

Problem Sheet #12

Problem 12.1: file system permissions

(1+1+1 = 3 points)

Unix file system objects have basic permissions associated with (i) the file owner, (ii) the file's group members, and (iii) everybody else with access to the file system. Answer the following questions:

a) What is the meaning of the following file access permissions?

```
$ ls -l foo
-rwxrw-r-- 1 alice co-562 0 Nov 30 14:53 foo
$ ls -ld bar
drwx-wx--- 2 alice co-562 4096 Nov 30 14:56 bar
$ ls -l /usr/bin/sudo
-rwsr-xr-x 1 root root 182600 Feb 27 2021 /usr/bin/sudo
$ ls -ld /tmp
drwxrwxrwt 8 root root 700 Nov 17 13:02 /tmp
$ ls -ld /usr/local/bin
drwxrwsr-x 2 root staff 4096 Jul 30 2018 /usr/local/bin
```

b) Can bob, a member of the group co-562, read the content of the directory bar? Can bob create a file in the directory bar? Explain.

c) The regular user bob (with a umask of 0022) executes the following shell commands.

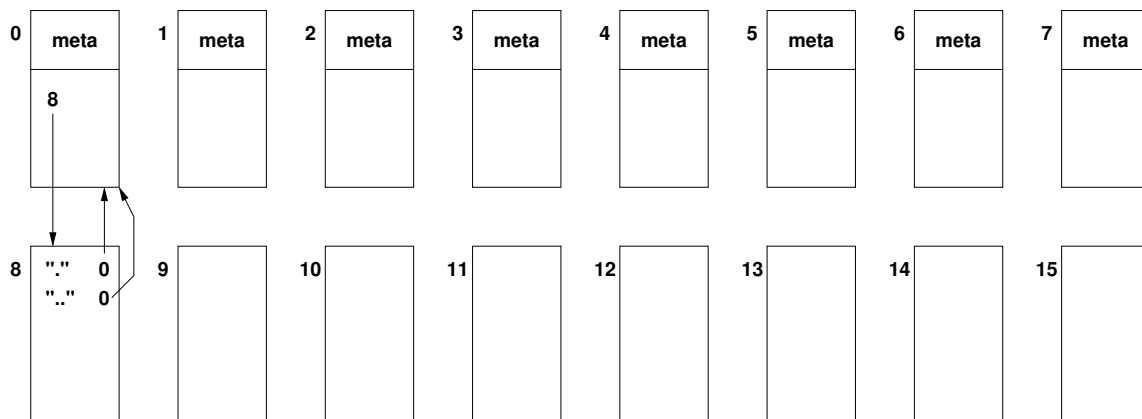
```
$ rm -f world
$ sudo echo hello > world
```

What are the file permissions of the new file and who is the owner of the file? Explain.

Problem 12.2: file system updates

(4 points)

Consider a small file system residing on a block device with 16 blocks (numbered 0..15). All blocks have the same size. The first 8 blocks (0..7) are reserved for inodes. An inode stores file attributes (metadata) of a file system object as well as references to data blocks. The remaining blocks (8..15) are reserved for storing data or directory information, they are called data blocks, or short dnodes. The root directory is always found in inode 0. Below is a figure visualizing the situation right after initializing the file system.



In the following, we focus on the block references that the file system maintains. The initial situation can be described as follows:

```

inode 0: 8 // inode 0 refers to dnode(s) 8
inode i: undef // inode i has undefined content (i in {1..7})
dnode 8: {(".", 0), ("..", 0)} // dnode 8 has 2 directory entries (to inode 0)
dnode i: undef // dnode i has undefined content (i in {9..15})

```

We are now making a number of changes to the file system. In the following steps, write down (using the notation shown above) which inodes and/or dnodes are updated. In each step, write down only the blocks that actually change. If you need to allocate so far unused inodes or dnodes, choose lower numbered unused inodes or dnodes first.

- a) `mkdir /a`
- b) `mkdir /b`
- c) `mkdir /a/c`
- d) `echo "hello" > /a/h`
- e) `mv /a/c /`
- f) `ln /a/h /b/h`
- g) `ln -s /a/h /c/m`
- h) `rm /a/h`

Problem 12.3: *redundant arrays of independent disks*

(1+1+1 = 3 points)

You are given n identical storage disks, each disk has a failure probability p . Lets assume disks fail independently from each other.

- a) What is the failure probability F_0 of a RAID 0 (striping) configuration of all n disks? Derive a general expression for $F_0(p, n)$.
 What is the failure probability F_1 of a RAID 1 (mirroring) configuration of all n disks? Derive a general expression for $F_1(p, n)$.
- b) A RAID 10 combines a RAID 1 with a RAID 0. Lets assume that n is an even number and that pairs of two disks are put into RAID 1 configurations and the resulting $\frac{n}{2}$ RAID 1 configurations are put into a RAID 0. What is the resulting failure probability $F_{10}(p, n)$?
 A RAID 01 combines a RAID 0 with a RAID 1. Lets assume that n is an even number and that $\frac{n}{2}$ disks are put into two RAID 0 configurations and the resulting two RAID 0 configurations are put into a RAID 1. What is the resulting failure probability $F_{01}(p, n)$?
- c) If you can choose between a RAID 10 and a RAID 01, which configuration do you prefer? Explain.