Operating System Design & Implementation

Lab 10: Block-device drivers

TA: 周昆霖、吳崧銘

Contributor: 鄭加祥

Objective:

In this Lab, you can learn

* How to communicate with device drivers via ioctl() and write your own ioctl handlers.
* How block device drivers handle block requests
* Implement device snapshots using copy-on-write

Experiment:

* 10-1 Setting up a ramdisk
* 10-2 Adding new ioctl commands to the ramdisk driver
* 10-3 Implementing snapshot function in ramdisk driver

Experiment:

Exp. 10-1

* Setting up a ramdisk

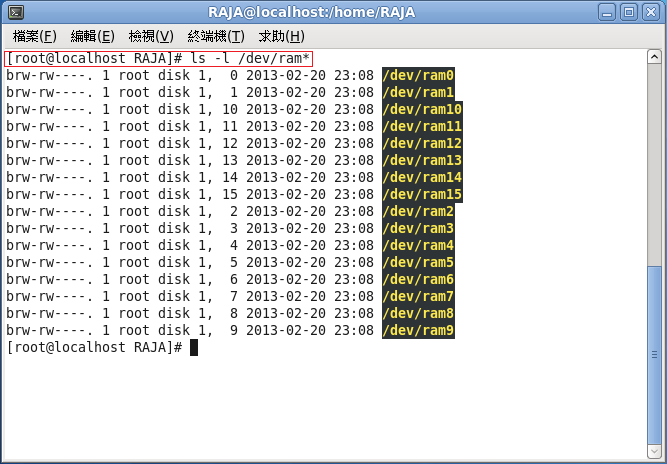
1. What is ramdisk?

→ A block device that uses memory pages as storage. Contents are volatile.

1. Testing basic functionality of a ramdisk.
2. List the ramdisk device nodes in /dev

Command: ls –l /dev/ram\*

Output:

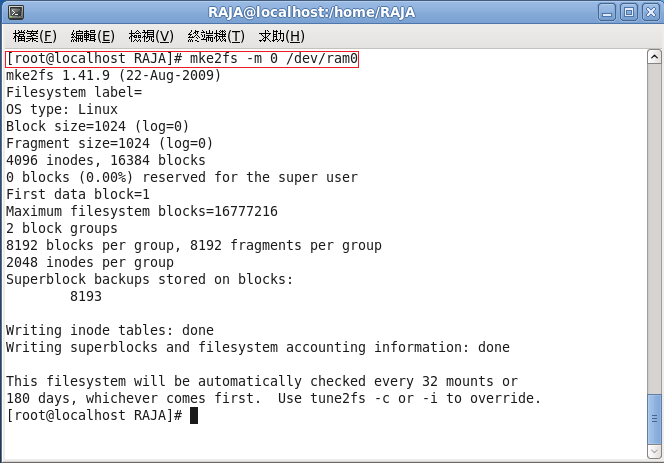


1. Select a ramdisk and format it to ext2

Note: In this lab, we use ram0.

Command: mke2fs –m 0 /dev/ram0

Output:

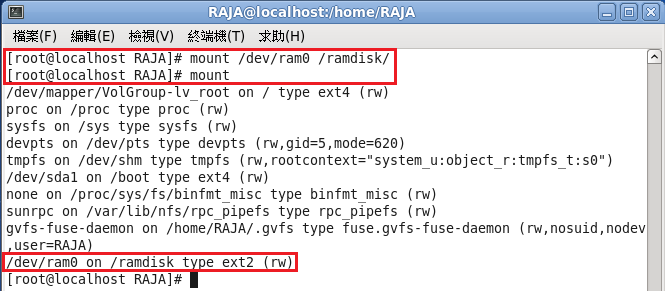


1. Mount a ramdisk

Command: mount /dev/ram0 /ramdisk

Note: Use "mount" to check the current mounted volumes.

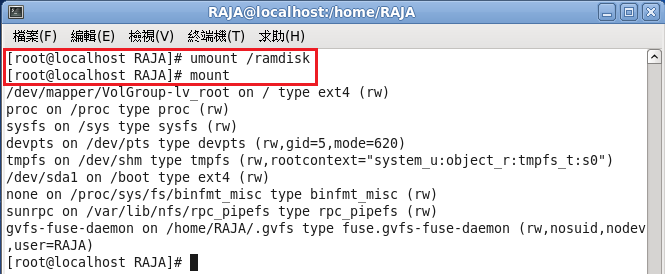
Output:



1. Unmount a ramdisk

Command: umount /ramdisk

Note: Again use the “mount” command to check the result

Output: 

Exp. 10-2

* Adding dummy ioctl handlers and writing user programs to call ioctl.

Part A : driver modification: ioctl() in “drivers/block/brd.c”

1. Refer to the following document for existing ioctl codes. <http://lxr.free-electrons.com/source/Documentation/ioctl/ioctl-number.txt>

2. Choose two unused ioctl number for your own use; one for snapshot

and the other for rollback.

3. Add dummy handlers for the two ioctl handlers (snapshot and rollback). Use printk() to indicate that your driver successfully received the events. For example, printk(KERN\_INFO "snapshot") and printk(KERN\_INFO "rollback");

4. You can modify brd\_ioctl() to handle the commands SNAPSHOT and ROLLBACK

5. Declare a flag as a global variable in brd.c. The flag indicates whether snapshot is enabled.

Part B : writing user programs

1. write two user programs SNAPSHOT.c and ROLLBACK.c
2. SNAPSHOT.c and ROLLBACK.c will include

→ include<unistd.h>, include<sys/ioctl.h >, include<fnctl.h>

1. You can use open() to open ramdisk, ram0 device!

e.g. open(“/dev/ram0”,O\_RDWR);

Note : In this LAB we use ram0.

1. You can use ioct() to call ramdisk ioctl !

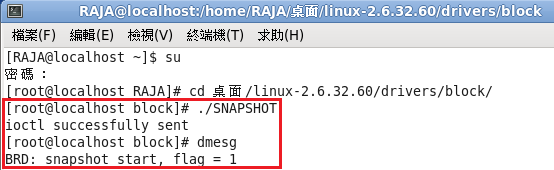
e.g. ioctl(fd,SNAPSHOT,0);

Part C : verify ioctl is received by the driver?

1. in ioctl() you can add some printk(“XXX is sucessful”) to print

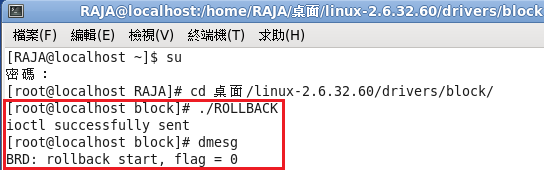
message

e.g. snapshot success



Enable Snapshot

e.g. rollback success



Enable Rollback

Exp. 10-3

* Implement snapshot function in ramdisk driver

1. What is the concept of radix tree?

→ The ramdisk driver uses a radix tree to manage all the cached pages while providing efficient lookup to the pages.

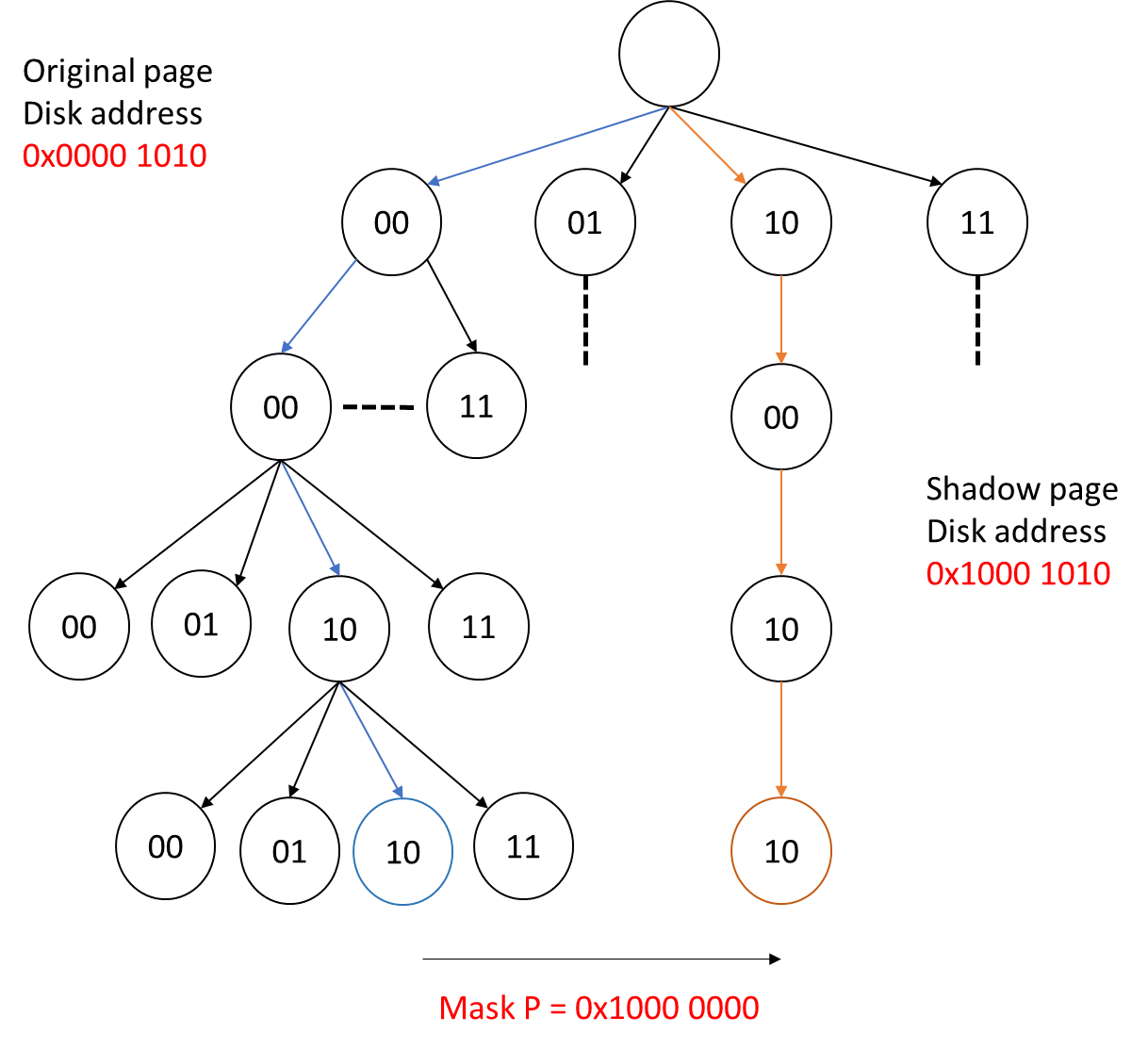
1. Put pages and their shadow pages in the same radix tree. Pages are indexed by their "disk offset".

Question: How to distinguish a page from its shadow page?

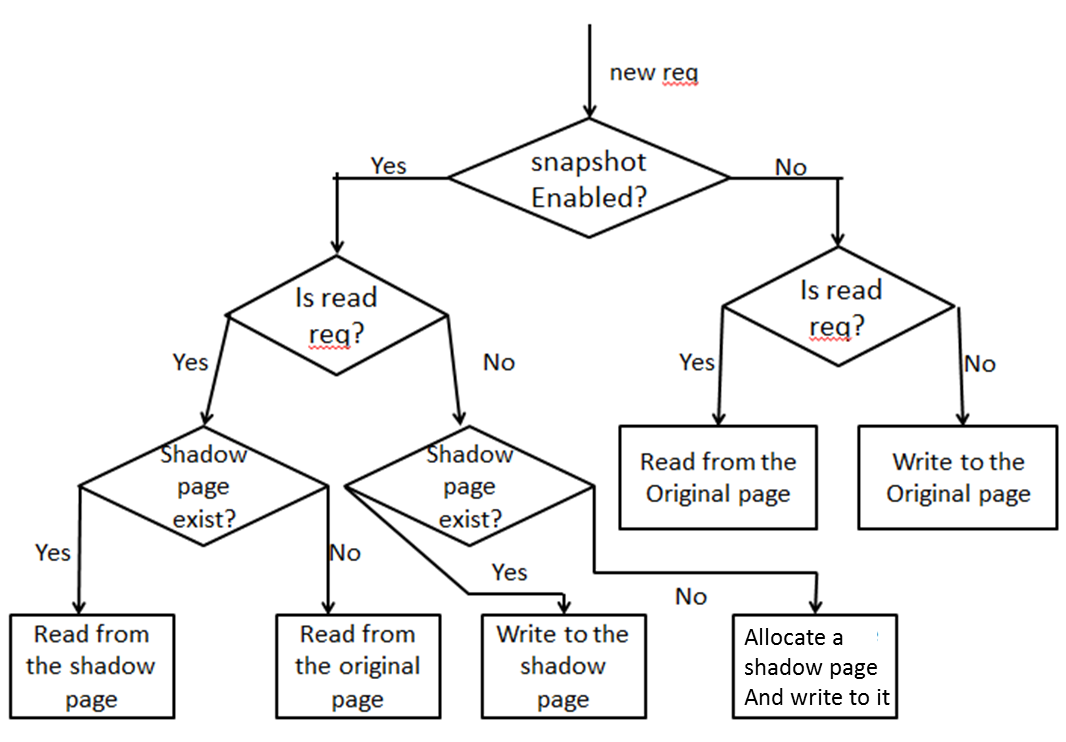
Suggestion: Give shadow pages a prefix. For example, for a page with a disk address

P, the "disk address" of its shadow page is 0x10000000 | P

e.g.



1. Control flow :



1. Relative API of radix tree :

|  |  |
| --- | --- |
| API format | static struct page \*brd\_lookup\_page(struct brd\_device \*brd,  sector\_t sector) |
| Function | Lookup and return the page indexed by the sector number |

|  |  |
| --- | --- |
| API format | static struct page \*brd\_insert\_page(struct brd\_device \*brd,  sector\_t sector) |
| function | look up and return the page indexed by the sector number. If the page does not exist, allocate it. |

|  |  |
| --- | --- |
| API format | page = alloc\_page(gfp\_flags); |
| function | allocate a free page (the recommended gfp\_flag is  GFP\_NOIO | \_\_GFP\_ZERO |

1. Hint: Modify brd\_lookup\_page() and brd\_insert\_page() to implement

snapshot and rollback.

You may have to implement a new function when rollback is called.

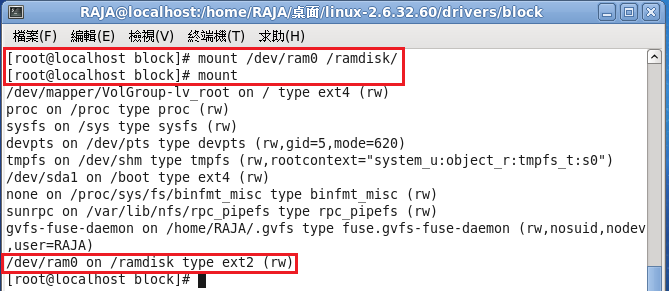
|  |  |
| --- | --- |
| API format | static void brd\_free\_page(struct brd\_device \*brd) |
| Function | Free all shadow pages when rollback |

Demonstration

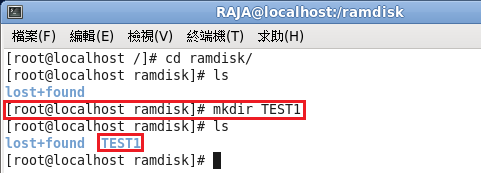
Step1：Create a directory named "ramdisk" under root.



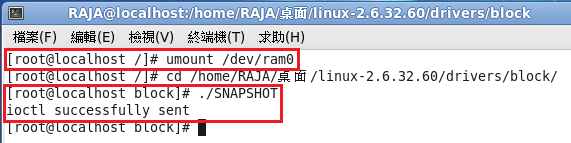
Step2：Format /dev/ram0 and mount it to the ramdisk directory



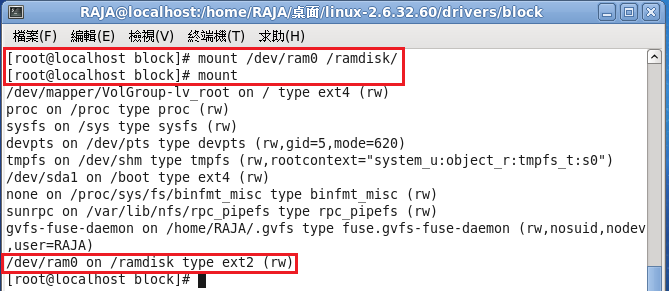
Step3：Create a new directory named "TEST1" in the ramdisk.



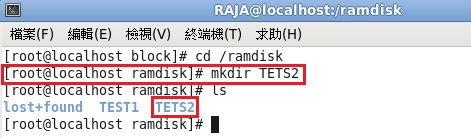
Step4：Unmount the ramdisk and create a snapshot.



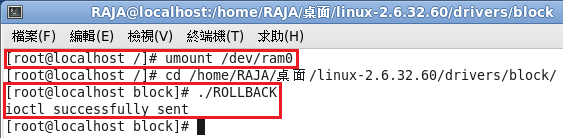
Step5：Mount the ramdisk again.



Step6：Create another directory named "TEST2".



Step7：Unmount the ramdisk and perform rollback.



Step8：Mount the ramdisk again, now the newly created "TEST2" directory should disappear, and only "TEST1" is left.



Note: The result of Step8 shall be the same as Step3