

本文链接：https://blog.csdn.net/coroutines/article/details/70141086

可参考：<http://www.it610.com/article/4522348.htm> [//v4l2官方翻译](#)

基于V4L2的应用，通常面临着大块数据的读取与拷贝等问题。尤其在 **嵌入式** 系统中，对于实时性能要求较高的应用，拷贝会花上几十个ms的时间，这通常轻则造成用户体验差，重则导致产品质量不达标。V4L2 Framework定义了三种不同的方式，用于从设备中读取数据，这篇文章简要介绍下在Streaming I/O模式下，如何使用这几种数据的获取与使用方法。Streaming I/O设计的目的是为了减少在数据处理的各个环节中，拷贝的次数，从而实现各阶段硬件的无缝配合。本文针对的是USB Camera (Capture)设备。

1. 设备支持

对于设备特性来说，需要设备支持Streaming能力，这个需要通过V4L2的Capability来判断，方式如下：

```
struct v4l2_capability _cap;
ioctl(_fd, VIDIOC_QUERYCAP, &_cap);
```

其中，_fd是打开的V4L2设备的描述符，通过：

```
if(_cap.capabilities & V4L2_CAP_STREAMING) != 0 {
    printf(" V4L2_CAP_STREAMING");
}
```

判断是否支持Streaming方式访问。
2 Memory Map

通过Memory Map访问V4L2设备驱动中分配的内存。设备收到的数据存在驱动内的Buffer中，通过Map方式，将内存Map到用户空间。使用这种方式只有指向这段内存的用户空间指针在各个处理环节中传递，不会发生真实的数据拷贝。

```
struct v4l2_requestbuffers reqbuf;
struct {
    void *start;
    size_t length;
} *buffers;
unsigned int i;

memset(&reqbuf, 0, sizeof(reqbuf));
reqbuf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
reqbuf.memory = V4L2_MEMORY_MMAP;
reqbuf.count = 20;

if (-1 == ioctl(fd, VIDIOC_REQBUFS, &reqbuf)) {
    if (errno == EINVAL)
        printf("Video capturing or mmap-streaming is not supported\n");
    else
        perror("VIDIOC_REQBUFS");

    exit(EXIT_FAILURE);
}
```

/* We want at least five buffers. */

```
if (reqbuf.count < 5) {
    /* You may need to free the buffers here. */
    printf("Not enough buffer memory\n");
    exit(EXIT_FAILURE);
}
```

```
buffers = calloc(reqbuf.count, sizeof(*buffers));
assert(buffers != NULL);
```

```
for (i = 0; i < reqbuf.count; i++) {
    struct v4l2_buffer buffer;

    memset(&buffer, 0, sizeof(buffer));
    buffer.type = reqbuf.type;
    buffer.memory = V4L2_MEMORY_MMAP;
    buffer.index = i;

    if (-1 == ioctl(fd, VIDIOC_QUERYBUF, &buffer)) {
        perror("VIDIOC_QUERYBUF");
        exit(EXIT_FAILURE);
    }

    buffers[i].length = buffer.length; /* remember for munmap() */

    buffers[i].start = mmap(NULL, buffer.length,
        PROT_READ | PROT_WRITE, /* recommended */
        MAP_SHARED, /* recommended */
        fd, buffer.m.offset);

    if (MAP_FAILED == buffers[i].start) {
        /* If you do not exit here you should unmap() and free()
        the buffers mapped so far. */
        perror("mmap");
        exit(EXIT_FAILURE);
    }
}
```

```
buffers[i].length = buffer.length; /* remember for munmap() */

buffers[i].start = mmap(NULL, buffer.length,
    PROT_READ | PROT_WRITE, /* recommended */
    MAP_SHARED, /* recommended */
    fd, buffer.m.offset);

if (MAP_FAILED == buffers[i].start) {
    /* If you do not exit here you should unmap() and free()
    the buffers mapped so far. */
    perror("mmap");
    exit(EXIT_FAILURE);
}
}
```

/* Cleanup. */

```
for (i = 0; i < reqbuf.count; i++)
    munmap(buffers[i].start, buffers[i].length);
```

用户程序首先需要通过VIDIOC_REQBUFS，通知驱动进行驱动内存分配；之后通过VIDIOC_QUERYBUF取得驱动中各内存块的基本信息，主要是Buffer长度和Offset；取得这些信息后，通过mmap，将内存map到User Space中使用。在驱动中，V4L2对于Buffer的使用是队列形式，Buffer出队后，再次入队之前，驱动无法再使用这个Buffer，因此申请的Buffer个数通常是多个，避免数据丢失。在使用过程中，需要通过poll操作，判断是否有数据到达，然后通过VIDIOC_DQBUF，取得当前有数据的Buffer，通过Buffer的Index属性，找到对应的User Space指针，交由下个环节处理；处理完成后，通过VIDIOC_QBUF，将Buffer重新入队。

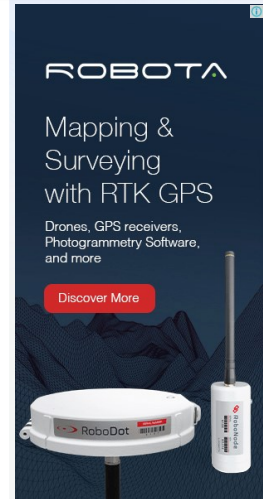
3 User Pointers

User Pointers方式会将用户空间分配的内存指针及长度传递给V4L2驱动（虽然是在用户空间分配，但不一定是在堆空间上分配的内存，可以是通过其它方式映射出来的内存，比如从另一个设备驱动中），这样数据到达后，可以直接传递这个指针到下个环节中处理。初始化方式如下：

```
struct v4l2_requestbuffers reqbuf;

memset(&reqbuf, 0, sizeof(reqbuf));
reqbuf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
reqbuf.memory = V4L2_MEMORY_USERPTR;

if (ioctl(fd, VIDIOC_REQBUFS, &reqbuf) == -1) {
    if (errno == EINVAL)
        printf("Video capturing or user pointer streaming is not supported\n");
    else
        perror("VIDIOC_REQBUFS");
}
```



分类专栏		
	Camera_tuning	7篇
	AF	6篇
	Camera_driver	5篇
	V4l2	3篇
	Android_OS	
	Adroid HIDL	2篇
	Linux	8篇
	Linux-MMC子系统	5篇
	Linux-spi	2篇
	linux-环境配置	8篇
	语言	3篇

```
exit(EXIT_FAILURE);
}

4 DMA buffer importing

DMA方式的初始化方式如下：

struct v4l2_requestbuffers reqbuf;

memset(&reqbuf, 0, sizeof (reqbuf));
reqbuf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
reqbuf.memory = V4L2_MEMORY_DMABUF;
reqbuf.count = 1;

if (ioctl(fd, VIDIOC_REQBUFS, &reqbuf) == -1) {
    if (errno == EINVAL)
        printf("Video capturing or DMABUF streaming is not supported\n");
    else
        perror("VIDIOC_REQBUFS");

    exit(EXIT_FAILURE);
}

DMA方式的传输基于文件描述符进行，fd的传递是通过VIDIOC_QBUF中的描述符设置：

int buffer_queue(int v4lfd, int index, int dmafd)
{
    struct v4l2_buffer buf;

    memset(&buf, 0, sizeof buf);
    buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
    buf.memory = V4L2_MEMORY_DMABUF;
    buf.index = index;
    buf.m.fd = dmafd;

    if (ioctl(v4lfd, VIDIOC_QBUF, &buf) == -1) {
        perror("VIDIOC_QBUF");
        return -1;
    }

    return 0;
}

poll操作返回后，可以通过这个dmafd进行下一步处理。
```

DMA方式的传输基于文件描述符进行，fd的传递是通过VIDIOC_QBUF中的描述符设置：

```
int buffer_queue(int v4lfd, int index, int dmafd)
{
    struct v4l2_buffer buf;

    memset(&buf, 0, sizeof buf);
    buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
    buf.memory = V4L2_MEMORY_DMABUF;
    buf.index = index;
    buf.m.fd = dmafd;

    if (ioctl(v4lfd, VIDIOC_QBUF, &buf) == -1) {
        perror("VIDIOC_QBUF");
        return -1;
    }

    return 0;
}

poll操作返回后，可以通过这个dmafd进行下一步处理。
```

Native层打开 V4L2 video0摄像头-V4L2_MEMORY_DMABUF-YUYV格式-1plane.cpp11-14

Native层打开 V4L2 video0摄像头-V4L2_MEMORY_DMABUF-YUYV格式-1plane.cpp 本文链接：《[Native层打开 V4L2 video0摄像头-V4L2_MEMORY_D...

V4L2文档翻译（十三）Feed Technology 4780

http://linuxtv.org/downloads/v4l-dvb-apis/dmabuf.html I/O流（DMA缓存引用） 这是一个实验性接口，将来可能发生改变 DMABUF框架提供了在多设备见...

评论 1 条 >

hehui0921 热评 int buffer_queue(int v4lfd, int index, int dmafd) 这个是哪里调用的？

linux v4l2架构分析之用户层应用编程 最新发布ismyscdn的博客 41

本文详细介绍了4l2架构的应用程序编程的具体流程。

学习笔记，英伟达api，DMA内存使用流程，例程12lord1st的博客 813

一、使用jetson nano 提供的结构体context_t_1，初始化变量ctx。 typedef struct { /* Camera v4l2 context */ const char * cam_devname;//相机接口名 char ...

V4L2视频驱动tiangwan2011的专栏 2384

V4L (video4linux是一些视频系统，视频软件、音频软件的基础，经常时候在需要采集图像的场所，如视频监控，webcam,可视电话，经常使用在embedd...

基于Streaming I/O的V4L2设备使用认真你就输了 3786

基于V4L2的应用，通常面临着大块数据的读取与拷贝等问题。尤其在嵌入式系统中，对于实时性能要求较高的应用，拷贝会花上几十个ms的时间，这通...

Native层打开 V4L2 video0摄像头-V4L2_MEMORY_DMABUF-YUYV格式-1plane [一--热爱生活、努力学习的小伙伴--] 3295

Native层打开 V4L2 video0摄像头-V4L2_MEMORY_DMABUF-YUYV格式-1plane一、配置驱动准备DMA 类型Buffer二、应用层申请DMA Buffer三、申请D...

深入学习Linux摄像头（一）v4l2应用编程u010783226的专栏 5904

v4l2 use V4L2_MEMORY_MMAPP方式导出为 DMA BUF fd 方式使用huifeidedabian的博客 2003

V4L2_MEMORY_MMAPP 导出 fd 需要使用 vb2_ioctl_expbuf (只能用于VB2_MEMORY_MMAPP 方式)。 int buffer_export(int v4lfd, enum v4l2_buf_type ...

v4l2 use dma buffer mode huifeidedabian的博客 968

user space struct v4l2_buffer buf; memset(&buf, 0, sizeof buf); buf.index = 0; buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE; buf.memory = V4L2_ME...

NVIDIA 平台中的V4L2架构分析qq_29042531的博客 65

nvidia 边缘计算设备v4l2 架构的实现

深入理解v4l2bufweixin_43360707的博客 451

举个例子理解一下，使用mmap方式获取磁盘上的文件信息，只需要将磁盘上的数据拷贝至那块共享内存中去，用户进程可以直接获取到信息，而相对于...

v4l2应用框架-摄像头v4l2编程(08)-申请缓存VIDIOC_REQBUFSweixin_42581177的博客 341

若是V4L2_MEMORY_DMABUF类型，则q->io_modes必须设置为VB2_DMABUF，mem_ops->attach_dmabuf，q->mem_ops->detach_dmabuf，q->mem...

v4l2 use V4L2_MEMORY_MMAPP方式，driver将buffer 转换为DMA 地址huifeidedabian的博客 511

建议使用以下方式进行vb2_queue 进行init struct vb2_queue *q; q->mem_ops = &vb2_dma_sg_memops; 这样 userspace reqbufs 就会使用 vb2_dma_s...

V4L2 零基础入门（一）——打开摄像头和获取摄像头基本信息weixin_48885322的博客 1888

Linux上用v4l2框架通过摄像头拍照的新手教程

v4l2框架-查询缓存信息（VIDIOC_QUERYBUF）weixin_43824344的博客 665

应用层查询缓存信息 调用vb2_ioctl_querybuf,平台是xilinx 重点分析__fill_v4l2_buffer __buffer_in_use call_mempop(vb, num_users, mem_priv)会调用vb2...

从应用调用vivi驱动分析v4l2 -- 申请缓存（VIDIOC_REQBUFS）转载和创作优秀的博客 438

vivi代码 v4l2测试代码 step 5：设置缓存 1，申请缓存 struct v4l2_requestbuffers req; req.count = nr_bufs; //缓存数量 req.type = V4L2_BUF_TYPE_VID...

Linux 视频设备驱动V4L2最常用的控制命令使用说明（1.02） 热门推荐阿牛——做专业的专家工程师 27+

Linux视频设备驱动常用控制命令使用说明 设置视频设备属性通过ioctl来进行设置，ioctl有三个参数，分别是fd, cmd,和parameter,表示设备描述符，控制命...

v4l2摄像头采集流程及应用程序转载和创作优秀的博客 3142

V4L2是linux上用于采集图片、视频、音频数据的一套框架，对上向应用程序提供统一的接口，对下支持各类复杂硬件的灵活扩展，在远程会议、视频监控...

“相关推荐”对你有帮助？

非常没帮助 没帮助 一般 有帮助 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务

中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心 中国互联网举报中心

啵啵啵电脑 关注

0 4 1

啵啵啵电脑 码龄8年 暂无认证

50 6万+ 2万+ 15万+ 原创 周排名 总排名 访问 等级

1149 100 79 27 539 积分 粉丝 获赞 评论 收藏

私信 关注

搜博主文章

热门文章

Linux /dev/sda1磁盘满了，清理办法。 44225

Chromatix 7 使用 <2> 12604

DDR、Flash、Ram、Rom的区别 7087

notepad++搜索结果窗口不见了 4864

Linux SD卡/SIO驱动开发 0 - 基本知识 3959

最新评论

AF base tuning <0>

weixin_42203563: 没太看懂最后这些“lensp os与DAC的关系”，这些推导最后就是为...

V4L2- Memory

hehui0921: int buffer_queue(int v4lfd, int i ndex, int dmafd) 这个是哪里调用的？

V4L2-框架

海底的雄鹰: 写的不错，赞一个，优秀

V4L2 驱动层分析

记不住Pipeline的杨同学: 能不能开放私信啊，想请教一些问题

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

ISO12233分辨率测试卡分类及功能说明

ISP-ASF

V4L2- Memory

2023年 1篇

2022年 20篇

2021年 4篇

2020年 12篇

2019年 15篇

2018年 3篇

2016年 1篇

2015年 3篇