



## Exercise sheet 13 – Libraries, user management, and file systems

### Goals:

- Shared libraries
- User management
- File systems

### Exercise 13.1: Shared libraries

- (a) Update the `OS_exercises` repository with `git pull`.

**Proposal for solution:** `git pull`

- (b) Change into the  
`OS_exercises/sheet_13_libs_user_fs/library` directory.

**Proposal for solution:**

`cd sheet_13_libs_user_fs/library`

- (c) Compile the `supermath.c` into a shared library with the name `libsupermath.so`.

**Proposal for solution:**

```
1 #unversioned
2 gcc -fPIC -shared -o libsupermath.so supermath.c -lc
3
4 #versioned variant (just as an info)
5 gcc -fPIC -shared -Wl,-soname,libsupermath.so.1 -o libsupermath.so.1.0 supermath.c -lc
```

- (d) Copy the header to `/usr/local/include`. Check the permissions after copying.

**Proposal for solution:**

```
1 sudo cp supermath.h /usr/local/include/
```

- (e) Copy the shared library to `/usr/lib`. You may also update the shared library cache with `ldconfig`. Check the permissions after copying.

**Proposal for solution:**

```
1 sudo cp libsupermath.so /usr/lib/
2 sudo ldconfig
```

- (f) Change into the  
`OS_exercises/sheet_13_libs_user_fs/main_program` directory.



**Proposal for solution:**

```
cd sheet_13_libs_user_fs/main_program
```

- (g) Compile the math program by linking to the supermath.

**Proposal for solution:**

```
1 gcc -o math math.c -lsupermath #It is important that -l... is at the last position!
```

- (h) Use the math program with ./math 1 + 2.

**Proposal for solution:** ./math 1 + 2

### Exercise 13.2: User management

- (a) Which users exist on your system?

**Proposal for solution:** cat /etc/passwd to get all local users.

- (b) Which groups exist on your system?

**Proposal for solution:** cat /etc/group to get all local groups.

- (c) Create a new user test.

**Proposal for solution:** sudo adduser test

- (d) In which group is your new user?

**Proposal for solution:** The user test is part of the group test, which is was created at creating the user.

- (e) Create a new group dev\_data.

**Proposal for solution:** sudo addgroup dev\_data

- (f) Add the user test to the group dev\_data.

**Proposal for solution:** sudo adduser test dev\_data

- (g) Inspect the passwd and group files again, on your system.

**Proposal for solution:**

```
1 cat /etc/passwd
2 cat /etc/group
```

### Exercise 13.3: File systems questions

- (a) Visualise the directory inodes of /home/dev, similar to the slide 22 on „OS 15 – File systems“

**Proposal for solution:** Cmp. slide 22 on „OS 15 – File systems“

- (b) Consider a file that is moved with:

```
mv ./file subdir/
```

Why is it not required to copy the file content into the subdir and remove the content from the current directory?

**Proposal for solution:** Because the directory to file dependency is only management by referencing the inode nr. and the file name inside the directory inode. When a file is moved into another directory, the inode nr. and the file name is removed from the source directory inode and added to the destination inode.

(c) What is faster? Explain your answer.

- a) `mv ~/file ~/subdir/`
- b) `mv ~/file /mnt/USB_STICK`

(d) What is a journaling file system? You may do some research to answer that.

**Proposal for solution:**

*A journaling file system is a file system that keeps track of changes not yet committed to the file system's main part by recording the intentions of such changes in a data structure known as a "journal", which is usually a circular log. In the event of a system crash or power failure, such file systems can be brought back online more quickly with a lower likelihood of becoming corrupted.*  
[https://en.wikipedia.org/wiki/Journaling\\_file\\_system](https://en.wikipedia.org/wiki/Journaling_file_system)

(e) Are EXT4, BTRFS, FAT32, or NTFS journaling file systems? You may do some research to answer that.

**Proposal for solution:** EXT4, BTRFS, and NTFS are journaling file systems, FAT32 not (see: [https://en.wikipedia.org/wiki/Comparison\\_of\\_file\\_systems#Features](https://en.wikipedia.org/wiki/Comparison_of_file_systems#Features)).

(f) Do your own research for the following file system properties

- Case sensitive
- Hard links
- Online grow
- Snapshotting
- Max file size

and check which of the file systems EXT4, BTRFS, FAT32, or NTFS supports these properties.

**Proposal for solution:**

See: [https://en.wikipedia.org/wiki/Comparison\\_of\\_file\\_systems#Features](https://en.wikipedia.org/wiki/Comparison_of_file_systems#Features)

### Exercise 13.4: File system handling

(a) Switch your Linux VM off.

(b) Add a new hard disk to your VM:

Settings -> Storage -> Adds hard disk ->

Create new disk -> VMDK -> Dynamically allocated -> 1 GB

(c) Start your Linux VM.

(d) Use `gparted`: to create a GPT partition table on your new hard disk.

**Proposal for solution:**

`gparted`: on `/dev/sdb`

Device -> Create Partition Table... -> `gpt`



- (e) Use `gparted`: to create a partition with the `ext4` file system.

**Proposal for solution:**

`gparted`: on `/dev/sdb`

- New
- Create as: Primary Partition
- Partition name: data
- File system: ext4

- (f) Mount your newly created file system to `/mnt`.

**Proposal for solution:**

```
1 sudo mount /dev/sdb1 /mnt
```

- (g) Create a new file on your mounted file system.

**Proposal for solution:**

```
1 sudo touch /mnt/file
```

- (h) Unmount the mounted file system.

**Proposal for solution:**

```
1 sudo umount /mnt
```

- (i) List the content of `/mnt`

**Proposal for solution:**

```
1 sudo ls -l /mnt
```

The file is gone, because the file system on `/dev/sdb1` is unmounted.