

a) file extension in xv6

fs.c 의 bmap함수를 수정하여 구현했습니다. NDIRECT 값을 11로 바꾸고 NINDIRECT 값을 128, doubly-indirect block을 만들기 위한 DNINDIRECT 값을 128×128 로 주었습니다. 11번째 블록까지는 기존의 bmap과 동일하게 작동하다가 12번째 블록까지 오게 되면 NINDIRECT와 비교하여 작으면 12번째 블록과 연결된 페이지에 계속 매핑합니다. 마찬가지로 NINDIRECT까지 짝 차게 되면 13번째 블록으로 가게 되는데 balloc으로 새로운 블록을 할당하고 그 안에 또 balloc을 해서 할당하고 그 안부터 넣기 시작합니다. b) block group을 만들려고 하다가 보니 FSSIZE를 21113으로 설정하고 실행했을 때의 캡처는 찍지 못했습니다.

b) make block group

mkfs.c : bootblock, superblock, logblock 애들은 앞부분에 고정을 해놨습니다. 합쳐서 32개의 블록으로 구성되어있습니다. 일단 앞에 32개 블록은 고정인 거 같아서 FSSIZE에서 32를 빼고 그것을 32로 나눈 값이 한 그룹의 사이즈가 되도록 했습니다. 물론 4096보다 작다면 조건이 4096보다는 크거나 같아야하니까 그냥 디폴트 느낌으로 4096으로 설정을 해주었습니다. 그리고 그룹별 비트맵 블록 개수는 그룹사이즈를 4096으로 나눈 값으로 설정했습니다. 근데 여기서 나눈 몫으로 해버리면 나머지를 위한 비트맵 블록이 빠지기 때문에 1을 더해줍니다. 또한 아이노드 블록 개수도 그룹사이즈/32로 설정해줍니다. 설정이 끝나면 앞서 말한 메타데이터 블록을 건너뛴 지점부터 balloc을 해서 내용을 bzero를 통해 0으로 셋팅해줍니다. 그리고 이후에 mkfs.c의 메인 함수에서 iappend()로 디렉토리 관련해서 빈 datablock 들을 찾아 할당해주고 inode와 연결하는 것 같은데 이 때 기존에 있던 블록 넘버 시작점인 freeblock에서 시작하지 않고 건너 뛰어야 하는 메타데이터블록의 크기만큼 시작점에서 더해줍니다.

fs.c : iappend와 조금 비슷하게 balloc도 비트맵을 확인하고 사용해도 되는 datablock을 찾아 리턴해주기 위해 사용되는 것을 알 수 있는데, 기존의 탐색 방법과 다른 것은 여러 개의 블록 그룹별로 따로 봐야하는 것입니다. 블록그룹의 개수만큼 반복하면서 탐색하는데, 1바이트는 8비트여서 한 개의 블록에 4096개의 비트를 설정할 수 있습니다. 그래서 빈 블록을 찾아 해당 비트맵 비트를 1로 바꿔줄 때 1이라는 숫자를 찾은 비트맵 인덱스를 8로 나눈 횟수만큼 왼쪽으로 쉬프트해서 그 값을 비트맵으로 넣어줍니다.

"FSSIZE를 BGSIZE로 나누었을 때에 남는 block의 경우 기존 설명에는 log area에 추가하도록 되어 있었으나, log area에 추가하는 것이 아닌 log area 뒤 영역에 추가하시기 바랍니다. swap area로 활용될 공간이라고 생각하시면 되며, 실제 swapping을 구현하실 필요는 없습니다. super block 메시지 출력 시에 이렇게 생성된 swap area의 크기를 추가로 출력해주시면 되겠습니다." 라는 공지를 보긴 했는데 저는 swap area가 따로 존재하지는 않고 그냥 그룹 사이즈로 나누어 떨어지는 지점부터 디스크에 할당했습니다.

```
우분투 [실행 중] - Oracle VM VirtualBox
파일 편집 보기 입력
현재 활동 터미널 (일) 14:15
yhs@yhs-VirtualBox: ~/fs/xv6-sse-local

..QEMU: Terminated
yhs@yhs-VirtualBox:~/fs/xv6-sse-local$ make qemu-nox
gcc -Werror -Wall -o mkfs mkfs.c
./mkfs fs.img README_cat echo_forktest_grep_init_kill_in_ls_mkdir_rm_sh_stressfs_usertests_wc_zombie_swaptest_test
nmeta 32 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 44) blocks 179968 total 180000
argc: 20
balloc: first 745 blocks have been allocated
balloc: write bitmap block at sector 207
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie -c -o fs.o fs.c
fs.c: In function 'balloc':
fs.c:84:4: error: implicit declaration of function 'printf'; did you mean 'cprintf'? [-Werror=implicit-function-declaration]
    printf("ok\n");
    ^~~~~~
cc1: all warnings being treated as errors
<내장>: recipe for target 'fs.o' failed
make: *** [fs.o] Error 1
yhs@yhs-VirtualBox:~/fs/xv6-sse-local$ make qemu-nox
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie -c -o fs.o fs.c
ld -m elf_i386 -T kernel.ld -o kernel.entry.o bio.o console.o exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o spinlock.o string.o switch.o syscall.o sysfile.o sysproc.o trapasm.o trap.o uart.o vectors.o v
m.o -b binary initcode entryother
objdump -S kernel > kernel.asm
objdump -t kernel | sed '1,/SYMBOL TABLE/d; s/ / /; /$/d' > kernel.sym
dd if=/dev/zero of=xv6.img count=180000
100000+0 레코드 들어옴
100000+0 레코드 나감
5120000 bytes (51 MB, 49 MiB) copied, 1.04316 s, 49.1 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 레코드 들어옴
1+0 레코드 나감
512 bytes copied, 0.00157489 s, 325 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
352+1 레코드 들어옴
352+1 레코드 나감
180068 bytes (181 kB, 176 KiB) copied, 0.00403856 s, 44.7 MB/s
qemu-system-i386 -nographic -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -n 512
xv6...
cpu1: starting 1
0010460 2c
cpu0: starting 0
sb: size 180000 nblocks 179968 nnodes 5600 nlog 30 logstart 2 inodestart bmap start
init: starting sh
$ test
*****
```

```
우분투 [실행 중] - Oracle VM VirtualBox
파일 편집 보기 입력
현재 활동 터미널 (일) 14:20
yhs@yhs-VirtualBox: ~/fs/xv6-sse-local

..QEMU: Terminated
yhs@yhs-VirtualBox:~/fs/xv6-sse-local$ make qemu-nox
gcc -Werror -Wall -o mkfs mkfs.c
./mkfs fs.img README_cat echo_forktest_grep_init_kill_in_ls_mkdir_rm_sh_stressfs_usertests_wc_zombie_swaptest_test
nmeta 32 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 44) blocks 179968 total 180000
argc: 20
balloc: first 745 blocks have been allocated
balloc: write bitmap block at sector 207
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie -c -o fs.o fs.c
fs.c: In function 'balloc':
fs.c:84:4: error: implicit declaration of function 'printf'; did you mean 'cprintf'? [-Werror=implicit-function-declaration]
    printf("ok\n");
    ^~~~~~
cc1: all warnings being treated as errors
<내장>: recipe for target 'fs.o' failed
make: *** [fs.o] Error 1
yhs@yhs-VirtualBox:~/fs/xv6-sse-local$ make qemu-nox
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector -fno-pie -no-pie -c -o fs.o fs.c
ld -m elf_i386 -T kernel.ld -o kernel.entry.o bio.o console.o exec.o file.o fs.o ide.o ioapic.o kalloc.o kbd.o lapic.o log.o main.o mp.o picirq.o pipe.o proc.o sleeplock.o spinlock.o string.o switch.o syscall.o sysfile.o sysproc.o trapasm.o trap.o uart.o vectors.o v
m.o -b binary initcode entryother
objdump -S kernel > kernel.asm
objdump -t kernel | sed '1,/SYMBOL TABLE/d; s/ / /; /$/d' > kernel.sym
dd if=/dev/zero of=xv6.img count=180000
100000+0 레코드 들어옴
100000+0 레코드 나감
5120000 bytes (51 MB, 49 MiB) copied, 1.04316 s, 49.1 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 레코드 들어옴
1+0 레코드 나감
512 bytes copied, 0.00157489 s, 325 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
352+1 레코드 들어옴
352+1 레코드 나감
180068 bytes (181 kB, 176 KiB) copied, 0.00403856 s, 44.7 MB/s
qemu-system-i386 -nographic -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -n 512
xv6...
cpu1: starting 1
0010460 2c
cpu0: starting 0
sb: size 180000 nblocks 179968 nnodes 5600 nlog 30 logstart 2 inodestart bmap start
init: starting sh
$ test
*****
```

param.h에서 FSSIZE에 180000을 주고 실행했을 때 결과입니다. 5분 뒤에 점이 여러 개 찍혀있는 것을 볼 수 있었습니다.