

## ПОСТАВКА

Позиционирајте се у директоријум `~/linux-kernel-labs/src/linux` који садржи кернел за RPI са github репозиторијума <https://github.com/raspberrypi/linux> и затим од полазне гране **rpi-4.4.y** направите нову грану **zad1** и позиционирајте се у исту:

- ➔ `cd ~/linux-kernel-labs/src/linux`
  - ➔ `git checkout rpi-4.4.y`
  - ➔ `git checkout -b zad1`
- 

Позиционирајте се у директоријум `~/linux-kernel-labs/modules` који садржи коренски систем датотека за RPI и затим од полазне гране **master** направите нову **zad1** и позиционирајте се у исту:

- ➔ `cd ~/linux-kernel-labs/modules`
  - ➔ `git checkout master`
  - ➔ `git checkout -b zad1`
- 

Распакујте архиву са коренским системом датотека ( **rootfs.tar.bz2**) у директоријум **nfsroot/**.

- ➔ `sudo cp rootfs.tar.bz2 ~/.../modules/nfsroot`
  - ➔ `sudo tar xvjf rootfs.tar.bz2`
  - ➔ `sudo chown -R rtrk.rtrk nfsroot` (ili koji vec folder)
- 

## ПРИМЕНА ЗАКРПЕ

Примените закрпу **zad1\_defconfig.patch.bz2** на линукс изворни код

- ➔ `cd ~/../src/linux`
  - ➔ `bzcat (apsolutna putanja) zad1_defconfig.patch.bz2 | patch -p1`
- 

## КОНФИГУРАЦИЈА КЕРНЕЛА

- ➔ `cd ~/linux-kernel-labs/src/linux`
  - ➔ `export ARCH=arm`
  - ➔ `export CROSS_COMPILE=arm-linux-gnueabi-`
  - ➔ `make bcm2709_rtrk_linux_kurs_defconfig`
  - ➔ `make xconfig`
    - oznaci sve koji se traza
  - ➔ `sudo /etc/init.d/nfs-kernel-server restart`
  - ➔ `make -j4`
  - ➔ `cd ~/linux-kernel-labs/src/linux/arch/arm/boot`
  - ➔ `sudo cp zImage /var/lib/tftpboot/`
- 

## ДОДАВАЊЕ ЈЕДНОСТАВНОГ МОДУЛА

Написати једноставан знаковни руковалац који на иницијализацији динамички заузима бафер. Величину бафера проследити као аргумент при учитавању модула.

Додати **write** функцију која уписује добијени садржај у бафер.

Додати **read** функцију која исчитава садржај бафера.

Омогућити приступ модулу из корисничког простора креирањем одговарајуће датотеке уређаја уколико је потребно. Тестирати додате функције. За тестирање **write** операције користити **echo** за **read** операције користити команду **cat**.

- ➔ У Makefile dodaj da ti pravi taj modul.ko
- ➔ - Učitavanje novog modula u kernel: `insmod <ime_modula.ko>`

- ➔ // - Brisanje učitanoг модула `rmsmod <ime_modula>`
  - ➔ Kreiranje fajla za komunikaciju sa modulom: `mknod <putanja/do/fajla> <tip> <major broj> <minor broj>`
  - ➔ Ako se dobro secam to je ono `hello.c` i u njega upisujes za `write` i `read`
  - ➔ `cd ~/.../ nfsroot/root/hello`
  - ➔ Za pisanje u modul: `echo "<podaci>" > <putanja do uređaja>`
  - ➔ Za čitanje модула: `cat <putanja do uređaja>`
- 

#### ДОДАВАЊЕ ПРОЦ СПРЕГЕ

// 16. linija, 96-99, 126. i dve funkcije to je promena u modulu

---

#### ДОДАВАЊЕ У ЛИНУКС СТАБЛО

Одговарајућом скриптом проверити форматирање новог модула и додати код у линукс стабло. По додавању изменити потребне датотеке да би нови руковалац постао видљив из конфигурацијских алата и да би могао да се преведе и као модул и заједно са керналом.

##### PROVERA FORMATIRANJA

- ➔ `cd ~/linux-kernel-labs/src/linux`
- ➔ `scripts/checkpatch.pl -h`
- ➔ `~/linux-kernel-labs/src/linux/scripts/checkpatch.pl--no-tree hello_version.c`

##### DODAVANJE KODA U LINUXS STABLO

- ➔ napravimo novi `bcm2709-rpi-2-b-novi.dts` (`linux/arch/arm/boot/dts`)
- ➔ iz originalnog `bcm2709-rpi-2-b.dts` kopiramo u ovaj novi
- ➔ u novom izbrisemo sve sem `DTS` include direktive, definiciju modela, compatible podesavanje a od definicije uređaja zadrži samo `uart` i `core`. (ovo zavisi od toga šta nam treba, nije uvek ovako)
- ➔ U `Makefile` koji se tu nalazi dodamo samo liniju koja će imati naziv `bcm2709-rpi-2-b-novi.dtb` po uzoru na `vec` postojeće.
- ➔ `bcm2709-rpi-2-b-novi.dtb` kopiramo u `/var/lib/tftpboot/`
- ➔ obavezno promeniti u `bootcmd` da na `0x02000000` bude `bcm2709-rpi-2-b-novi.dtb`

##### PODESITI DA SE RUKOVALAC VIDI IZ KONF ALATA

- ➔ `cd ~/linux-kernel-labs/src/linux/drivers/misc`
  - ➔ `gedit Kconfig &`
    - tu dodamo još jedan blok identican na pocetak
      - `config DRAJVER`
        - `tristate`
        - `default m`
        - `help`
  - ➔ Zatim u `Makefile`-u po uzoru na ostalo dodamo `obj-$(CONFIG_DRAJVER) += drajver.o` // `drajver.c` ili kako se `vec` zove
  - ➔ `cd ~/linux-kernel-labs/src/linux`
  - ➔ `make xconfig`
  - ➔ ОТКАЦИ НОВИ DRAJVER
  - ➔ `make -j4`
  - ➔ `sudo cp zImage /var/lib/tftpboot/`
  - ➔ u `buildroot`-u na `rpi`-ju: `modprobe ime_drajvera`
-

NIJE NEOPHODNO  
УЧИТАВАЊЕ НОВОГ МОДУЛА

Превести додати руковалац као модул и учитати га коришћењем ***modprobe***.

```
sudo apt-get install gcc-arm-linux-gnueabi
```

```
modinfo <module_name>
```

```
modinfo <module_path>.ko
```

dobijemo informacije o modulu (parametri, licenca....)

```
sudo insmod <module_path>.ko
```

ocitavanje modula

```
sudo modprobe <module_name>
```

Najčešća upotreba modprobe: pokušaj učitavanja svih modula od kojih dati modul zavisi praćen učitavanjem datog modula