

基于 V4L2 的远程图片采集系统

张聪敏, 游向东

北京邮电大学信息与通信工程学院, 北京 (100876)

E-mail: superthinker2@163.com

摘 要: 本文对 linux 系统下的 V4L2 进行了深入的研究, 给出了使用 V4L2 进行图片采集的具体过程, 包括打开设备、设置图片格式、分配缓冲区、读取数据、关闭设备等五个步骤。同时介绍了 linux socket 编程, 并在此基础上, 给出了一种基于 V4L2 的远程图片采集系统。该系统采用 C/S 结构, 服务器端可以运行在连有图片采集设备的嵌入式终端上, 负责图片的采集、处理和传输。用户通过运行在 PC 上面的终端与服务器交互, 通过该系统实时采集高质量的监控图片。该系统既可以独立使用, 也可以作为监控程序的扩展插件使用。

关键词: Linux 系统; V4L2; 视频监控; 网络 socket; 图片捕获

中图分类号: TN919.81

1 引言

随着社会的发展和生活水平的提高, 视频监控广泛应用于生活中的各个角落。Linux 以其开源、稳定、良好的移植性、优秀的网络功能、对各种文件系统完备的支持等特点被广泛应用于各种嵌入式设备的开发中, 包括视频监控设备和视频通信设备等。V4L2 作为 linux 系统下的一种通用视频架构应用于许多视频设备之中。

本文正是基于 linux 系统下的 V4L2 实现了一种远程图片采集系统, 用户可以通过该系统动态实时采集重要的监控图片, 并传输到用户 PC 机, 该方法确保了图片的高质量。该系统可以独立使用, 也可以作为监控程序的扩展插件使用。

2 V4L2 介绍

2.1 概述

V4L(Video for Linux)是 Linux 内核中关于视频设备的子系统, 它为 linux 下的视频驱动提供了统一的接口, 使得应用程序可以使用统一的 API 函数操作不同的视频设备, 极大地简化了视频系统的开发和维护。

由于 V4L 有很多缺陷, Bill Dirks 等人对其进行了重新设计, 并取名为 Video for Linux Two(V4L2), 最早出现于 Linux2.5.x 版本。V4L2 相比于 V4L 有更好的扩展性和灵活性, 并且支持的硬件设备更多。但是需要注意的是, V4L2 对 V4L 进行了彻底的改造, 因而两者并不兼容^[1]。

Linux 系统中, 所有的外部设备都被看成一种特殊的文件, 称之为设备文件。应用程序可以通过访问这些特殊文件实现对应设备的控制。V4L2 视频设备的设备文件为/dev 目录下的 videoN(N 为 0~63)文件, 其主设备号为 81, 次设备号为 N(N 为 0~63)。

2.2 ioctl 系统调用

V4L2 的绝大部分功能是通过 ioctl 系统调用完成的, 其语法为:

ioctl(int fd, int request, void *argp)

其中 fd 为设备文件描述符, 通过 open()函数获得; request 为系统调用类型, 用于告诉系统要做什么; argp 为用户数据指针, 用于传递参数或接收数据。以下是几种常用的命令类型, 这些类型都在 videodev.h 中进行了定义:

VIDIOC_G_FMT/VIDIOC_S_FMT, 获取/设置数据格式, 主要是图片的宽高、每像素所占 bit 和像素格式 (比如 YUV4:2:0 或 RGB24 等)。数据类型为 struct v4l2_format。

VIDIOC_G_CROP/VIDIOC_S_CROP, 获取/设置图片剪裁矩形框, 主要是设置相机的视角大小和位置。数据类型为 struct v4l2_crop。

VIDIOC_REQBUFS, 用于内存映射方式, 该命令可以在驱动层申请多个缓冲区存放采集到的图片。需要注意的是用户需要使用 mmap 将其映射到用户空间才能访问。数据类型为 v4l2_requestbuffers。

VIDIOC_QUERYBUF, 用于内存映射方式, 该命令可以查询由 VIDIOC_REQBUFS 申请到的缓冲区的大小和偏移地址, 这两个参量会作为参数传给 mmap。数据类型为 struct v4l2_buffer。

VIDIOC_QBUF, 用于内存映射方式, 该命令的作用是将 VIDIOC_REQBUFS 申请的缓冲区放入采集队列, 只有这样驱动才会将采集到的数据写入缓冲区。数据结构为 struct v4l2_buffer。

VIDIOC_DQBUF, 用于内存映射方式, 该命令的作用是将填充数据的一个缓冲区取出队列, 这样驱动就不会再向这个缓冲区写数据, 应用程序可以对该缓冲区的数据进行处理。需要注意的是, 应用程序处理完数据后, 需要使用 VIDIOC_QBUF 重新将该缓冲区放入采集队列。

VIDIOC_STREAMON/VIDIOC_STREAMOFF, 这两个命令用于启动/停止视频数据的采集或输出。数据结构为 enum v4l2_buf_type。

2.3 V4L2 数据交换模式

V4L2 支持三种数据交换模式, 分别是: 直接读取设备文件方式、用户指针方式以及 mmap 映射方式。

(1) 直接读取设备文件方式

直接调用 read()、write() 函数进行数据的读入和输出, 该方法一般配合 select() 使用。

(2) 用户指针方式

首先由应用程序申请一段缓冲区, 然后将缓冲区传给驱动, 驱动将其作为缓冲区, 从而实现了内存共享。

(3) mmap 映射方式

这种方式与用户指针方式类似, 只是采用了逆向操作。首先在内核空间申请多个缓冲区, 然后将每个缓冲区通过 mmap 映射到用户空间, 这样驱动和应用程序共享这些缓冲区, 在进行数据处理时不需要进行拷贝, 大大提高了效率。本系统正是采用的这种方式, 其映射过程如图 1 所示:

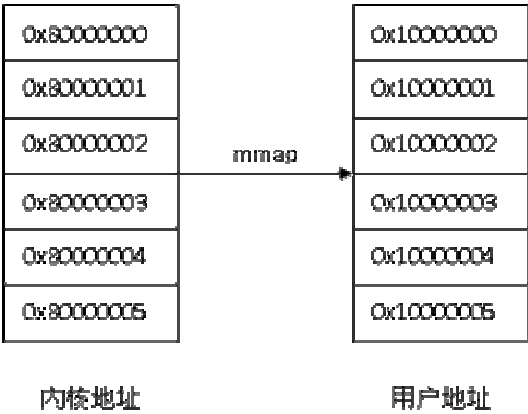


图 1 mmap 映射示意图

3 远程图片采集系统设计

3.1 设计方案

如下方图 2所示，本系统采用 C/S 结构，服务器端可以运行在连有视频采集设备的嵌入式终端上，负责图片的采集、处理和传输。用户通过运行在 PC 上面的终端与服务器交互。系统的工作过程如下：

- (1)由用户在客户端输入“capture”指令；
- (2)指令通过网络接口传输到服务器；
- (3)服务器的命令解析服务对字符串形式的指令进行解析，获得服务接口；
- (4)通过服务接口调用 V4L2 图片采集接口，采集图片；
- (5)调用 jpeg 压缩接口压缩图片^[2]；
- (6)将处理后的图片通过网络接口传回客户端，并存储为文件。

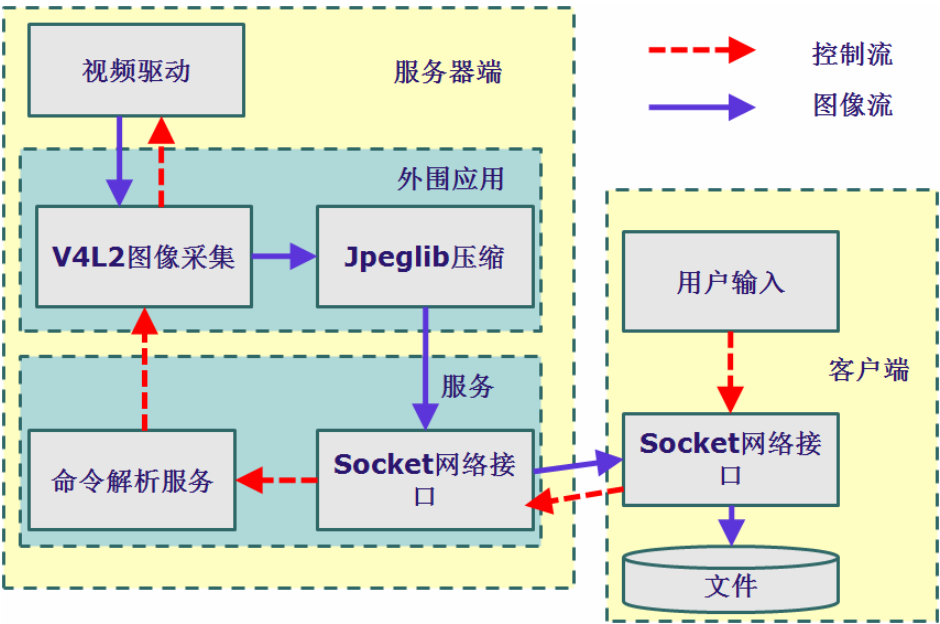


图 2 远程图片采集系统结构

该结构具有很强的扩展性，可以根据需要添加新的外围应用，可以通过客户端控制服务器，下面重点介绍服务器端程序。

3.2 V4L2 图片采集

V4L2 图片采集过程分为打开设备、设置图片格式、分配缓冲区、读取数据、关闭设备等步骤。

1) 打开设备, 获得文件描述符

```
int fd=open("/dev/video0", O_RDWR,0)
```

应用程序能够使用阻塞模式或非阻塞模式打开视频设备。如果使用非阻塞模式, 即使数据还未准备充分, 驱动依旧会把缓存的数据返回给应用程序, 但是缓存的数据可能并非所需有效数据, 这样可能造成错误。如果使用阻塞模式, 只有数据准备充分时驱动才可以把缓存的数据返回给应用程序。该系统保持默认的阻塞模式^[3]。

2) 设置图片格式

将图片宽设为 640, 高设为 480, 像素格式设为 YUYV。其他参数保持默认。

```
struct v4l2_format fmt;
memset ( &fmt, 0, sizeof(fmt) );
fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
fmt.fmt.pix.width = 640;
fmt.fmt.pix.height = 480;
fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUYV;
if (ioctl(fd, VIDIOC_S_FMT, &fmt) == -1)
{
    return -1;
}
```

3) 分配缓冲区

首先向驱动申请三个缓冲区, 为 mmap 映射做准备。

```
struct v4l2_requestbuffers req;
req.count = 3; //申请三个缓冲区
req.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
req.memory = V4L2_MEMORY_MMAP;
if (ioctl(fd, VIDIOC_REQBUFS, &req) == -1)
{
    return -1;
}
```

然后通过调用 `ioctl(fd, VIDIOC_QUERYBUF, &buf)` 获得缓冲区的长度(`buf.length`)和偏移地址(`fd, buf.m.offset`), 将这两个参量作为参数传给 `mmap` 函数 `buffers[num].start = mmap(NULL, buf.length, PROT_READ | PROT_WRITE, MAP_SHARED, fd, buf.m.offset);`

其中 `buffers` 的定义为

```
struct
{
    void *start;
    int length;
}buffers[MAX_BUFFER_CNT];
```

该结构用于管理缓冲区在用户空间的地址。应用程序对 `start` 所指向的存储区进行操作等同于对视频缓冲区进行操作。

缓冲区分配好了以后, 驱动并不会向里面写入数据, 还需要将每个缓冲区放入视频采集队列。

```
ioctl(fd, VIDIOC_QBUF, &buf);
```

4) 启动采集过程，读取数据

以上步骤完成了视频采集的准备工作，但驱动还没有启动采集过程，应用程序需要调用 VIDIOC_STREAMON ioctl 系统调用驱动才会开始采集数据。

```
enum v4l2_buf_type type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
```

```
ioctl (fd, VIDIOC_STREAMON, &type);
```

采集过程开始以后，驱动会不停地将数据写入分配的缓冲区内，当一个缓冲区的数据准备就绪后，驱动就会将其放入输出队列，等待应用程序的处理。当所有的缓冲区都进入输出队列后，驱动将停止采集，并等待缓冲区重新放入采集队列。

读取数据时，首先需要将一个缓冲区出队列：

```
struct v4l2_buffer buf;
```

```
ioctl (fd, VIDIOC_DQBUF, &buf);
```

驱动会从输出队列取出一个缓冲区，并将其序号赋值给 `buf.index`，应用程序可以通过 `buffers[buf.index].start` 来访问缓冲区的数据。当处理完成后，需要将其重新放入采集队列。

```
ioctl(fd, VIDIOC_QBUF, &buf);
```

5) 停止采集

首先停止采集过程 `ioctl (fd, VIDIOC_STREAMOFF, &type)`，然后使用 `munmap` 函数删除映射，最后调用 `close(fd)` 函数关闭设备。

3.3 数据传输

3.3.1 Socket 介绍

数据传输模块是由 linux socket 实现的，网络的 Socket 数据传输是一种特殊的 I/O，Socket 也是一种文件描述符。Socket 也具有一个类似于打开文件的函数调用 `Socket()`，该函数返回一个整型的 Socket 描述符，随后的连接建立、数据传输等操作都是通过该 Socket 实现的。常用的 Socket 类型有两种：流式 Socket (`SOCK_STREAM`) 和数据报式 Socket (`SOCK_DGRAM`)。流式 Socket 是一种面向连接的 Socket，针对于面向连接的 TCP 服务应用；数据报式 Socket 是一种无连接的 Socket，对应于无连接的 UDP 服务应用^[4]。

Socket 工作分为两种模式：阻塞和非阻塞，阻塞模式下，当系统完成指定任务之前，程序不能够继续执行后面的语句。例如，服务器运行到 `accept` 语句时，而没有客户连接服务请求到来，服务器就会停止在 `accept` 语句上等待连接服务请求的到来。而实际过程中，我们需要轮询多个 socket，因而不能够阻塞到某一个 socket 上，因而需要设置成非阻塞模式，但是这种“轮询”会使 CPU 处于忙等待方式，从而降低性能，浪费系统资源。Linux 提供了 `select` 函数很好的解决了这个问题，它允许把进程挂起，同时使系统内核监听所要求的一组文件描述符，当某一文件描述符准备就绪时，`select()` 调用将返回指示该文件描述符已准备好的信息。

3.3.2 实现方案

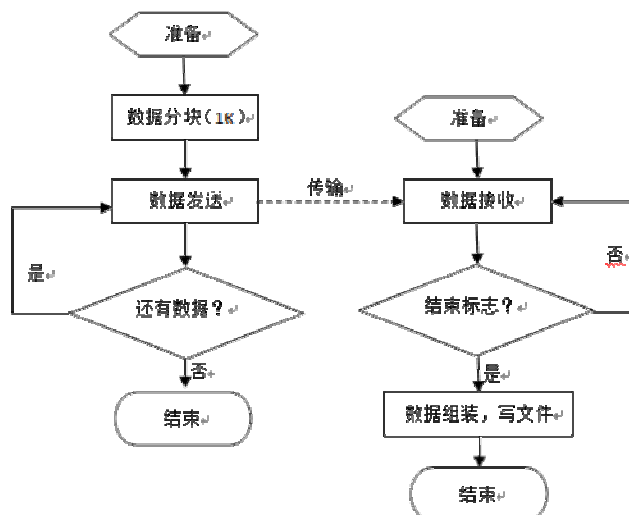


图 3 数据传输/接收过程

如图 3所示, 数据传输过程中, 服务器端首先将数据分块, 每数据单元最大为 1k, 数据发送完成时, 会发送一个结束标志, 表示数据传输完成。客户端收到数据后会先缓存起来, 当收到结束标志时才将数据组装并写入文件。

4 结束语

从以上的讨论来看, 本系统实现方法简单, 扩展性强, 具有很好的移植性。一般的视频监控系统中, 由于磁盘空间和网络带宽的限制, 监控视频的图像分辨率不够高。本系统存在的意义在于用户可以实时获得分辨率较高的重要监控图片, 作为一种补充配合视频监控系统工作。

参考文献

- [1] Michael H Schimek, Video for Linux Two API Specification Revision 0.24[EB/OL], <http://v4l2spec.bytesex.org/v4l2spec/v4l2.pdf>, 2008-03-04.
- [2] 蔡安妮.《多媒体通信技术基础》.北京: 电子工业出版社, 2008 年 7 月。
- [3] 王剑非, 孟彦杰, 赵勇. 基于 Linux 操作系统的视频采集卡驱动程序设计.《微计算机信息》(嵌入式 SOC) 2007 年第 23 卷第 1-2 期。
- [4] 李振德.基于 H_264 编码标准的远程视频采集系统的设计与实现. 湛江海洋大学学报. 2006 年 6 月第 26 卷第 3 期。

The system of image capturing over long-distance based on V4L2

Zhang Congmin, You Xiangdong

School of Information and Telecommunication engineering in Beijing University of Posts and Telecommunications, Beijing (100876)

Abstract

This paper conducts the thorough research about V4L2 in linux system, and provide the process of image capturing with V4L2, which includes open the device、configure the format of the image、alloc buffers、read data and finally close the device. Meanwhile, this paper introduces the linux socket programming, and based on this, we provide a system of image capturing over long-distance, which have c/s structure. The server, which could run on the embedded terminal with image capturing device, could capture、process and transfer the images. The user could capture real-time images of high

quality through the client terminal on the PC. The system could be used as independent application or plug-in tool kit.

Key Words: Linux system;V4L2;Video monitor;Network socket;image capture

作者简介:

张聪敏 北京邮电大学硕士研究生 研究方向为嵌入式移动终端与多媒体信息处理, 单位:北京邮电大学 信息与通信工程学院。电子邮箱:superthinker2@163.com

游向东 硕士生导师 高级工程师 研究方向为嵌入式移动终端与多媒体信息处理, 单位:北京邮电大学, 电子邮箱:youxiaangdong@263.net 。