



Computer Engineering II

Exercise Sheet 9

Basic

1 HDDs

- Which tracks are favored if SSTF is used for disk scheduling, assuming a random access workload?
- Explain how starvation can occur for SSTF, SCAN, C-SCAN, SPTF; how does F-SCAN solve the problem?
- If we always serve disk requests in the order they occur, we avoid starvation. Which disadvantages does this have?
- Assume you have two HDDs, one with 5ms average seek and 9000 rounds per minute, and one with 3ms average seek and 5400 rounds per minute. Both disks have a peak transfer rate of 120 MB/s. Which offers a better rate of I/O for a random access workload, assuming a 4KB sector size? How long is the total I/O time for a 200MB random access workload? Assume 1MB = 1000KB.

2 SSDs

Block	0				1			
Page	0	1	2	3	4	5	6	7
Content								
State	i	i	i	i	i	i	i	i

Perform the following writes and erases for an SSD, once for a direct mapped SSD, once for a log-structured one. Assume that a block consists of 4 pages, and each write is exactly the size of one page. For the log-structured version, use a page-level mapping table.

- write(0) content A
- write(6) content B
- write(4) content C
- write(1) content D
- write(0) content E
- garbage collect (only log-structured)
- write(4) content F

- garbage collect (only log-structured)

Below you'll find templates for the exercise. Show the SSD after each set of writes as well as after each block erasure.

Direct mapped SSD:

write(0) content A

Content									
State									

write(6) content B

Content									
State									

write(4) content C

Content									
State									

write(1) content D

Content									
State									

write(0) content E

Content									
State									

Content									
State									

write(4) content F

Content									
State									

Content									
State									

Log-structured SSD:

write(0) content A

Table									
Content									
State									

write(6) content B

Table									
Content									
State									

write(4) content C

Table									
Content									
State									

write(1) content D

Table									
Content									
State									

write(0) content E

Table									
Content									
State									

garbage collection

Table									
Content									
State									

Table									
Content									
State									

write(4) content F

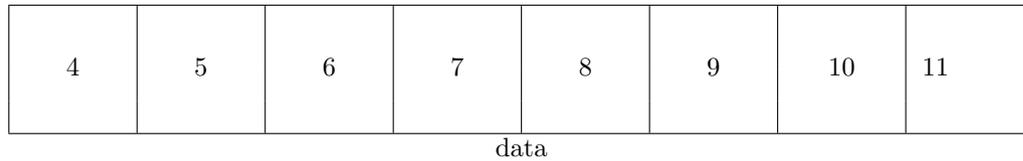
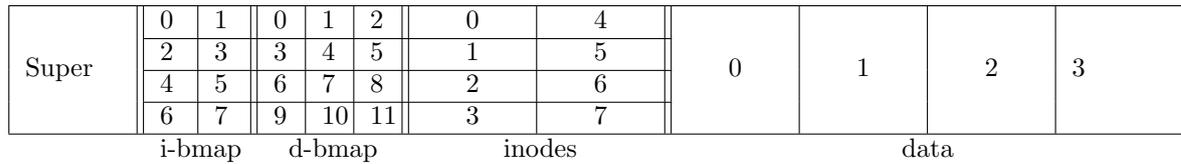
Table									
Content									
State									

garbage collection

Table									
Content									
State									

Table									
Content									
State									

3 File System



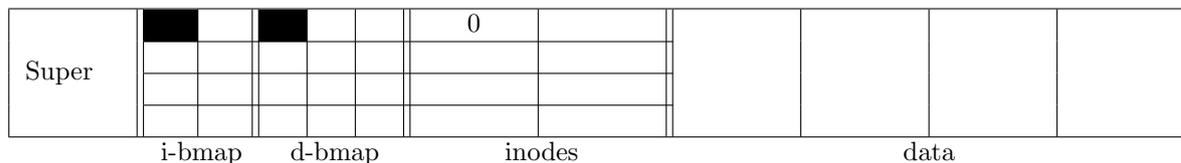
Assume we have a file system as depicted above with an inode table consisting of two blocks that can hold a total of 8 inodes, a data region that can hold 12 blocks, and uses bitmaps as allocation structures.

- a) Starting with only the root directory /, the following actions are performed:
- Create directory /a
 - Create directory /b
 - Create file /b/a.txt, containing "This is a long text file". Assume that this file needs two blocks.
 - Create file /a/b.txt, containing "This is another long text file". Assume that this file needs two blocks.
 - Delete file /b/a.txt
 - Create file /a/a.txt, containing "Hi". Assume that this file needs one block.

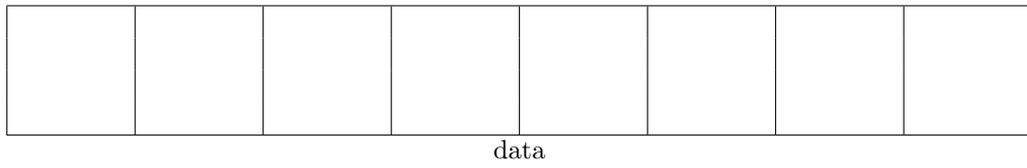
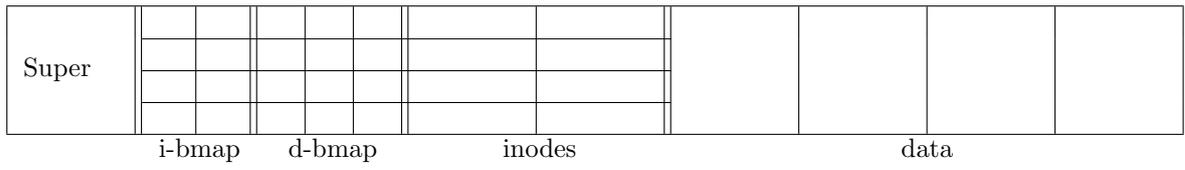
List the shell commands to perform these actions and show the state of the file system after each of these. For the inodes, you only need to indicate the indices of the data blocks and for the directory blocks the names and inode indices are sufficient. You find templates below.

- b) How does the file system resolve the address /a/b.txt?
 c) What is the difference between soft and hard links from an implementation view?

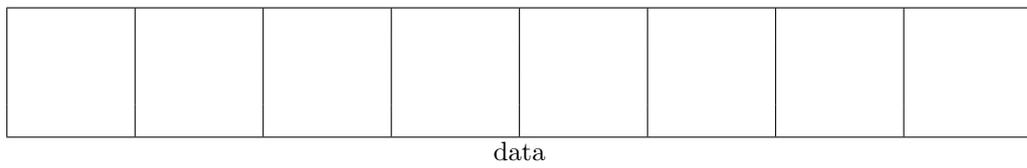
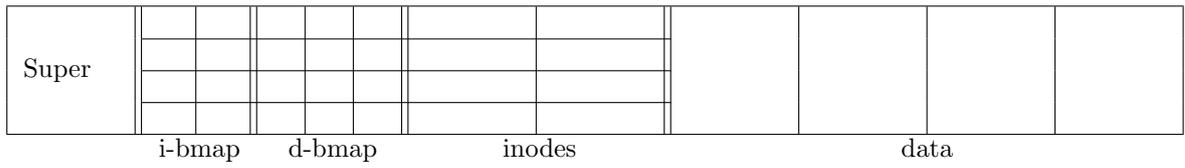
Starting state



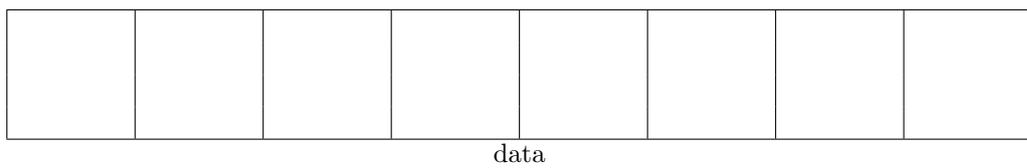
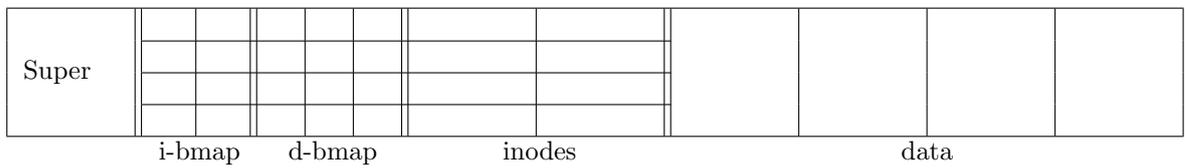
Command:



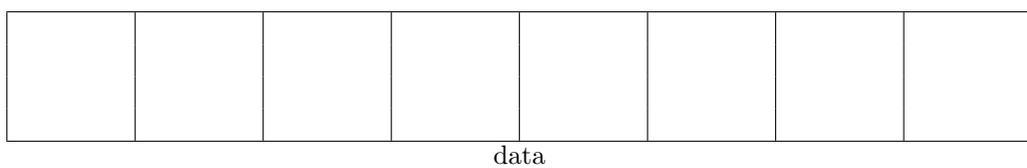
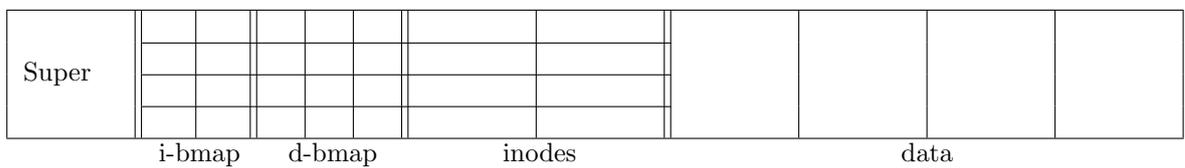
Command:



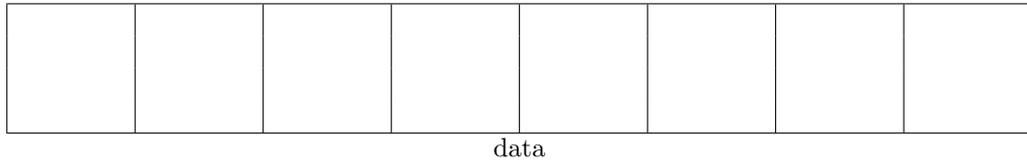
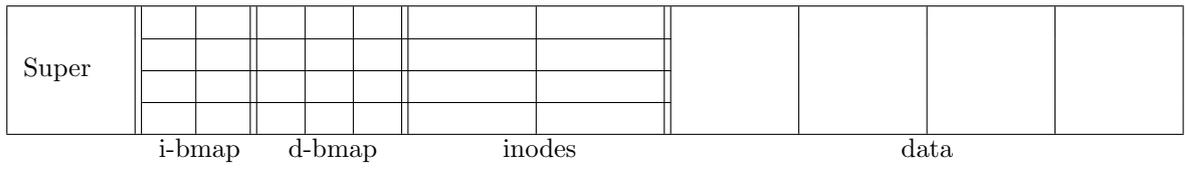
Command:



Command:



Command:



Command:

