Operating Systems	Homework 4
CS4410 Summer 2013	Due Sunday, 16 June

1 File-System Implementation

Suppose we have a system with an ordinary directory hierarchy. Assume that the data of each directory or file can be accommodated in one block. Each directory is implemented as a linear list. Suppose that a process A is working on the root directory (current directory) and it needs to open the file with absolute path *root/natalie/doc/myCV.pdf*. Assume that, currently, only the block of the root directory is loaded into the memory. Of course, the system-wide open-file table and all the necessary per-process open-file tables are already constructed in the OS's memory space.

(a) What is the structure of a directory entry?

(b) If the file *natalie/doc/myCV.pdf* has already been opened by another process, how many blocks should be transferred from the hard disk into the memory, in order A's open request to be successfully served? For each one of these blocks, please, explain why it is needed and how it is located (i.e where the information about its location is saved).

(c) Repeat sub-question (b) in case no process has opened this file before.

2 File-System Implementation

A UNIX file system has 1-KB blocks and 4-byte disk address. What is the maximum file size that the file system can accommodate, if each i-node contains 10 direct entries, and one single, one double and one triple indirect entry?

Can we save a file of 100MB in this file system? If no, which change would you propose for the file system design in order to accommodate this file? If yes, what is the maximum number N of disk accesses that would be needed in order to retrieve a data block of this file (the number of accesses should be relative to the directory that hosts that file)? What is an alternative design choice that we can make for the i-node, in order to decrease N, but still be able to accommodate this file?

3 Mass-Storage Structure

Suppose that a disk drive has 2,000 cylinders. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is:

86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

(a) Starting from the current head position, what is the sequence with which the requests will be satisfied for each one of the algorithms below (serviced sequence)?

(i) SSTF

(ii) LOOK

(iii) C-LOOK

Observing the final sequences, please, answer the questions bellow:

(b) Give a queue of pending requests that would generate the same serviced sequence for all the three algorithms.

(c) Give a queue of pending requests that would generate the same serviced sequence for the first two algorithms but exact reverse serviced sequence for the third algorithm.

(d) What is one conclusion that we can make?

4 Mass-Storage Structure

(a) Give an example of an environment or application that best matches the characteristics of each of the following RAID levels: 0,1,3,5 and 6.

(b) Are RAID 01 and RAID 1, essentially, the same configuration? Why?
Describe how RAID 03 would look like and whether it would differ from RAID 3.

5 Review

5.1 Hardware

Suppose that our system supports the system call *rename*, which takes as an argument the name of an existing file and the new name we want to give to that file. Describe briefly the operations that take place exactly after the execution of the system call, and until its completion. Do not forget to mention context switches, interrupts, directory search to find the file and the place where the file names are saved.

5.2 Synchronization

Please, fill the code gaps in the file study.py.