOR为0(自动分配主设备号 )，使用insmod命令加载模块时会在终端显示分配的主设备号和次设备号，在/dev目录下建立该节点，比如 设备leds，如果加载该模块时分配的主设备号和次设备号为253和0，则建立节点:mknod leds c 253 0。使用register\_chrdev (LED\_MAJOR,DEVICE\_NAME,&dev\_fops)注册字符设备驱动程序时都要手动建立节点 ，否则在应用程序无法打开该设备。

2014.12.11:

(1)在C语言编程中，当定义了unsigned char时，如果我们在赋值的时候赋的是十进制的数，那么也就相当于赋值了一个字符，在ascall表中有一一对应关系，这时候重点就是要注意如果调用printk函数时要注意如果用%d则显示的是十进制的数，如果用%s显示的是字符串的格式，如果用%c显示的是字符的格式，以此类推；

（2）printk有时候显示有时候不显示的问题未解决，而且用dmesg也查看不到全部的信息；难道真的是没有执行到？需要再次检查；

首先，查看printk的等级命令如下：cat /proc/sys/kernel/printk，修改printk等级命令如下: echo "7 1 1 7" > /proc/sys/kernel/printk;如果需要查看所有的未打印的信息，输入命令：cat /proc/kmsg；

参考链接：http://blog.csdn.net/zhenwenxian/article/details/4336211

（3）已经调试好nrf24l01的驱动和应用程序，保存在[linux program\nrf24l01调试成功代码](linux%20program/nrf24l01调试成功代码)；

2014.12.13：

（1）今天开始重新书写usb 摄像头关于v4l2的驱动以及应用程序；

(2)在C语言当中，enum是表示的是枚举类型；查找资料有：<http://blog.csdn.net/logogcn/article/details/7879398>

2014.12.16:

(1)在C语言的文件编程中，运用哪个read和write函数需要参考不同的情况，参见教科书393页；

2014.12.18:

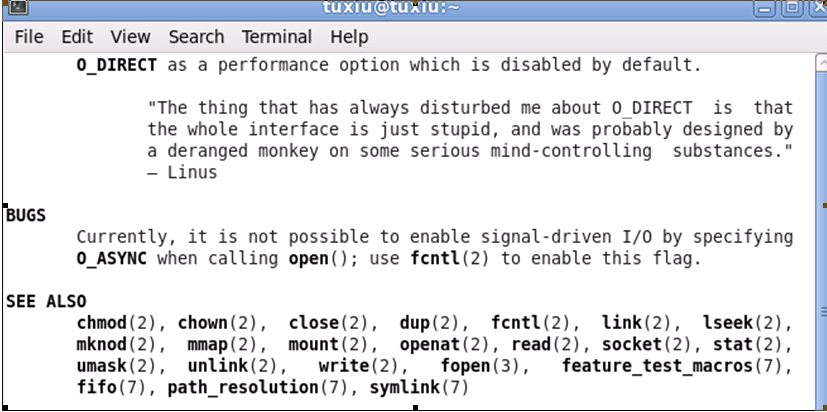
(1)对于misc的驱动应该是如何去卸载和动态加载呢，这点需要学习的；：

解决方法：将设备名字改成不同的，还是insmod加载；

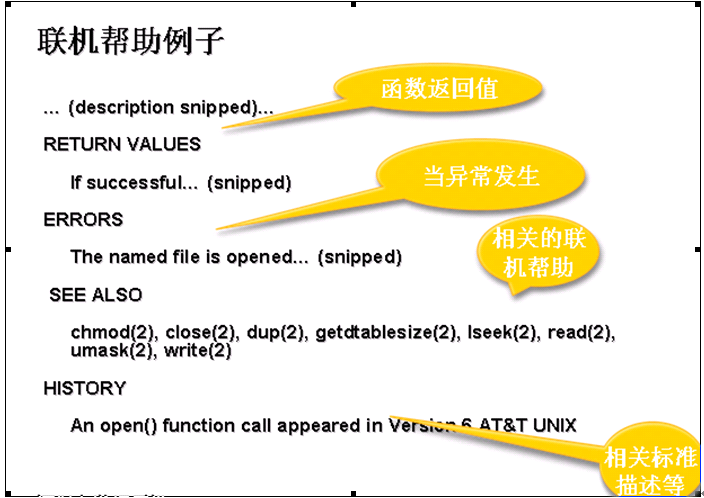
2014.12.19:

1.看帮助最重要看三点，一点是看函数描述和函数参数类型，二点是看包含的函数头文件，三点是看返回值，英文为return values;

2.如果命令只有一个含义，则可以直接man来查询帮助，如果该命令或者函数有多个含义，则这样子查询：man n　命令；如果ｔｉｍｅ（３）代表函数，则这样子输入：ｍａｎ　３　ｔｉｍｅ；即可实现查询帮助；



3.这个图标可以看成是个ｍａｎ帮助的结构解读：



4.#include <stdio.h>　表示存放在系统默认位置，即／ｕｓｒ／ｉｎｃｌｕｄｅ位置存放的；

#include “myinc.h”表示存放在当前其它位置；需要再次确认该定义；

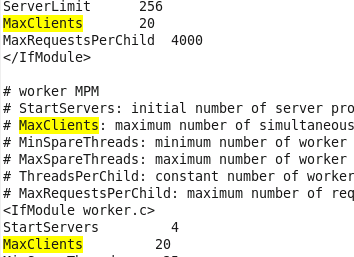
2014.12.21:

(1)今天开始配置apache服务器，现在记下来每一步所做的：

1.修改设定apache服务器的网页根目录: /home/zhangsan123456；



2.设置最大连接数：注意，这里有两个.c文件，貌似都要设置；



3.设置主机名：（前提是已经改成这个主机名了，要不然会出现错误）



4.设置为big5语言：





改成你想要的即可；

5.开始设置个人认证，参考鸟哥的804页，设置好后，我应该输入的是：

<http://luotuxiu2012130138:1080/protect/>，这里容易因为拼写错误而导致访问失败；

参照794页设置个人主页；设置好个人主页后，应该输入的是：<http://luotuxiu2012130138:1080/~luotuxiu2012130138/>

如果设置了/home/www为网页认证文件夹，则需要这样子输入：http://luotuxiu2012130138:1080/~luotuxiu2012130138/html/

(2)设置linuxz终端为护眼模式，如下：

点到color, 将background color颜色改为#CCE8CF即可；

2014.12.23:

1.如果需要将驱动程序编译进内核，那么需要linux-3.0.8目录下执行make menuconfig进行配置，配置选项有三个，功能分别如下：

  Y－-将该功能编译进内核

N－-不将该功能编译进内核

M－-将该功能编译成可以在需要时动态插入到内核中的模块 ；

接着,进行make，就能够在arch/arm/boot下编译出内核文件；

2.关于linux 出现内存错误：

参照网址：<http://www.cnblogs.com/justacoder/archive/2010/04/14/segmentation_fault.html>

造成segment fault，产生core dump的可能原因

1.内存访问越界

a) 由于使用错误的下标，导致数组访问越界

b) 搜索字符串时，依靠字符串结束符来判断字符串是否结束，但是字符串没有正常的使用结束符

c) 使用strcpy, strcat, sprintf, strcmp, strcasecmp等字符串操作函数，将目标字符串读/写爆。应该使用strncpy, strlcpy, strncat, strlcat, snprintf, strncmp, strncasecmp等函数防止读写越界。

2 多线程程序使用了线程不安全的函数。

3 多线程读写的数据未加锁保护。对于会被多个线程同时访问的全局数据，应该注意加锁保护，否则很容易造成core dump

4 非法指针

a) 使用空指针

b) 随意使用指针转换。一个指向一段内存的指针，除非确定这段内存原先就分配为某种结构或类型，或者这种结构或类型的数组，否则不要将它转换为这种结构或类型的指针，而应该将这段内存拷贝到一个这种结构或类型中，再访问这个结构或类型。这是因为如果这段内存的开始地址不是按照这种结构或类型对齐的，那么访问它时就很容易因为bus error而core dump.

5 堆栈溢出.不要使用大的局部变量（因为局部变量都分配在栈上），这样容易造成堆栈溢出，破坏系统的栈和堆结构，导致出现莫名其妙的错误。

例如，我的数组引用了越界的数据，就会这样子，必须注意；

2014.12.24:

1.今天查到210上面的video的说明如下：

S5PV210会在/dev/下生成几个video相关的设备节点，分别为

/dev/video0, /dev/video1, /dev/video2

/dev/video14, /dev/video21, /dev/video22

video0, video1, video2的作用是和Android camera/overlay HAL密切相关的，HAL层来决定这三个设备的具体作用，也就是说在设备驱动层并不会定义他们的具体作用。

/dev/video0 被android camera HAL用于capture和preview时，获取camera sensor数据

/dev/video1 被android overlay HAL用于视频输出，参见samsung/proprietary/liboverlay/overlay.cpp，overlay\_createOverlay函数，v4l2\_overlay\_open打开设备节点/dev/video1

/dev/video2 被android camera HAL用于录像时，获取camera sensor数据

/dev/video14 是TVout mixer的video设备

/dev/video21 是TVout overlay1

/dev/video22 是TVout overlay2

2014.12.25：

1.关于fprint的用法：

参考网址：http://baike.baidu.com/view/3233662.htm

【unix】标准输出(设备)文件，对应终端的屏幕。进程将从标准输入文件中得到输入数据，将正常输出[数据输出](http://baike.baidu.com/view/324739.htm)到标准输出文件，而将[错误信息](http://baike.baidu.com/view/627766.htm)送到标准错误文件中。在C中，程序执行时，一直处于开启状态。

可能的使用方法有：fprintf(stderr,"error message")

stderr -- 标准错误输出设备

stdout -- 标准输出设备 (printf("..")) 同 stdout。  
　　两者默认向屏幕输出。  
　　但如果用转向标准输出到磁盘文件，则可看出两者区别。stdout输出到磁盘文件，stderr在屏幕。  
　　例如：

fprintf(stderr, "Can't open it!\n");  
　　fprintf(stdout, "Can't open it!\n");

2.关于linux函数的退出方式提示：

EXIT\_FAILURE是[C语言](http://baike.baidu.com/view/1219.htm)头文件库中定义的一个符号常量，在vc++6.0下头文件stdlib.h中定义如下： #define EXIT\_FAILURE 1 .

EXIT\_FAILURE 可以作为exit()的参数来使用，表示没有成功地执行一个程序。

EXIT\_SUCCESS 作为exit()的参数来使用，表示成功地执行一个程序。

注意：记得包括头文件：stdlib.h

3.xioctl函数对于ioctl函数的封装：

参考：http://www.jayrambhia.com/blog/capture-v4l2/

#include <sys/ioctl.h>

**static** int xioctl(int fd, int request, void \*arg)

{

int r;

**do** r = ioctl (fd, request, arg);

**while** (-1 == r && EINTR == errno);

**return** r;

}

2014.12.26:

1如何查看linux系统版本号等信息：

方法一：命令：cat /proc/version

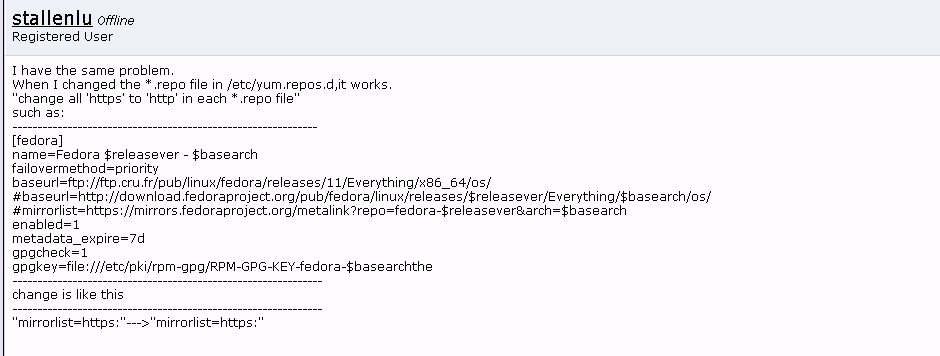
2今天在虚拟机时发现yum安装安装不了，出现的错误提示如下：

Error: Cannot retrieve repository metadata (repomd.xml) for repository: fedora. Please verify its path and try again

经过谷歌，找到解决方法：

参考网址：<http://forums.fedoraforum.org/showthread.php?t=223472>

修改文件：vi /etc/yum.repos.d/fedora.repo

正如最后一行所言（最后的https打错了，应该是http），将出现错误提示的段的https改成http就好了；

即可解决；

那么问题来了，我的另一台电脑也是一样的配置，为啥不可以呢？

3.如何开启linux的telnet服务：

对于fedora:

先yum install telnet-server;

然后就是修改配置文件：

vi /etc/xinetd.d/telnet   
service telnet   
{   
disable = yes   
flags = REUSE   
socket\_type = stream   
wait = no   
user = root   
server = /usr/sbin/in.telnetd   
log\_on\_failure += USERID

disable=no //仅修改了这里

#disable=yes  
}

4. stat函数讲解（转）

表头文件:    #include <sys/stat.h>  
            #include <unistd.h>  
定义函数:    int stat(const char \*file\_name, struct stat \*buf);  
函数说明:    通过文件名filename获取文件信息，并保存在buf所指的结构体stat中  
返回值:     执行成功则返回0，失败返回-1，错误代码存于errno

5.ioctl函数的返回值：

ioctl函数的返回值是一个整数类型的值，如果命令执行成功，ioctl返回零，如果出现错误，ioctl函数应该返回一个负值。这个负值会作为errno值反馈给调用此ioctl的用户空间程序。关于返回值的具体含义，请参考<linux/errno.h>和<asm/errno.h>头文件。

参考网址：<http://www.cnblogs.com/noaming1900/archive/2010/10/20/1856581.html>

2014.12.29:

1.下面介绍视频的相关知识点：

5.1.1 RGB与YUV格式简介 （1） RGB格式简介 RGB色彩模式是一种颜色标准，是通过对红(R)、绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样的颜色的。图像中每一个像素的RGB分量都分配一个0~255范围内的强度值。这个标准几乎包括了人类视力所能感知的所有颜色，主要应用在显示器上。 （2） YUV格式简介 YUV是一种颜色编码方法。Y'UV, YUV, YCbCr，YPbPr等专有名词都可以称为YUV。“Y”表示明亮度（Luminance、Luma），“U”和“V”则是色度、浓度（Chrominance、Chroma）。RGB诉求于人眼对色彩的感应，YUV则着重于视觉对于亮度的敏感程度。采用YUV色彩空间的重要性是它的亮度信号Y和色度信号U、V是分离的。如果只保留Y信号分量去掉U、V分量，那么这样表示的图像就是黑白灰度图像，从而实现使黑白电视机也能接收彩色电视信号。 YUV主要的采样格式有YCbCr 4:2:0、YCbCr 4:2:2、YCbCr 4:1:1和 YCbCr 4:4:4。因为摄像头常见的输入格式为YUV 4:2:2,所以本文主要介绍YUV4:2:2. YUV 4:2:2采样格式的每个色差信道的采样率是亮度信道的一半。例如两个像素点的Y,U,V值依次为：[Y0 U0 V0] [Y1 U1 V1]。采样后的码流为：Y0 U0 Y1 V1。 最后显示时的像素点又还原为：[Y0 U0 V1] [Y1 U0 V1]。显然舍弃了V0，U1，这样可以减少传输时占用的带宽，而且对画质并没有很大影响，因为人眼对彩度的敏感度远不如亮度，而亮度信息并没有减少。 （3） YUV与RGB格式的转换 本设计中需要将摄像头输出的YUV格式的图像数据，通过libjpeg压缩为JPEG。但是libjpeg不支持直接压缩YUV，所以本文先将YUV数据转换成RGB再调用libjpeg库来压缩成JPEG。 YUV转换为RGB只需要通过公式简单换算即可，具体换算方法见YUYV压缩为JPEG的函数compress\_yuyv\_to\_jpeg。

若是基于libjpeg库将YUYV图像转换为jpeg格式进行网络传输，则参考下列网址：

<http://wenku.baidu.com/link?url=ea_bUrwm2HyS0Rgn6NzBBuQLALSKld_bWG9Mbuj3lteXySknGinm0sGObIL4uhsPHtklk3Z-ovtkGQzPmMuSwmjSpo4CGjFlL8BYiBqLkva>

2. gzip: stdin: not in gzip format 解决办法

今天解压文件时出现如下问题：

# sudo tar zxvf ./jdk-7ull-linux-i586.tar.gz -C /usr/lib/jvm

gzip: stdin: not in gzip format

tar: Child returned status 1

tar: Error is not recoverable: exiting now

问题解决方法如下：将z参数换成j参数问题解决

bz2格式用j

gz格式用z

c是创建

x是解压缩

v是详细信息

f是指定文件

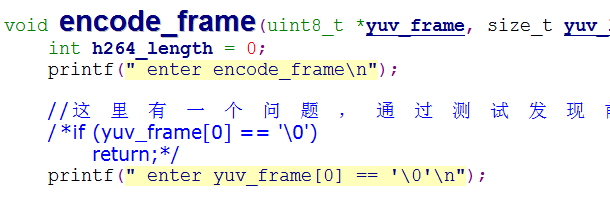
3.一开始在开发板上面提示找不到:

./camera\_h264encode: error while loading shared libraries: libx264.so.122: cannot open shared object file: No such file or directory

所以我将libx264.so.122放到/usr/lib，解决；

还有一定需要注意：最后面的122要和程序的对应，要不然会报错，所以有可能网络传输后解码会有问题，因为ubuntu的貌似是140的（编译的时候发现它提示\_140错误）；

4.代码错误在这里：



初步判断是内存非法使用导致程序退出；

2014.12.30：

1.make时提示"make: 警告:检测到时钟错误。您的创建可能是不完整的"的警告；原因是有文件的时间比当前时间还要晚，解决方法：（对该文件夹或者文件使用）

[root@luo camera]# touch .

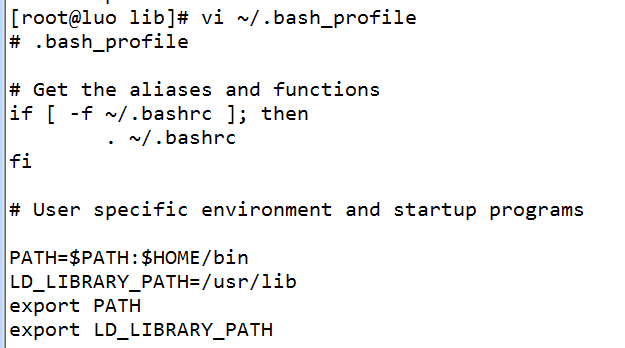
2.今天开始搭建gdb的环境，尝试找出段错误的地方；

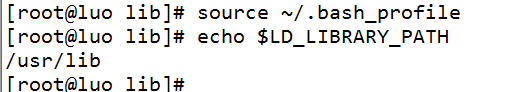
参照网址：<http://www.arm9home.net/read.php?tid=1396>

在开发板上输入：gdbserver 172.31.101.222:2012 /home/plg/camera/bin/camera\_h264encode（172.31.101.222对应的是linux的地址，/home/plg/camera/bin/camera\_h264encode表示要调试的那个可执行文件）

搭建好后，在linux上输入arm-linux-gdb /home/luo/camera/bin/camera\_h264encode（/home/plg/camera/bin/camera\_h264encode表示要调试的那个可执行文件，这个文件是在linux上的），进入gdb调试模式后，再输入target remote 172.31.101.180:2012（172.31.101.180对应着开发板的ip）；初步即可调试；

3.添加共享库的寻找命令如下：





4.在用gdb调试时发现出现错误：

Error while mapping shared library sections:

/lib/ld-linux.so.3: No such file or directory.

解决方法：

参考网址：<http://blog.csdn.net/harry_helei/article/details/5740456>，安装完毕后，成功；

5. GDB的基本指令：   
load：装入一个程序   
symbol-file：装入符号库文件，可以是用-g参数编译的可执行文件。   
f(ile)：指定一个可执行文件进行调试，gdb将读取些文件的调试讯息，如f a.exe   
l(ist)：列程序出源文件   
r(un) ：装载完要调试的可执行文件后，可以用run命令运行可执行文件   
b(reak)：设置断点（break point），如b 25，则在源程序的第25行设置一个断点，当程序执行到第25行时，就会产生中断；也可以使用b funcname，funcname为函数的名称，   
当程序调用些函数时，则产生中断   
c(ontinue)：c命令可以另中断的程序继续执行，直到下一个中断点或程序结束   
p(rint)：输入某个变量的值，如程序定义了一个int aa的就是，p aa就会输出aa的当前值   
n(ext)：程序执行到断点时中断执行，可以用n指令进行单步执行   
s(tep)：程序执行到断点时中断执行，可以用s指令进行单步执行进某一函数   
q(uit)：退出GDB；

2015.1.4：

1.今天继续完成linux的期末大设计：

（1）学习wireshark来分析网络协议：

参考网址：<http://www.cnblogs.com/tankxiao/archive/2012/10/10/2711777.html>

# 封包详细信息 (Packet Details Pane)

这个面板是我们最重要的，用来查看协议中的每一个字段。

各行信息分别为

Frame:   物理层的数据帧概况

Ethernet II: 数据链路层以太网帧头部信息

Internet Protocol Version 4: 互联网层IP包头部信息

Transmission Control Protocol:  传输层T的数据段头部信息，此处是TCP

Hypertext Transfer Protocol:  应用层的信息，此处是HTTP协议

（2）完成6.1题

Uniq的用法：

参考网址：<http://blog.51yip.com/shell/1022.h>

文本中的重复行，基本上不是我们所要的，所以就要去除掉。linux下有其他命令可以去除重复行，但是我觉得uniq还是比较方便的一个。使用uniq的时候要注意以下二点

1，对文本操作时，它一般会和sort命令进行组合使用，因为uniq 不会检查重复的行，除非它们是相邻的行。如果您想先对输入排序，使用sort -u。

2，对文本操作时，若域中为先空字符(通常包括空格以及制表符)，然后非空字符，域中字符前的空字符将被跳过

1. [zhangy@BlackGhost ~]$ uniq --help
2. 用法：uniq [选项]... [文件]
3. 从输入文件或者标准输入中筛选相邻的匹配行并写入到输出文件或标准输出。
5. 不附加任何选项时匹配行将在首次出现处被合并。
7. 长选项必须使用的参数对于短选项时也是必需使用的。
8. -c, --count              //在每行前加上表示相应行目出现次数的前缀编号
9. -d, --repeated          //只输出重复的行
10. -D, --all-repeated      //只输出重复的行，不过有几行输出几行
11. -f, --skip-fields=N     //-f 忽略的段数，-f 1 忽略第一段
12. -i, --ignore-**case**       //不区分大小写
13. -s, --skip-chars=N      //根-f有点像，不过-s是忽略，后面多少个字符 -s 5就忽略后面5个字符
14. -u, --unique            //去除重复的后，全部显示出来，根mysql的distinct功能上有点像
15. -z, --zero-terminated   end lines with 0 byte, not newline
16. -w, --check-chars=N      //对每行第N 个字符以后的内容不作对照
17. --help              //显示此帮助信息并退出
18. --version              //显示版本信息并退出

2015.2.5:

1.fedora14的miniCOM出现了问题，在谷歌后找到一点解决方案，如下：

在用户账号下输入sudo minicom -S ttyS0 –o，即可进入超级终端；

2015.3.5:

1.发现一个现象，插入usb摄像头后，程序中使用video0或者video3都是一样的效果，这是为什么呢？

2.不管我在main函数或者在设置fmt那个结构体的变量的宽度和长度的时候，该摄像头都会自动设置为176\*144的大小，可是上个学期的时候我是可以根据程序来调整图片宽度的，这又是为什么呢？只有我在设置为160\*120的时候，它就会调整为160\*120，这样在压缩为h.264时才不会导致出现段错误，内存错误退出。

3.更换了网站上面新的内核后也没有改善实验效果。

2015.3.12:

1.在Ubuntu下重启samba的正确命令是：/etc/init.d/smbd start，网上面许多写的/samba是错误的；

2.vs对应的版本如下：



注意不要搞错；

2015.3.15:

1.vs2012更改字体参考网站：

<http://www.cnblogs.com/lushuicongsheng/archive/2012/07/17/2595006.html>

2.\_T的作用是：\_T是一个宏，作用是让你的程序支持Unicode编码

用法：[wchar\_t](http://baike.baidu.com/view/2063825.htm)Str[] = L"Hello World!";

2015.3.20:

1.在mfc界面输出printf信息需要加入一下步骤：

## 3）将标准输出定向到自己创建的控制台

      #include "io.h"  
      #include "fcntl.h"

        void InitConsole()  
      {  
            int nRet= 0;  
           FILE\* fp;  
           AllocConsole();  
           nRet= \_open\_osfhandle((long)GetStdHandle(STD\_OUTPUT\_HANDLE), \_O\_TEXT);  
           fp = \_fdopen(nRet, "w");  
          \*stdout = \*fp;  
          setvbuf(stdout, NULL, \_IONBF, 0);  
    }

  将此函数在MFC程序初始化的地方调用，即可使用控制台查看printf函数的打印信息

2015.05.08

1. **typedef与#define的区别**

              typedef：类型定义，其功能是用户为已有数据类型取“别名”。

              如：typedef  int  INT; 意思是将int重新定义为INT，以后使用INT a；就相当于int       a；

       用typedef定义数组、指针、结构等类型将带来很大的方便，不仅使程序书写简单，而且使意义更为明确，因而增强了可读性。例如：typedef int a[10]；表示a是整型数组类型，数组长度为10，然后就可用a定义变量，如：a s1，s2；完全等效于：int s1[10]，s2[10]；

define：宏定义。

       如：#define PI 3.14 意思是以后程序中出现PI的地方将用3.14代替，这个替换是在编译预处理阶段完成的，注意#define最后没有分号，否则编译时将分号一同带入PI中。

2015.07.31

1.linux下删除文件夹：

rm命令

-r 递归删除

-I 互动模式，提示是否删除

-f force的意思，忽略不存在的文件，不会出现警告信息；

2.