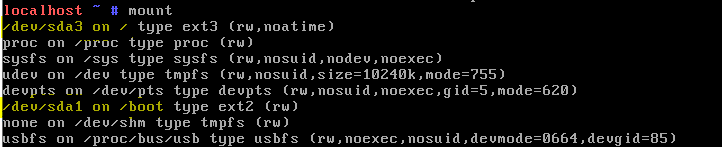
**오퍼레이팅시스템 11주차**

**12180626 성시열**

5. homework

1) Your Gentoo Linux has two disks: /dev/sda3 and /dev/sda1. Which one is the root file system? Where is the mounting point for the other one? Use "mount" command to answer this.

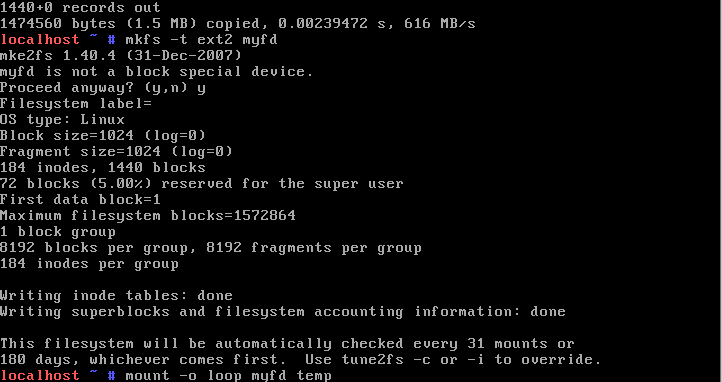


/dev/sda3는 /가 루트고

/dev/sda1는 /boot가 루트이므로

/dev/sda3가 루트 fs이다.

1-1) Redo 1) after mounting myfd to temp directory as you did in hw3 in lecture6-fs.docx.



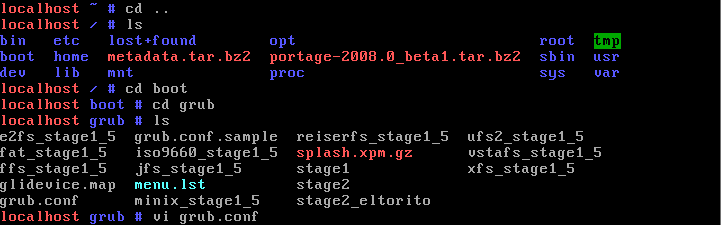
9주차 과제의 mount 과정을 진행하였다.

2) Add another entry in /boot/grub/grub.conf as below. This boot selection does not use initrd directive to prevent initramfs loading (initramfs is a temporary in-ram file system used for performance improvement).

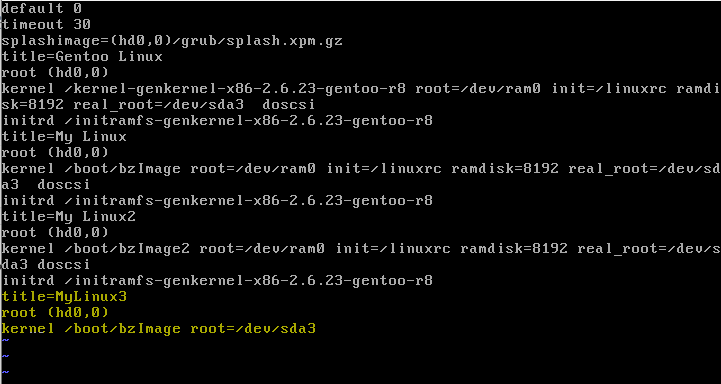
title=MyLinux3

root (hd0,0)

kernel /boot/bzImage root=/dev/sda3

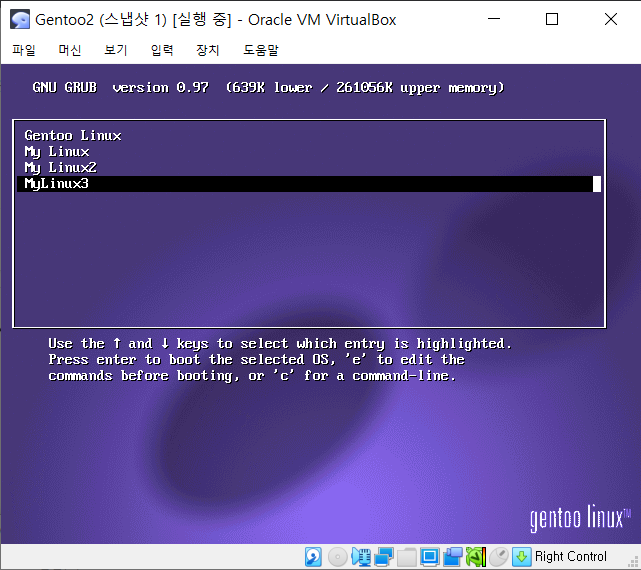


해당 위치로 이동하여 grub.conf로 이동한다



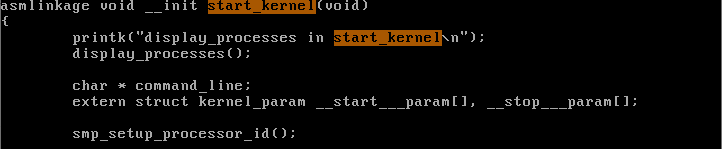
해당 문장을 추가하여 MyLinux3를 생성한다.

**From now on, use MyLinux3.**



Reboot 후 MyLinux3가 생성된 모습이다.

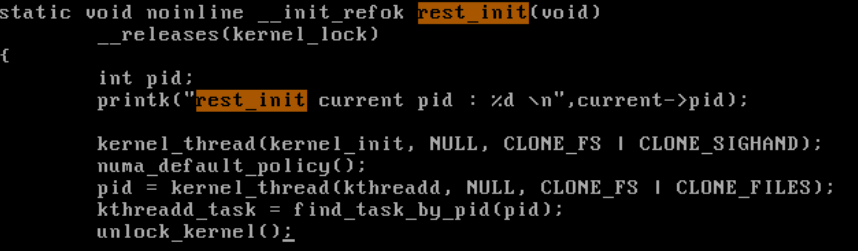
3) The kernel calls "mount\_root" to cache the root file system. Starting from "start\_kernel", find out the chain of intermediate functions that eventually calls "mount\_root". Confirm your prediction by printing out messge at each intermediate function of this chain until you reach mount\_root().



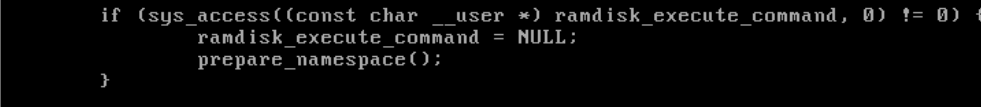
…



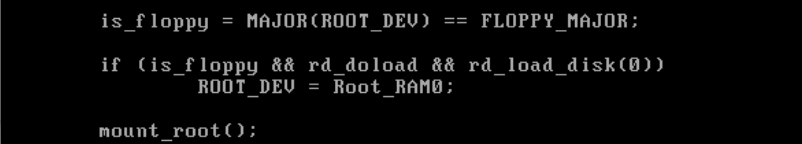
Linux-2.6.25.10/init/main.c의 Start\_kernel로 이동하여 확인해본 결과 rest\_init()을 마지막에 호출함을 알 수 있다.



rest\_init은 kernel\_init()을 호출한다.



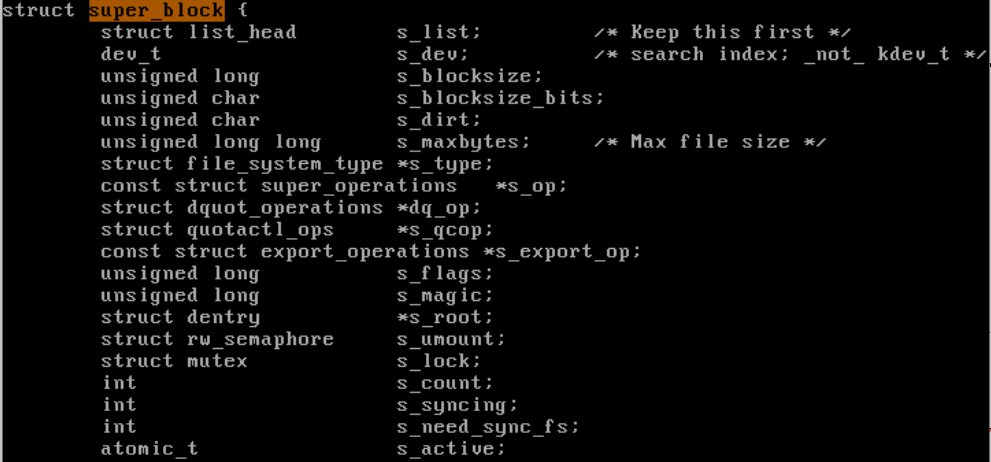
Kernel\_init()은 prepare\_namespace()를 호출한다.



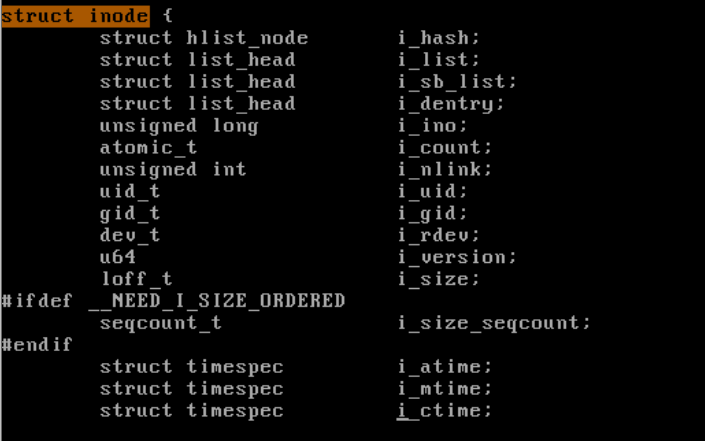
Linux-2.6.25.10/init/do\_mounts.c에서 해당 함수를 확인해보면 mount\_root()가 존재함을 알 수 있다.

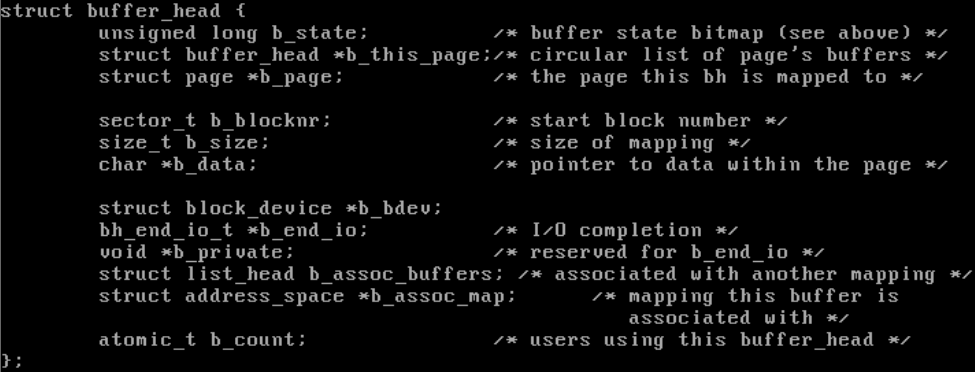
4) Find the data type for each added variable for super\_block, inode, buffer\_head, and dentry.

Include/linux/fs.h로 이동한다.



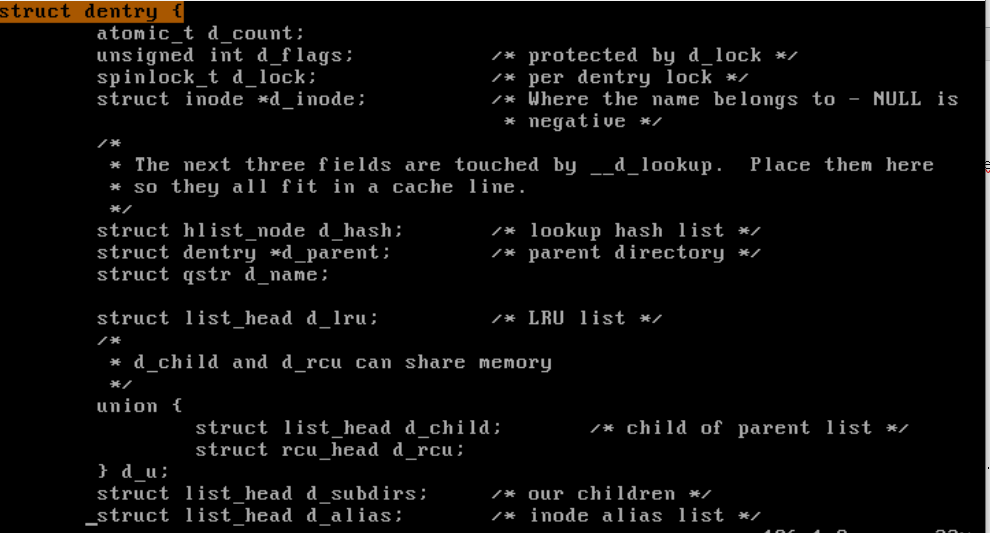
s\_list : next superblock  
s\_dev: device number. which disk this superblock came from?  
s\_type: file system type?  
s\_op : operations on superblock   
s\_root : root directory of the file system of this superblock  
s\_files : link list of file{} belonging to this file system  
s\_id : device name of this super block

  
i\_list : next inode  
i\_ino : inode number  
i\_rdev: device this inode belongs to  
i\_count: usage counter  
i\_op: operations on this inode  
i\_sb: pointer to super\_block{} this inode belongs to  
i\_pipe: used if a pipe



include/linux/buffer\_head.h로 이동하여 buffer head를 확인한다.

b\_blocknr : block number  
b\_bdev : device this block belongs to  
b\_size : block size  
b\_data : original block



include/linux/dcache.h로 이동하여 dentry를 확인한다.  
d\_inode: pointer to the corresponding inode  
d\_op : operations on this dentry  
d\_mounted: this inode is a mounting point if d\_mounted > 0  
d\_iname: corresponding file name

5) Change the kernel such that it displays all superblocks before it calls "mount\_root" and after "mount\_root". Boot with MyLinux3 to see what happens.

To display all superblocks, use below.

void display\_superblocks(){

struct super\_block \*sb;

list\_for\_each\_entry(sb, &super\_blocks, s\_list){

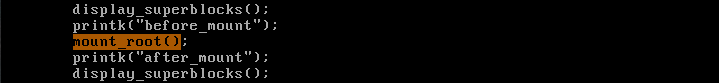
printk("dev name:%s dev maj num:%d dev minor num:%d root ino:%d\n",

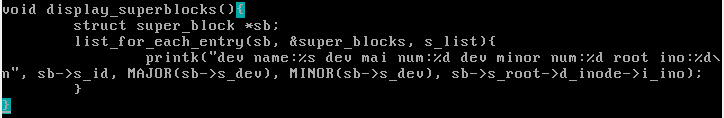
sb->s\_id, MAJOR(sb->s\_dev), MINOR(sb->s\_dev),

sb->s\_root->d\_inode->i\_ino);

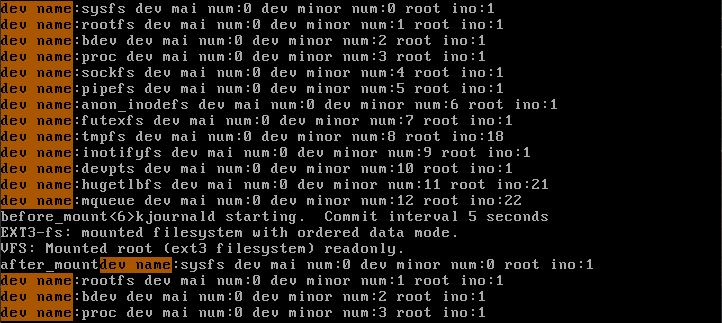
}

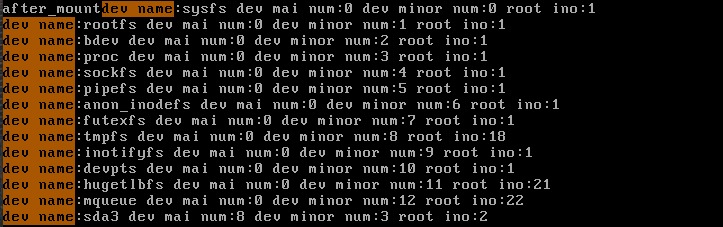
}





Linux-2.6.25.10/init/do\_mounts.c에서 다음과 같은 함수를 추가하고 mount\_root() 전후로 실행한다.





Recompile & reboot를 진행하고 부팅메세지를 출력해본 결과 해당 정보들이 출력됨을 알 수 있다. 전후의 차이점은 dev name : sda3 가 추가로 출력된 것을 알 수 있다.

6) Change the kernel such that it displays all cached inodes before it calls "mount\_root" and after "mount\_root". Boot with MyLinux3 to see what happens.

To display all cached indoes, use below.

extern struct list\_head inode\_in\_use;

void display\_all\_inodes(){

struct inode \*in;

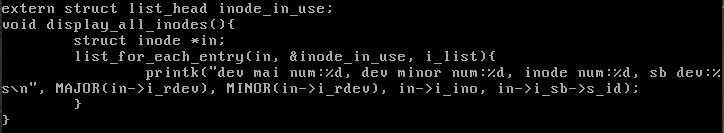
list\_for\_each\_entry(in, &inode\_in\_use, i\_list){

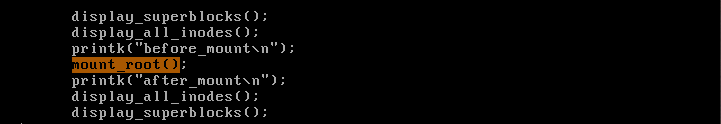
printk("dev maj num:%d dev minor num:%d inode num:%d sb dev:%s\n",

MAJOR(in->i\_rdev), MINOR(in->i\_rdev), in->i\_ino, in->i\_sb->s\_id);

}

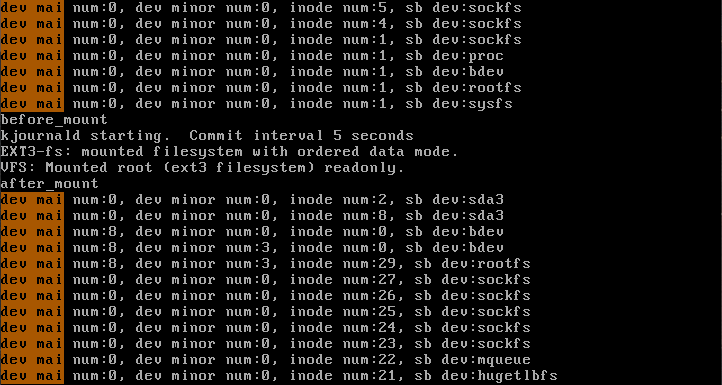
}





위와 같은 위치에 함수를 추가하고 mount\_root() 전후로 실행하였다.

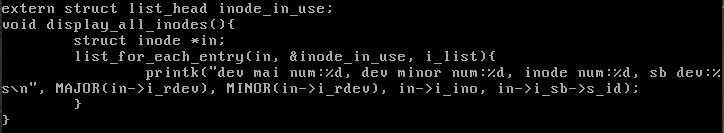






전후 차이로는 상단에 sda3에 대한 정보가 출력됨을 알 수 있다.

6-1) Modify display\_all\_inodes such that it can also diplay the file name and file byte size of each file represented by the inode.



해당 함수에서 각 파일의 파일 이름과 크기를 inode 별로 출력해준다.

6-2) Make a system call that displays file name and file byte size of all inodes in use. Show only the first 100 files. Look at the result with dmesg command.

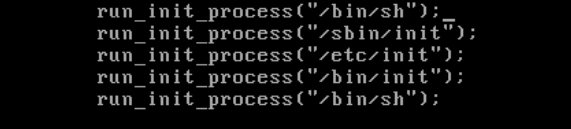
100개의 파일까지만 위 내용을 출력해준다.

6-3) Modify your system call in 6-2) so that it can display mounting points. Mount myfd to temp directory and confirm your system call can detect it.

7) The pid=1 process (kernel\_init) eventually execs to /sbin/init with

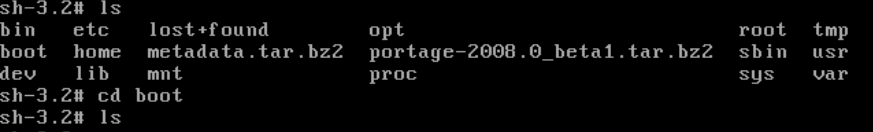
run\_init\_process("/sbin/init");

by calling kernel\_execve("/sbin/init", ....) in “init/main.c/init\_post()”. Change the kernel such that it execs to /bin/sh. Boot the kernel, and you will find you cannot access /boot/grub/grub.conf. Explain why.





위와 같이 login 하는 과정도 없이 바로 shell이 시작됨을 볼 수 있다.



Ls를 하고 boot에 들어가보니 아무 것도 없음을 볼 수 있다.

시스템을 시작하기 전에 초기화를 해주는데, 이는

Sda3를 mount 하는 과정이 포함되어 있다.

Etc/inittab 에는 window의 시작프로그램 리스트가 담겨있다.

/sbin/init 이 실행되면 child를 만들고 inittab 안에 있는 프로그램들을 execve를 한다.

Inittab 중 /sbin/agetty 이 실행되면login을 띄우는 프로그램(/bin/login) 을 execve하고 /bin/sh을 execve 하여 비로소 원래 자주 보던 쉘의 형태를 가진다.

보통은 이러한 과정으로 시스템이 시작되어진다.

하지만 init\_post()에서 /bin/sh를 먼저 시작해주면 이러한 초기화 과정이 일어나지 않아. Mount가 일어나지 않아서 /boot에 접근하여도 아무것도 없는 것이다.