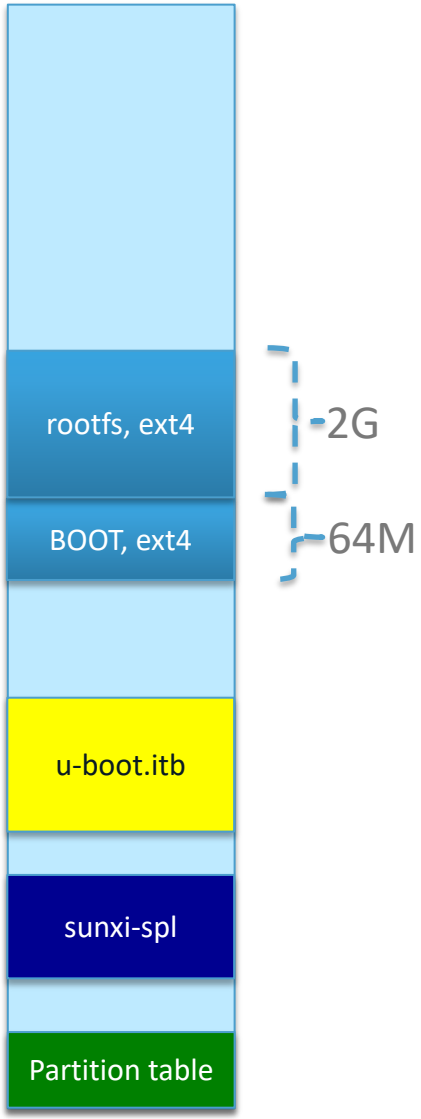
Ecole d'ingénieurs et d'architectes de Fribourg
Hochschule für Technik und Architektur Freiburg

5: File system

Hes·so FRIBOURG FREIBURG
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz

SDCard configuration

- Since the u-boot laboratory, the SDcard has this configuration:



Kernel and rootfs configuration

mkfs.ext2 and tune2fs

```
cd workspace/nano/buildroot
make busybox-menuconfig
```

Go to “Linux Ext2 FS Progs” → [*] tune2fs

Go to “Linux System Utilities” → [*] mkfs.ext2

Btrfs, NILFS2, F2FS, XFS

```
cd workspace/nano/buildroot
make linux-menuconfig
```

Go to “File Systems” and activate the filesystems: ext2-3-4, XFS, BTRFS, NILFS2, F2FS

Important remark: Add these filesystems inside the kernel <*>, not as module <M>

```
<*> XFS filesystem support
[ ]   XFS Quota support (NEW)
[ ]   XFS POSIX ACL support (NEW)
```

```
<*> Btrfs filesystem support
[*]   Btrfs POSIX Access Control Lists
[ ]   Btrfs with integrity check tool compiled
```

```
<*> NILFS2 file system support
<*> F2FS filesystem support
[*]   F2FS Status Information (NEW)
[*]   F2FS extended attributes (NEW)
[*]     F2FS Access Control Lists (NEW)
[ ]     F2FS Security Labels (NEW)
```

Kernel: USB mass storage activation

USB Mass Storage

```
cd workspace/nano/buildroot
make linux-menuconfig
```

Device drivers → USB Support →

Important remark: Add these filesystems inside the kernel <*>, not as module <M>

```
<*>   USB Mass Storage support
[*]     USB Mass Storage verbose debug
<*>     Realtek Card Reader support
[*]       Realtek Card Reader autosuspend support
<*>     Datafab Compact Flash Reader support
<*>     Freecom USB/ATAPI Bridge support
<*>     ISD-200 USB/ATA Bridge support
<*>     USBAT/USBAT02-based storage support
<*>     SanDisk SDDR-09 (and other SmartMedia, including DPCM) su
<*>     SanDisk SDDR-55 SmartMedia support
<*>     Lexar Jumpshot Compact Flash Reader
<*>     Olympus MAUSB-10/Fuji DPC-R1 support
<*>     Support OneTouch Button on Maxtor Hard Drives
<*>     Support for Rio Karma music player
<*>     SAT emulation on Cypress USB/ATA Bridge with ATACB
<*>     USB ENE card reader support
<*>     USB Attached SCSI
```

Kernel and rootfs configuration

Cryptsetup

```
cd workspace/nano/buildroot
make menuconfig
```

Go to: target packages → hardware handling → [*] cryptsetup
→ Miscellaneous → haveged //Pseudo random generator

```
cd workspace/nano/buildroot
make linux-menuconfig
```

Go to: device driver → <*> Multiple Devices drivers support (RAID and LVM) → <*>Device mapper support → <*> Crypt target support

Important remark: Add these filesystems inside the kernel <*>, not as module <M>

```
<*>   Device mapper support
[ ]     Device mapper debugging support
< >     Unstriped target
<*>     Crypt target support
< >     Snapshot target
```

Kernel and rootfs configuration

Initramfs

```
cd workspace/nano/buildroot
make linux-menuconfig
```

Go to: General setup ---> [*] Initial RAM filesystem and RAM disk (initramfs/initrd) support

Go to: Device Drivers → Generic Drivers options → [*] Maintain a devtmpfs filesystem to mount at /dev → [*] Automount a devtmpfs at /dev, after the kernel mounted the rootfs

Important remark: Add these filesystems inside the kernel <*>, not as module <M>

```
[*] Initial RAM filesystem and RAM disk (initramfs/initrd) suppor
()    Initramfs source file(s)
```

```
[ ] Support for uevent helper
[*] Maintain a devtmpfs filesystem to mount at /dev
```

- Install new kernel and rootfs on the SDcard

Question 1: EXT4

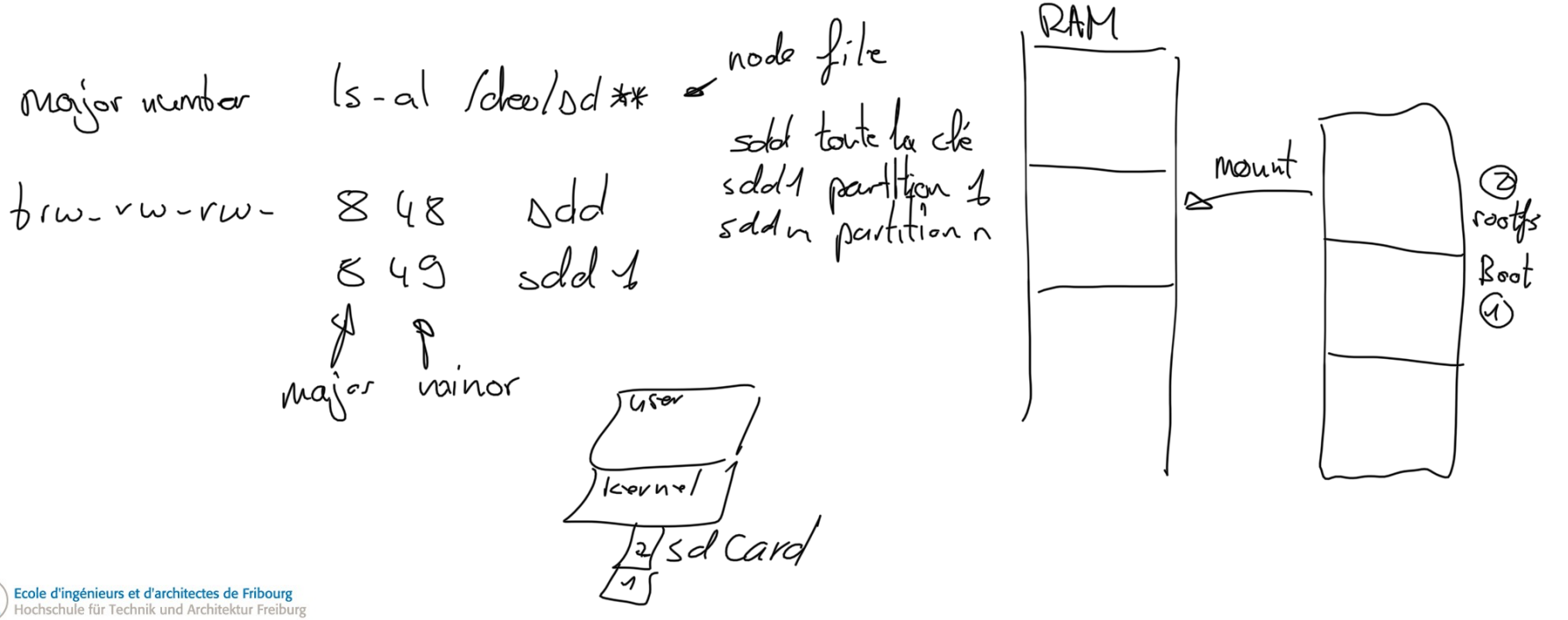
dmesg | info sdb

On NanoPi

Questions

- How the kernel knows that rootfs is in the second partition of the SDcard
- Mount the first partition of the Sdcard on /mnt
- What are the major and minor number of the node file managing the SDcard

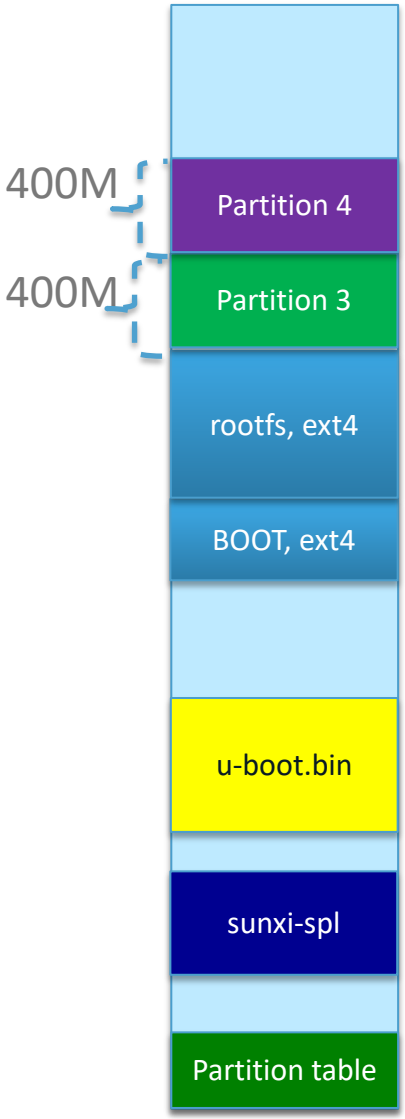
sur le nanoPI



Question 2: btrfs, f2fs, nilfs2, xfs

On PC

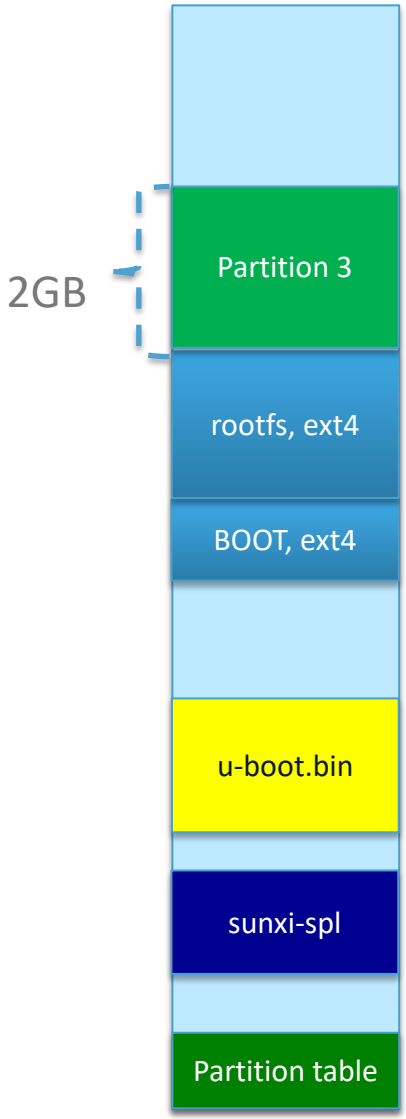
1. Create 2 new partitions (partition3 and 4). You can use fdisk or parted commands (fdisk is simpler)
2. Choose two file systems among the four (e.g. btrfs-f2fs or nilfs2-xfs or …) and format these two partitions with selected file systems
3. Write a program which write 1000 small files of 1024 bytes and one big file of 1MB. Measure the writing time (small and big files) on an ext4 file system (rootfs) and on partitions 3 and 4. On moodle you can find a program skeleton.



Question 3: LUKS, cryptsetup, dmcrypt

On PC

1. Create partition 3. On PC, you can use fdisk or parted commands (fdisk is simpler). For next questions, this partition will be used as a LUKS partition



Question 3.1: LUKS, cryptsetup options

- Cryptsetup: Explain simply what is the difference between the « Plain Mode » « Luks extension mode » (**man cryptosetup**). Which mode is the best to use?
- Cryptsetup: what means the `–hash` option for the luks mode?
- Cryptsetup: What is the default cipher for the luks mode?
- Cryptsetup: What means the `–key-file` option?

Question 3.2: LUKS test 1

On PC

- Initialise a LUKS partition (partition 3) with this option `--pbkdf pbkdf2`, format the LUKS partition as ext4, and mount it in the directory `/mnt/usr`
- Copy a file in the LUKS partition
- Add a new passphrase to the LUKS partition
- Dump the header partition and the crypted master key
- With the `dd` command, dump 1 Mbytes of the partition /dev/sbd3 to a file. Can you find the header partition and the crypted master key
- Connect your SDCard card on NanoPi and activate the crypted partition

Question 3.3: rootfs in a luks partition

The goal of this question is to have a crypted rootfs on partition 3

On Host PC

- Generate a random passphrase in a file:
  With `dd and /dev/urandom` generates the random file « `passphrase` » with 64 bytes
- Initialize a LUKS partion (for partition 3) with these options:
  - `pbkdf pbkdf2`
  - Key-size: 512
  - Passphrase: in the file "`passphrase`"
- Create a mapping `/dev/mapper/usrfs1`
- Format the LUKS partition as ext4 partition
- With the command below, copy the rootfs to the luks partition:

```
sudo dd
if=~/workspace/nano/buildroot/output/image/rootfs.ext4
of=/dev/mapper/usrfs1 bs=4M
```

Question 3.3: rootfs in a luks partition

On NanoPi

- Boot NanoPi and mount manually the luks partition (partition 3).
- Write as init script (/etc/S40luks) in order to mount automatically the partition 3

Question 4: initramfs

- On your PC generate an initramfs but the `/init` script don't execute the `exec switch_root` command but the `exec sh`

```
mount -t proc none /proc
mount -t sysfs none /sys
mount –t ext4 /dev/mmcblk1p2 /newroot
mount -n -t devtmpfs devtmpfs /newroot/devexec sh

exec sh
# exec switch_root /newroot /sbin/init
```

- Initialize NanoPi in order to start the initramfs

- Start NanoPi and start manually the `exec switch_root` command

Question 5: initramfs-LUKS partition

- From the shell on the initramfs (exec sh), mount the partition 3 as LUKS partition.
- Start manually the `exec switch_root` command to this encrypted rootfs partition
- Write a script in order to start automatically the encrypted rootfs partition