# Block Device Drivers

You will turn the file lab.c into a skeleton block driver. The program consists of functions that primarily use printk() to display messages to the console.

A) Create macro SIMPLEBLOCKMAJOR set to 168 and the static ints nsectors set to 1024, sector\_size set to 512, and major set to SIMPLEBLOCKMAJOR;

Answer:

#define SIMPLEBLOCKMAJOR 168

static int nsectors = 1024;

static int sector\_size = 512;

static int major = SIMPLEBLOCKMAJOR;

B) Create a struct block\_device\_operations called simpleblock\_ops. Initialize the open, release, media\_changed, and revalidate\_disk methods to the corresponding functions already present in the file.

Answer:

struct block\_device\_operations simpleblock\_ops = {

.open = simpleblock\_open,

.release = simpleblock\_release,

.media\_changed = simpleblock\_check\_media\_change,

.revalidate\_disk = simpleblock\_revalidate

};

C) Have init\_module() register the block driver with register\_blkdev() and initialize the queue, with the handler function simpleblock\_request. Give it the name DEVICE\_NAME and the major number MAJOR\_NR. Have cleanup\_module() release the driver and kill the queue.

Answer:

int init\_module(void)

{

int retval =

register\_blkdev(MAJOR\_NR,DEVICE\_NAME);

/\* Fail to load if register\_blkdev fails \*/

if (retval < 0)

return retval;

. . .

dev ->queue = blk\_init\_queue(simpleblock\_request,&dev->lock);

}

void cleanup\_module(void)

{

int retval = unregister\_blkdev(MAJOR\_NR,DEVICE\_NAME);

blk\_cleanup\_queue(dev->queue);

}

D) Have my\_init\_module() initialize dev ->gd->major, dev->gd->fops, and dev->gd->queue.

Answer:

dev->gd->major = major;

dev->gd->fops = &simpleblock\_ops;

dev->gd->queue = dev->queue;

E) Compile and load the driver.

Answer:

# make all

# insmod ./lab.ko

F) Create a block device file /dev/simpleblock with major 168 and minor 0.

Answer:

# mknod /dev/simpleblock b 168 0

G) Load and compile the driver.

Answer:

# make all

# insmod ./answer.ko

H) Create an ext2 file system on the device.

Answer:

# mkfs -t ext2 /dev/simpleblock

I) Mount the device to /ramdisk.

Answer:

# mkdir /ramdisk

# mount -t ext2 /dev/simpleblock /ramdisk

J) Add some files to the device. Unmount and remount the device. Do you still see the files?

Answer:

# cp /etc/passwd /ramdisk/FILE1

# cp /etc/group /ramdisk/FILE2

# ls /ramdisk

lost+found

FILE1

FILE2

# umount /ramdisk

# ls /ramdisk

# mount -t ext2 /dev/simpleblock /ramdisk

# ls /ramdisk

lost+found

FILE1

FILE2

You can see the files as long as the device is mounted