Operating Systems
Jacobs University Bremen
Dr. Jürgen Schönwälder

Module: CO-562
Date: 2021-11-18
Due: 2021-11-25

OS 2021 Problem Sheet #11

Problem 11.1: *open files and file updates and meta data changes* (1+1+1 = 3 points)

```
/*
 * catloop.c --
 */

#define _POSIX_C_SOURCE 201112L

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <fcntl.h>

int main(int argc, char *argv[])
{
    int fd;
    pid_t pid;

    if (argc != 2) {
        fprintf(stderr, "catloop: missing 'file' argument\n");
        return EXIT_FAILURE;
    }

    fd = open(argv[1], O_RDONLY);
    if (fd == -1) {
        perror("catloop: open");
        return EXIT_FAILURE;
    }

    while (1) {
        pid = fork();
        if (pid == -1) {
            perror("catloop: fork");
            (void) close(fd);
            return EXIT_FAILURE;
        }
        if (pid == 0) {
            char c;
            (void) lseek(fd, 0, SEEK_SET);
            while (read(fd, &c, 1) == 1) {
                write(STDOUT_FILENO, &c, 1);
            }
            (void) close(fd);
            return EXIT_SUCCESS;
        }
        (void) waitpid(pid, NULL, 0);
        sleep(1);
    }

    (void) close(fd);
    return EXIT_SUCCESS;
}
```

Save the source code shown above into the file `catloop.c`. Compile the C code to produce the executable `catloop` and afterwards execute the following shell commands on a Linux system (the behavior may be different on a Windows system):

```
$ rm -f foo
$ touch foo
$ ./catloop foo &
$ echo -n "hello " > foo
```

Answer the following questions, always with the same initial setup.

a) Describe what the program doing.

b) What happens if you append content to the file `foo` while `catloop` is running?

```
$ echo "world" >> foo
```

What happens if you truncate the file `foo` while `catloop` is running?

```
$ truncate -s 0 foo
```

Discuss the advantages and disadvantages of the behavior you have observed in the previous step. Could there be other file system update semantics?

c) What happens if you change the permissions of the file `foo` while `catloop` is running?

```
$ chmod 0 foo
$ ls -l foo
```

What happens if you remove the file `foo` while `catloop` is running?

```
$ rm -f foo
```

What are possible implications of this behavior?

Problem 11.2: *file system permissions* (1+1+1+1 = 4 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) Who has which access permissions for the file `foo`?

```
$ ls -l foo
-rwxrw-r-- 1 alice co-562 0 Nov 30 14:53 foo
```

Who has which access permissions for the directory `bar`?

```
$ ls -ld bar
drwx-wx--- 2 alice co-562 4096 Nov 30 14:56 bar
```

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) A regular user (with a umask of 0022) executes the following shell command. What are the file permissions of the file that is created and who is the owner of the file? Explain.

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

d) What is the meaning of the following access permissions?

```
$ 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/
```

Problem 11.3: *index node file system* (1+1+1 = 3 points)

You have designed a file system for a block storage device that supports blocks with a size of 128 bytes and blocks are numbered using a 32-bit numbers. Your file system uses index nodes (inodes) and direct, indirect, double indirect, and triple indirect indexes to data blocks. Your file system uses 64 bytes of the inode to store file attributes (file meta data).

a) What is the largest possible file system size? Explain.

b) The inode representing a file stores one indirect, one double indirect, and one triple indirect index, the remaining space is used to store direct indexes to data blocks. All additional index blocks use the entire block to store block numbers. What is the maximum file size? Explain.

c) An application opens a file and seeks to the position 123456. How many blocks (including the inode block) need to be read in order to find the relevant data block? Explain.