Disk Layout for project 4 (cs6456)

(1) virtual disk

(1.1) 128 blocks

(1.2) 16 bytes / block

(1.3) range of block numbers : 0..127

(1.4) disk capacity: 128 x 16 = 2048 bytes

(1.5) disk name : 4 lowercase or uppercase letters

(2) directory

(2.1) single root directory (up to 8 entries, see below)

(3) files

(3.1) up to 8 files

(3.2) up to 64 blocks for data files (64 .. 127)

(3.3) other 64 blocks are either for metadata (0.. 42) or unused (43 .. 63)
(see below)

(3.4) maximum file size = 64 x 16 = 1024 bytes

(3.5) file name : 4 lowercase or upper case letters

(3.6) open file table (OFT): up to 4 opened files simultaneously (i.e. size of OFT=4)

(4) functions you need to design and implement

a. make_fs()	e. fs_open()	i. fs_write()
b. mount_fs()	f. fs_close()	j. fs_get_filesize()
c. dismount_fs()	g. fs_delete()	k. fs_lseek()
d. fs_create()	h. fs_read()	l. fs_truncate()

(5) functions provided to you

a. make-disk()

b. open-disk()

c. close-disk()

d. block-read()

e. block-write()

(6) How to design "make-fs()" function?

a. use "make-disk()" to initialize a new disk (e.g. store ∅ in each byte on the virtual disk)

b. use "open-disk()" to make the virtual disk available to the FS to be created

c. initialize superblock, directory, i-node map, and data map on disk (see below)

d. use "close-disk()" to close the disk (ie. make the virtual disk unavailable)

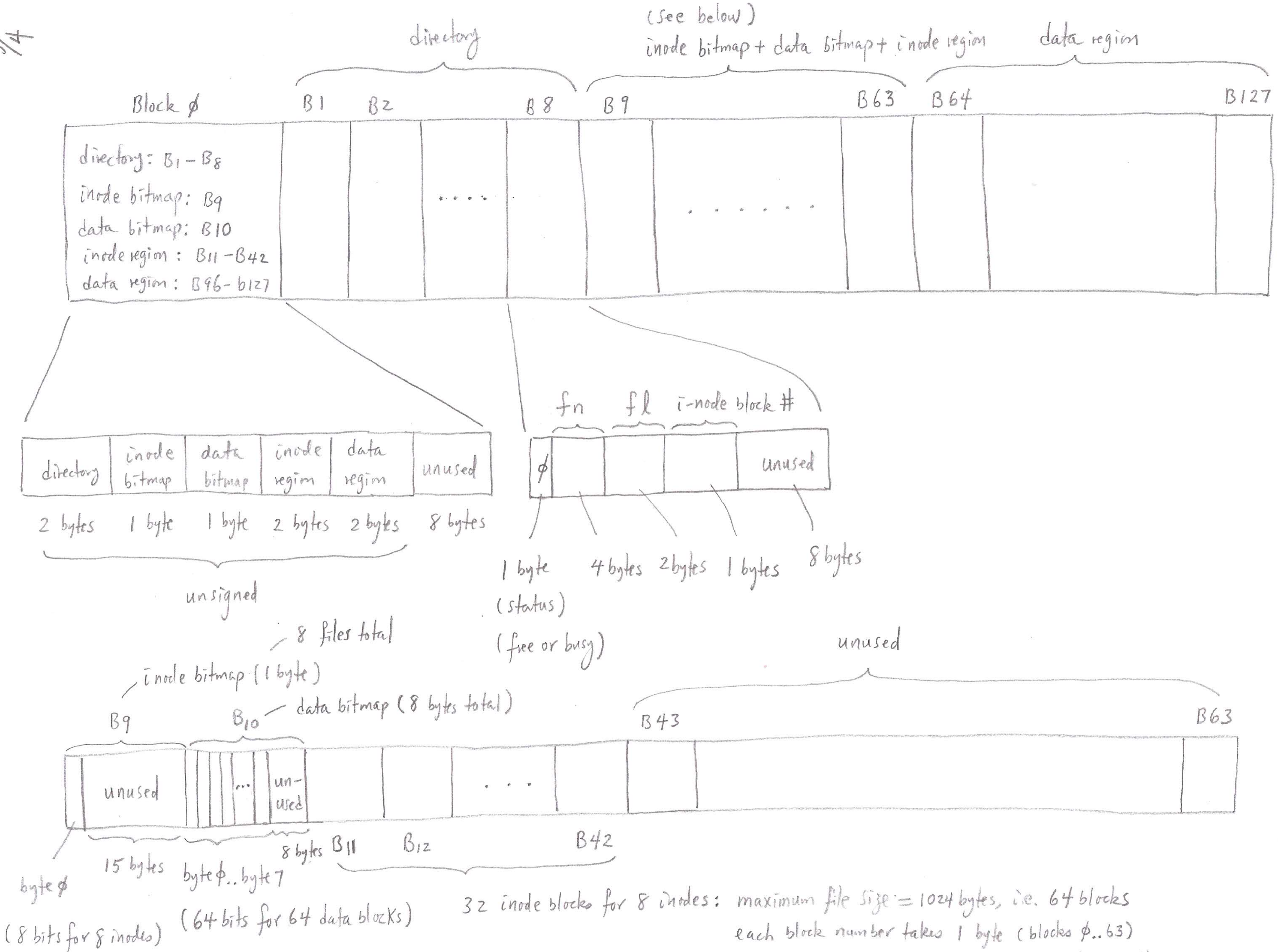
(7) How to design "mount-fs()" function?

a. use "open-disk()" function to make the virtual disk available to the FS to be mounted

b. load directory, i-node map, and data map into memory (use "block-read()" to do it)

c. create an OFT in memory

(8) disk layout (see next page)



32 inode blocks for 8 inodes: maximum file size = 1024 bytes, i.e. 64 blocks
each block number takes 1 byte (blocks ∅..63)
total blocks/inode = (1 × 64)/16 = 4 blocks (1 file)
8 files need 8 × 4 = 32 blocks

(9) implementation (for "mount_fs()")

(9.1) OFT (0..3)

	status	file offset	index in the directory
0	0	file ptr	
1	0		
2	0		
3	0		

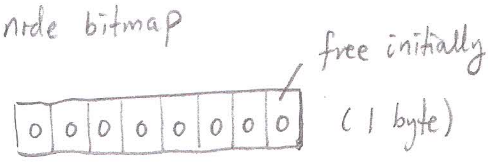
(9.2) directory (loaded into memory from the virtual disk)

each entry has 16 bytes (i.e. 1 block)

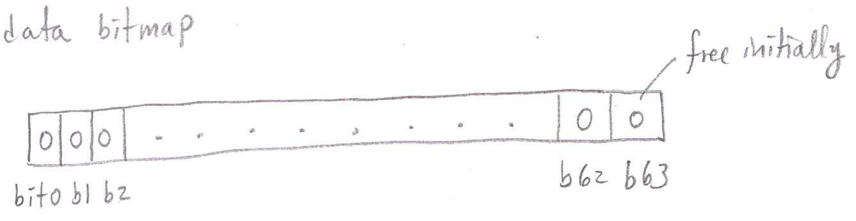
	status	fn	fl	inode number
0	0			
1	0			
2	0			
3	0			
4	0			
5	0			
6	0			
7	0			

(an array of struct)

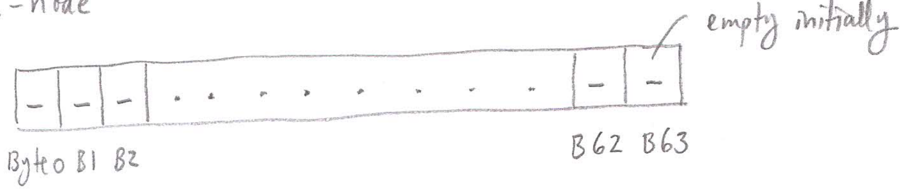
(9.3) i-node bitmap



(9.4) data bitmap



(9.5) i-node


empty initially
Byte0 B1 B2
B62 B63