V4L2

- Video for Linux -

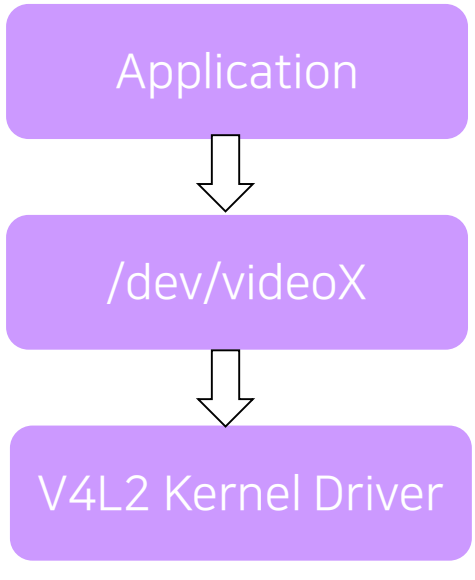
V4L2(Video For Linux Version 2)

What is V4L2?

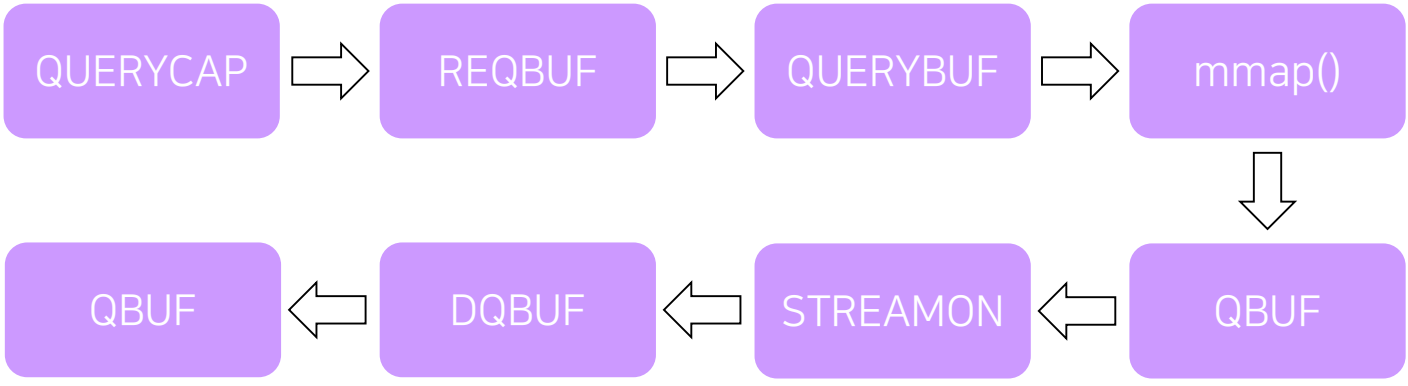
V4L2는 리눅스 시스템에서 카메라 입력을 받기 위한 표준 인터페이스이다. 최초 버전인 V4L 이후 1999년에 두번째 버전인 V4L2의 개발이 시작되었고, 초기 버전에서의 문제점들을 수정한 뒤 V4L2가 완성되었다.

V4l2 Driver

사용자 프로그램이 커널을 통해 시스템 하드웨어에 접근할 수 있도록 "dev" 디렉토리 내에 "video*"이라는 장치 파일이 생성된다. 사용자는 "/dev/video"를 통해 자료를 읽거나 기타 장치로 자료를 전송하는 것이 가능해진다. 일반적으로 Video Device는 Character Device로 생성된다. 이 Video Device는 일반 Character Device File과는 달리 read()나 write()로는 접근할 수 없으며 Ioctl()을 이용해야 접근이 가능하다. 이때 Device에 접근하기 편하도록 커널에서 제공하는 것이 V4L2이다.



V4L은 Application에서 Video Device에 접근하여 특정 명령을 요청하면 V4L2 Kernel Driver가 요청 받은 명령을 동작한 후에 다시 Application에게 수행 결과를 반환하는 식으로 진행된다. 여기서 중요한 것은 Application에서 보내는 명령의 순서이다.



Application에서는 ioctl()을 이용해 명령들을 반드시 순차적으로 전송해야 한다. 여기서 순서가 바뀐다거나, 필요한 명령어가 빠지는 등의 상황이 생기면 명령을 수행하는 중에 오류가 발생할 수 있으므로 주의해야 한다.
이런 명령어들 중에서 QBUF, DQBUF는 반복적으로 수행해야 영상을 얻을 수 있고, 한번만 수행한다면 프레임을 1개만 얻을 수 있다.

Device Open and Close

우리가 올바른 장치를 연결한다면, “video0”이 “/dev” 디렉토리에 나타날 것이다. 그럼 우리는 어떻게 장치를 열고 닫을 수 있을까?

open() 함수와 close() 함수를 사용해서 장치를 열고 닫을 수 있다.

```
int fd;
fd = open(“/dev/video0”, O_RDWR | O_NONBLOCK, 0);
if(fd == -1)
{
        perror(“Opening Video Device”);
        return 1;
}
close(fd);
```

여기서 O_NONBLOCK 옵션은 버퍼를 읽을 때 소프트웨어가 계속 차단되는 것을 방지한다.

VIDIOC_QUERYCAP

QUERYCAP(Query Capability)은 연결된 Device의 이름, 수행 가능한 동작 등 장치의 정보를 사용자 영역에 알려주는 역할을 한다. 여기서 우리는 기본적으로 capture가 가능한지 아닌지를 확인한다. V4L2는 일부 카메라를 지원하지 않기 때문에 이곳에서 오류를 발생시킬 것이다. 우리는 v4l2_capability 구조체와 VIDIOC_QUERYCAP을 사용할 것이다.

v4l2_capability 구조체의 선언이다.

```
struct v4l2_capability {
        __u8    driver[16];
        __u8    card[32];
        __u8    bus_info[32];
        __u32   version;
        __u32   capabilities;
        __u32   device_caps;
        __u32   reserved[3];
}
```

이런 것들을 통해 우리가 필요한 정보를 알아낼 수도 있다.

```
struct v4l2_capability caps = {0};
int fd;
fd = open("/dev/video0", O_RDWR | O_NONBLOCK, 0);
if(fd == -1)
{
        perror("Open Video Device");
        return 1;
}
if(-1 == ioctl(fd, VIDIOC_QUERYCAP, &caps))
{
        perror("Querying Capabilities");
        return 1;
}
printf("Driver Caps:₩n"
       "  Driver: %s₩n  Card: %s₩n  Bus: %s₩n  Version: %s₩n",
       caps.driver, caps.card, caps.bus_info, caps.version);
close(fd);
```

VIDIOC_S_FMT

V4L2는 웹캠이 지원하고 제공하는 이미지 형식과 colorspace를 쉽게 확인할 수 있는 인터페이스를 제공한다. v4l2_format 구조체는 이미지 포맷을 변경하는데 사용된다.

v4l2_format 구조체의 선언이다.

```
struct v4l2_format {
        __u32  type;
        union  {
            struct v4l2_pix_format          pix;
                                    .
                                    .
                                    .
}
```

다른 구조체들이 궁금하다면 videodev2.h를 참고하길 바란다.

v4l2_pix_format 구조체의 선언이다.

```
struct v4l2_pix_format {
        __u32  width;
        __u32  height;
        __u32  pixelformat;
        __u32  field;
        __u32  bytesperline;
}
```

```
struct v4l2_format fmt = {0};
fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
fmt.fmt.pix.width = 640;
fmt.fmt.pix.height = 480;
fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_MJPEG;
fmt.fmt.pix.field = V4L2_FIELD_NONE;

if(-1 == ioctl(fd, VIDIOC_S_FMT, &릇))
{
        perror("Setting Pixel Format");
        return 1;
}
```

이미지 폭과 높이를 각각 640과 480으로 설정했다. Pixelformat은 카메라가 지원하는 형식을 확인해야 한다. 현재는 이미지 형식을 MJPEG로 설정하였다.