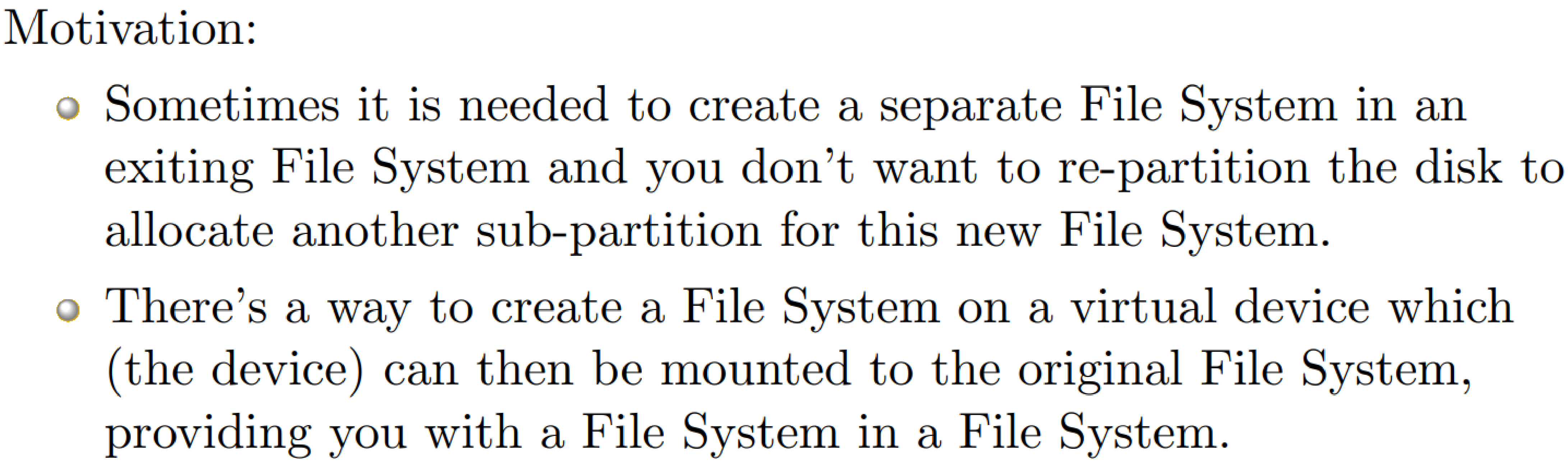
Exercise 1 (1/5)



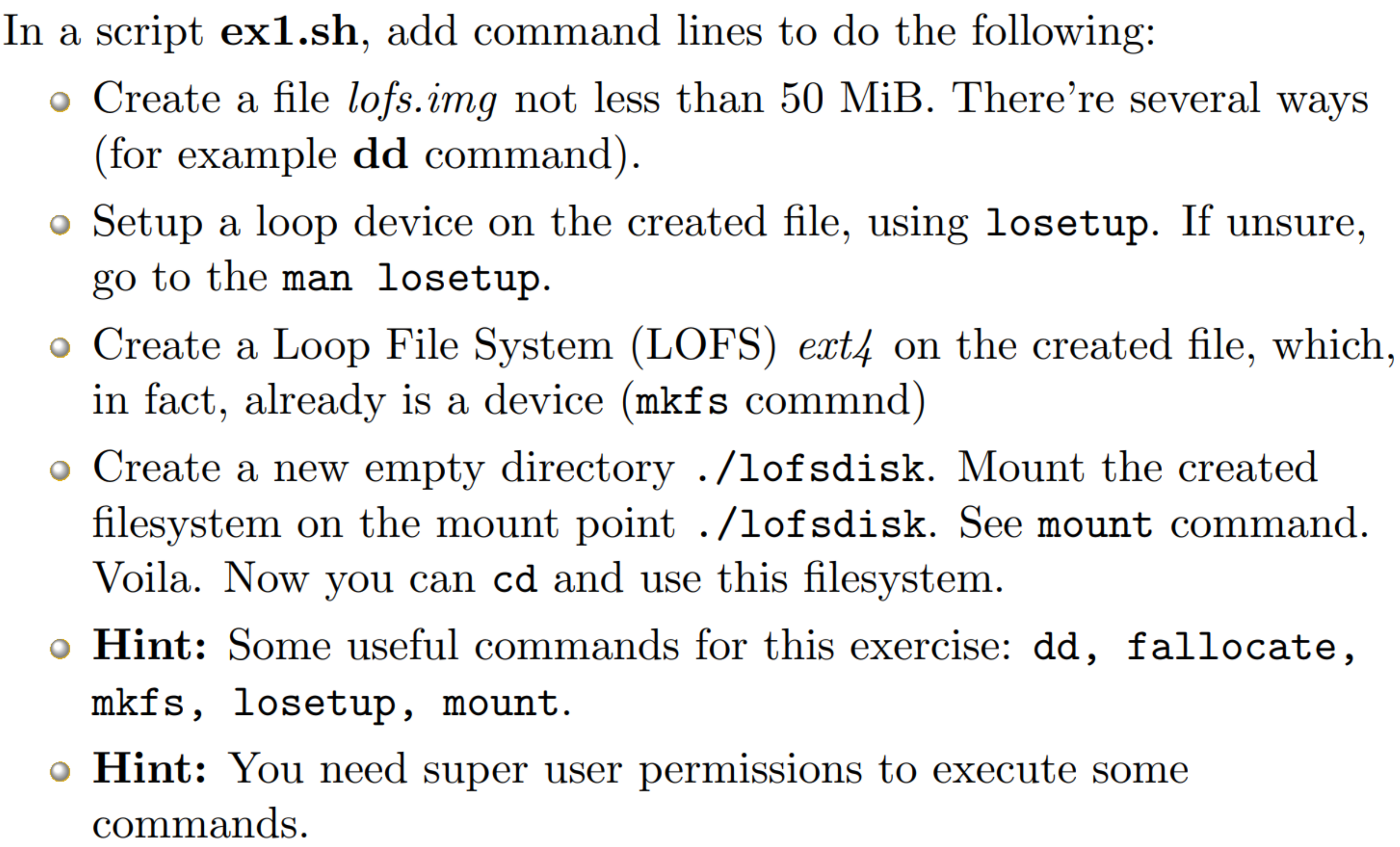
Motivation:

- Sometimes it is needed to create a separate File System in an exiting File System and you don’t want to re-partition the disk to allocate another sub-partition for this new File System.
- There’s a way to create a File System on a virtual device which (the device) can then be mounted to the original File System, providing you with a File System in a File System.

Exercise 1 (2/5)

- What if you want to run a process, that you don't trust. That you don't want to see your files and data? What if you could isolate it in such a way that the process will see only what you want it to see?
- There are multiple ways to achieve it. One of which is `NameSpaces`. But today we will use another older way - `chroot`. This command allows you to change what is the *root* dir for the process.
- For example, you could create a separate file system, mount it onto a virtual device (loop) on a file, create a process and chroot it on this file. Everything that the process will then create can be transfered as a single file. Or removed. Or.. whatever.

Exercise 1 (3/5)



In a script **ex1.sh**, add command lines to do the following:

- Create a file *lofs.img* not less than 50 MiB. There're several ways (for example **dd** command).
- Setup a loop device on the created file, using `losetup`. If unsure, go to the `man losetup`.
- Create a Loop File System (LOFS) *ext4* on the created file, which, in fact, already is a device (`mkfs` commnd)
- Create a new empty directory `./lofsdisk`. Mount the created filesystem on the mount point `./lofsdisk`. See `mount` command. Voila. Now you can `cd` and use this filesystem.
- **Hint:** Some useful commands for this exercise: `dd`, `fallocate`, `mkfs`, `losetup`, `mount`.
- **Hint:** You need super user permissions to execute some commands.

Exercise 1 (4/5)

- Add two files **file1**, **file2** to the **LOFS** where **file1** contains your first name, and **file2** contains your last name.
- `[Do not need to add this step to the script]` Write a simple C program `ex1.c` which will list the contents of the `root` directory (/) (use **opendir** and **readdir**).
- Define a function get_libs() in the script to return the paths of all shared libraries of a binary file whose path is passed as the first argument `$1` to the function.
- Use the previous function to get shared libraries of commands `bash`, `cat`, `echo`, `ls` and add them with their shared libraries to the LOFS.
- Change the root directory of the process to the mount point of the created LOFS and run the program `ex1`. Save the output of the program in a file `ex1.txt`.
- **Note:** If you `chroot` and there are no binaries like `cd`, `touch`, etc., you may want to first copy them to the chrooted location.

Exercise 1 (5/5)

- Run the same program again (DON'T change the root directory of the process). Append the output to the file `ex1.txt`.
- What is the difference between the outputs in both cases? Add your findings to the file `ex1.txt`.
- You have to put all commands that you executed in a script *ex1.sh*.
- You have to submit all files *ex1.c*, *ex1.txt*, *ex1.sh*.
- **You must explain each command line in the script and add the explanation as a comment to the same script.**
- The results should be reproducible by running the scripts *ex1.sh*
- Don't forget to add `sudo` for commands which need that permission.