<https://blog.csdn.net/fadefaraway/article/details/53954160>

ioctl用于向设备发控制和配置命令，有些命令也需要读写一些数据，但这些数据是**不能**用read/write读写的，称为Out-of-band数据。也就是说，read/write读写的数据是in-band数据，是I/O操作的主体，**而ioctl命令传送的是控制信息**，其中的数据是辅助的数据。例如，在串口线上收发数据通过read/write操作，而串口的波特率、校验位、停止位通过ioctl设置，A/D转换的结果通过read读取，而A/D转换的精度和工作频率通过ioctl设置。

#include <sys/ioctl.h>

int ioctl(int d, int request, ...);

　　d是某个设备的文件描述符。request是ioctl的命令，可变参数取决于request，通常是一个指向变量或结构体的指针。若出错则返回-1，若成功则返回其他值，返回值也是取决于request。   
　　以下程序使用TIOCGWINSZ命令获得终端设备的窗口大小。

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/ioctl.h>

int main(void)

{

struct winsize size;//该结构体是在sys/ioctl.h中被内核定义的

//isatty(fd);作用是检测fd中的文件描述符是否为tty终端文件,是则返回大于0的数

if (isatty(STDOUT\_FILENO) == 0)

exit(1);

//获取终端窗口的大小,将获取的值放到size结构体中

if(ioctl(STDOUT\_FILENO, TIOCGWINSZ, &size)<0) {

perror("ioctl TIOCGWINSZ error");

exit(1);

}

printf("%d rows, %d columns\n,%d", size.ws\_row, size.ws\_col,size);

return 0;

}

//在图形界面的终端里多次改变终端窗口的大小并运行该程序，观察结果。

ioctl是对应于不同的驱动的,没有固定的用法,这里举例获取终端的窗口大小。   
ioctl不同于fcntl,前者可以获取设备的物理信息,而后者之只获取访问控制属性。   
不同的设备文件的ioctl会有不同的操作,对比于下一个例子。

#include <stdio.h>

#include <unistd.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <stdlib.h>

#include <linux/fb.h>

int main(void)

{

int fd;

struct fb\_var\_screeninfo fb\_info;

fd = open("/dev/fb0",O\_RDONLY);

if(fd < 0)

{

perror("open /dev/fb0");

exit(1);

}

ioctl(fd, FBIOGET\_VSCREENINFO, &fb\_info);

printf("R = %d, C = %d\n",fb\_info.xres,fb\_info.yres);

close(fd);

return 0;

}