Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Spring Term 2014

Operating Systems and Networks
Assignment 8

Assigned on: **10th April 2014**
Due by: **17th April 2014**

1 File Systems

a) Is the "open" system call in UNIX absolutely essential? What would be the consequences of not having it?

b) Some OSes provide a system call "rename" to give a file a new name. Is there any difference at all between using this call to rename a file and just copying the file to a new file with the new name, followed by deleting the old one?

c) Name one advantage of hard links over symbolic links and one advantage of symbolic links over hard links.

d) It has been suggested that the first part of each file be kept in the same disk block as its inode. What good would this do?

e) An Operating System only supports a single directory, but allows that directory to have arbitrarily many files with arbitrarily long file names. Can something approximating a hierarchical file system be simulated? How?

f) Systems that support sequential files always have an operation to rewind files. Do systems that support random access files need this too?

g) Contiguous allocation of files leads to disk fragmentation. Is this internal fragmentation or external fragmentation?

h) One way to use contiguous allocation of disk and not suffer from holes is to compact the disk every time a file is removed. Since all files are contiguous, copying a file requires a seek and rotational delay to read the file, followed by the transfer at full speed. Writing the file back requires the same work. Assuming a seek time of 5 msec, a rotational delay of 4msec, a transfer rate of 8MB/sec and an average file size of 8KB, how long does it take to read a file into main memory then write it back to the disk at a new location? Using these numbers, how long would it take to compact half of a 16GB disk?

i) Consider the inode structure with 10 direct addresses and one indirect address. If the inode contains 10 direct addresses of 4 bytes each and all disk blocks are 1024 Bytes, what is the largest possible file? Consider 32-bit addressing.

j) Free disk space can be kept track of using free list or a bit map. Disk addresses require D bits. For a disk with B blocks, F of which are free, state the condition under which the free list uses less space than the bit map. For D having the value 16 bits, express your answer as a percentage of the disk space that must be free.

k) The beginning of the free space bitmap looks like this after the disk partition is first formatted: `1000 0000 0000 0000` (the first block is used by the root directory). The system always searches for free blocks starting at the lowest numbered block, so after writing a file A which uses 6 blocks, the bitmap looks like this: `1111 1110 0000 0000`. Show the bitmap after each of the following additional actions:

(a) File B is written, using 5 blocks.

(b) File A is deleted.

(c) File C is written, using 8 blocks.

(d) File B is deleted.