申请字符类设备号：

回顾：

1. 字符设备和杂项设备的区别（复习）：
   1. 杂项设备的主设备号是固定的，并且固定为10。

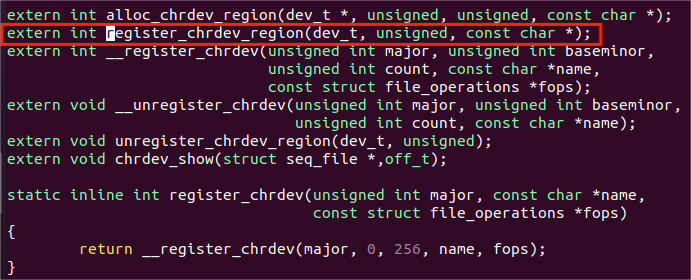
字符设备的主设备号不是固定的，需要我们 手动 或 系统自动 分配主设备号。

* 1. 杂项设备可以自动生成设备节点

字符设备需要我们手动生成设备节点。

1. 注册字符类设备号的两种方法：
   1. 静态分配一个设备号：

内核提供了相关的函数： 定义在：include/linux/fs.h 文件里面。



extern int register\_chrdev\_region(dev\_t, unsigned, const char \*);

使用这个函数就可以静态分配设备号了。

参数：

Dev\_t：设备号的起始值。（设备号是由主设备号+次设备号组成的，设备号类型是dev\_t类型。）

Unsigned:次设备号的个数。（我们要生成的连续设备编号的个数）

Char \*：设备的名称。（也就是 我们在串口终端对开发板执行：cat /proc/device 命令查看到的设备名称）

返回值：成功返回0

失败返回 非0

注意：

使用这个函数有一个前提条件：我们必须明确知道我们需要的设备号有没有被系统使用，如果系统已经使用了我们指定的设备号，我们就需要换一个指定的设备号。

/\*

重点解释一下 dev\_t 设备类型：

它定义在 include/linux/types.h 文件里面：

typedef u32 \_\_kernel\_dev\_t;

typedef \_\_kernel\_dev\_t dev\_t;

dev\_y 是用来保存我们的设备号的。

它是一个32位数。

其高12位是用来保存主设备号的。

其低20（有的是12）位是用来保存次设备号的。

\*/

Linux提供了几个宏定义来操作设备号：

其定义在：include/linux/kdev\_t.h 文件里面：

#define MINORBITS 20

次设备号的位数，一共20位

#define MINORMASK ((1U << MINORBITS) - 1)

次设备号的掩码

#define MAJOR(dev) ((unsigned int) ((dev) >> MINORBITS))

在dev\_t里面获取主设备号（右移20位）

#define MINOR(dev) ((unsigned int) ((dev) & MINORMASK))

在dev\_t里面获取次设备号

#define MKDEV(ma,mi) (((ma) << MINORBITS) | (mi))

由主设备号和次设备号得到设备号dev\_t，该宏定义的第一个参数是主设备号，第二个参数是次设备号

* 1. 动态分配：

其相关函数 定义在：include/linux/fs.h 文件里面：

extern int alloc\_chrdev\_region(dev\_t \*, unsigned, unsigned, const char \*);

参数：

dev\_t：用于保存生成的设备号。

unsigned：我们请求的第一个次设备号，通常是0。

unsigned：连续申请的设备号的个数。也就是说我们申请的这些设备号其主设备号相同，次设备号不同。

char \*：设备名称。

返回值：成功返回0

失败返回 非0

注意：

使用动态分配，其优先使用255~234。

1. 注销设备号：

定义在include/linux/fs.h文件：

extern void unregister\_chrdev\_region(dev\_t, unsigned);

参数：

dev\_t：分配设备号的起始地址（主设备号+值为0的次设备号组成的设备号）

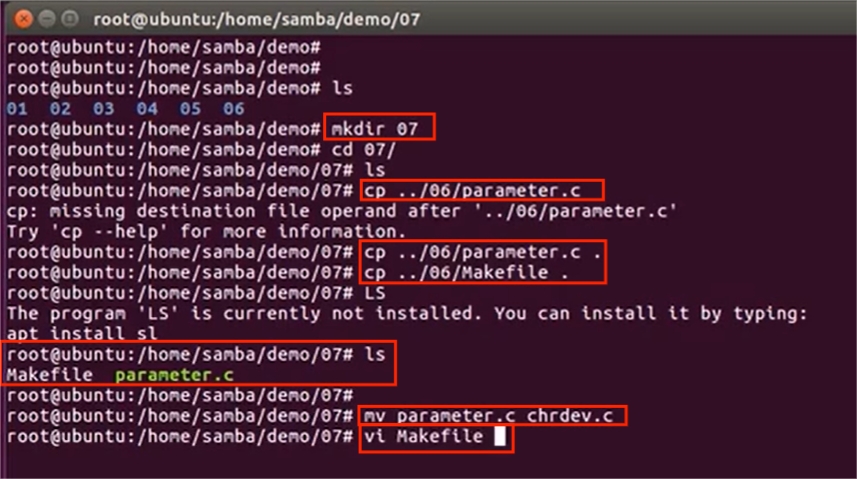
unsigned：申请的连续 次设备号的个数

建议使用：动态分配设备号

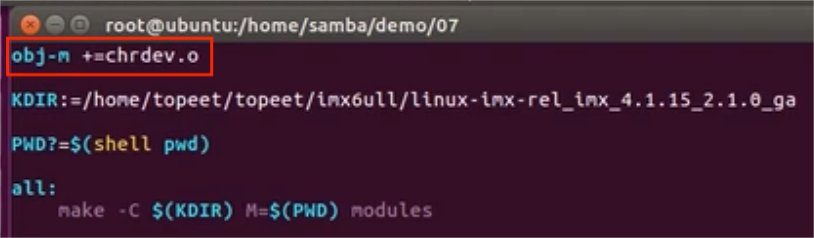
——————————————————————————————————

实操：

创建第七个实验：把上节课的内核传参代码拷贝进来。（我们涉及到内核传参，我们需要把主设备号传进去，驱动改名为chrdev.c）



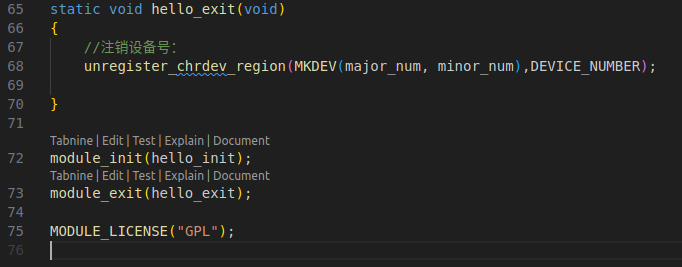
Makefile修改为：



chrdev.c:



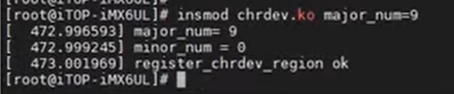




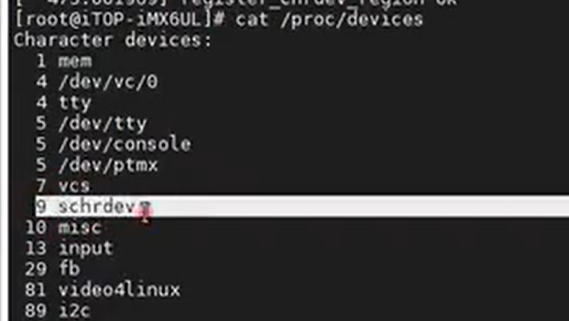
更改完驱动程序后，编译，拷贝到开发板：

静态分配：

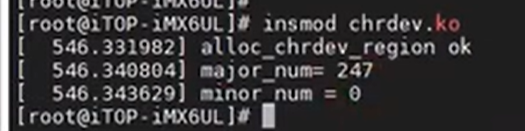
（静态分配前需查看:主设备9有没有被使用）



并且能够看到该schrdev设备



动态分配：卸载驱动后重新加载：（优先分配255~234）



执行cat /proc/devices 查看设备名称：

