OBL4-OS

April 5, 2019

This is a mandatory assignment. Use resources from the course to answer the following questions. Take care to follow the numbering structure of the assignment in your submission. Some questions may require a little bit of web searching. Some questions require you to have access to a Linux machine, for example running natively or virtually on your own PC, or by connecting to gremlin.stud.iie.ntnu.no over SSH (Secure Shell). Working in groups is permitted, but submissions must be individual.

1 File systems

1. Name two factors that are important in the design of a file system.

**Being able to systematically store data. This involves finding space on the drive, which involves either scattering or linking together data on the drive.**

**Being able to systematically read data. This involves having sufficient metadata to be able to find back to where on the drive it was stored.**

1. Name some examples of file metadata.

* **Free map: list (bitmap) of disk cells that tells if the cells have a data block on it or not.**
* **Filename, filesize, date of creation, last date of file edited.**

2 Files and directories

1. Consider a Fast File System (FFS) like Linux’s ext4.

1. Explain the difference between a hard link and a soft link in this file system. What is the length of the content of a soft link file?

- **A hard link is a new element in the catalog file with the same inode number as the original. They share data blocks, and changes regarding access rights will affect both. It will not physically appear, but in the “ls” command result in the file, it will state an additional link.**

**- A soft link is a new file with a separate inode number and its own data blocks. It is marked in the “ls” command result with the letter “l” at the beginning of the line.**

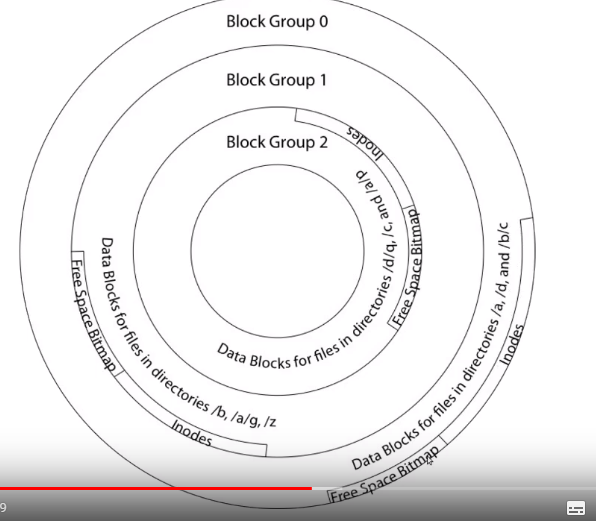
(b) What is the minimum number of references (hard links) for any given folder?

**2. One for the directory itself (.), and one for the parent directory (..).**

1. Consider a folder /tmp/myfolder containing 5 subfolders. How many references (hard links) does it have? Try it yourself on a Linux system and include the output. Use ls -ld /tmp/myfolder to view the reference count (hint, it’s the second column in the output).



1. Explain how spatial locality is acheived in a FFS.

**Blocks are grouped in sets of nearby cylinders. Files in the same directory are located in the same group. Subdirectories can be located in different block groups. The inode table is spread throughout disk, each inode is located close to the file blocks they relate to. Also, small files can be fragmented, but large files should be contiguous (close together) for rapid access.** ****

2. NTFS - Flexible tree with extents

1. Explain the differences and use of resident versus non-resident attributes in NTFS.

**Using resident attributes, we access the attributes directly, while in non-resident ones, because we don’t have a fixed block size, we reference to another mft record (with the start of an extent and the lengt of it) if the attribute has too much data.**

1. Discuss the benefits of NTFS-style extents in relation to blocks used by FAT or FFS.

**One benefit is being able to store the data using only two numbers: start of extent, and the length, instead of storing each and every block in the range. Also, extent allocation generally leads to less corruption.**

1. Explain how copy-on-write (COW) helps guard against data corruption.

**If the write operation fails e.g. due to power failure, the data will be corrupted, but if it was copied, it can easily be recovered.**

3 Security

1. Authentication

(a) Why is it important to hash passwords with a unique salt, even if the salt can be publicly known?

**When hashing a password, it only adds to the complexity of the password, and not the number of combinations. One can crack a password by guessing (brute force) simply by hashing each guess. With any salt added, there are many more combinations, making brute force much less useful as it will then take too long to crack.**

(b) Explain how a user can use a program to update the password database, while at the same time does not have read or write permissions to the password database file itself. What are the caveats of this?

**Maybe by getting access to an admin account somehow…?**

2. Software vulnerabilities

(a) Describe the problem with the well-known gets() library call. Name another library call that is safe to use that accomplishes the same thing.

“man gets” result states that the method is deprecated. Under bugs: “It is impossible to tell without knowing the data in advance how many characters get() will read, and because get will continue to store characters past the end of the buffer, it is extremely dangerous to use. Instead we can use fgets().”

1. Explain why a microkernel is statistically more secure than a monolithic kernel.

**Because of the fact that a microkernel is inherently more basic (basic scheduler, basic memory handling, basic I/O primitives), there is generally less that can go wrong. In turn, it is crash resistant.**