

ПОСТАВКА

Позиционирајте се у директоријум `~/linux-kernel-labs/src` и затим од полазне гране `master` направите нову грану `dan06` и позиционирајте се у исту:

- ➔ `git checkout master`
 - ➔ `git checkout -b zad2`
-

Преузмите изворни код `busybox`-а командом:

`wget http://www.busybox.net/downloads/busybox-1.21.1.tar.bz2`

и распакујте добијену архиву

- ➔ `sudo tar xvf busybox-1.21.1.tar.bz2`
-

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 `dan06` и позиционирајте се у исту:

- ➔ `git checkout rpi-4.4.y`
 - ➔ `git checkout -b dan06`
-

КОНФИГУРАЦИЈА И ПРЕВОЂЕЊЕ BUSYBOX-A

За конфигурацију Линукса искористите конфигурациону датотеку `linux.config` и преведите Линукс кернел

- ➔ `cd ~/linux-kernel-labs/modules/data/linux.config`
 - ➔ `cp linux.config ~/linux-kernel-labs/src/linux`
 - ➔ `cd ~/linux-kernel-labs/src/linux/`
 - ➔ `mv linux.config .config`
 - ➔ `ls -la //da li je promenio naziv u .config`
 - ➔ `sudo chown -R rtrk.rtrk /naziv foldera`
 - ➔
 - ➔ `export ARCH=arm`
 - ➔ `export CROSS_COMPILE=arm-linux-gnueabi`
 - ➔
 - ➔ `make -j4 //prevodjenje kernela, izgenerisace zImage koji se nalazi na putanji :`
 - ➔ `/linux-kernel-labs/src/linux/arch/arm/boot/`
-

ДОДАВАЊЕ НОВОГ АПЛЕТА

Додајте нови бусибок аплет искористите предност са следећим функционалним уређајима. Аплет прима 2 или више параметара. Сви параметри су текстуалне датотеке, са тим што је последњи параметар излазио а сви пре њега су унети. Потребно је отворити све датотеке које су доступне као улазни параметри и њихови садржај преписали у УРЛ-у који је одредио излазни параметар (последње посећено). При уписивању садржаја у излажном УРЛ-у претворили сте сва мала слова у велику.

Измените све потребне датотеке како би аплет могао да предузме заједно са заузетом сликом такође обрадити греску изазван погрешним пројектним параметрима при покретању аплета на исправан начин (у стилу бусибоба)

- ➔ - U fajl `**include/applets.src.h**` dodati:
 - `IF_<IME_APPLETA>(APPLET(<ime_appleta>, BB_DIR_USR_BIN, BB_SUID_DROP))`
- ➔ U fajl `**include/usage.src.h**` dodati opise:

- #define <ime_appleta>_trivial_usage \
 - "[param1] [param2] ..."
 - #define <ime_appleta>_full_usage \
 - "Opis programa"
 - ➔ U fajl ****miscutils/Config.src**** dodati konfiguraciju za buildovanje:
 - config <IME_APPLETA>
 - bool "<ime_appleta>"
 - default n
 - depends on LFS
 - help
 - Opis appleta koji se ispisuje u meniju
 - ➔ U fajl ****miscutils/Kbuild.src**** dodati izgenerisani fajl:
 - lib-\$(<IME_APPLETA>) += <ime_appleta>.o
 - ➔ potrebno je napisati kod appleta, koji će se prevesti i vezati prilikom bildovanja:
 - #include "busybox.h"
 -
 - int <ime_appleta>_main(int argc, char **argv)
 - {
 - if (<neispunjen_uslov_pozivanja) {
 - bb_show_usage(); // Prikazuje poruku o tome kako se poziva program
 - }/*Kod appleta */}
 - ➔ Nakon ovoga potrebno je otkaćiti applet u menuconfig-u, kako bi se preveo (`make menuconfig`).
- Applet će se nalaziti ****miscutils**** kategoriji.
- Postupak je isti i za neku drugu.
- ➔

КОНФИГУРАЦИЈА И ПРЕВОЂЕЊЕ BUSYBOX-A

Позиционирајте се у директоријум у који сте распаковали busybox. За конфигурацију busybox-a искористите конфигурациону датотеку `~/linuxkernel-labs/modules/data/tinyBusybox.config`. Додатно, укључите опцију да се busybox преведе са статички увезаним библиотекама и подесите путању на коју ће се инсталирати. Преведите и инсталирајте busybox у празан директоријум. Покрените претходно преведени Линукс кернел. За rfs користите директоријум у који је инсталиран busybox, преко мреже (NFS). Додајте све потребне датотеке уређаја да би се добила функционална Линукс конзола.

- ➔ `cd /linux-kernel-labs/modules/data/`
- ➔ `cp tinyBusybox.config ~/linux-kernel-labs/src/busybox-1.21.1`
- ➔ `cd ~/linux-kernel-labs/src/busybox-1.21.1`
- ➔ `mv tinyBusybox.config .config`
- ➔ `make menuconfig` // укључити опцију да се busybox преведе са статички увезаним библиотекама, подесити путању на којој се инсталира ако је то тражено.
- ➔ `make` // преводjenje
- ➔ `make_install` // instaliranje
- ➔ `sudo gedit /etc/exports/`
- ➔ променити путању за rfs(root file system) на путању у којој је инсталиран busybox

- ➔ /linux-kernel-labs/src/busybox-1.21.1/_install
 - ➔ mkdir /linux-kernel-labs/src/busybox-1.21.1/_install/dev/
 - ➔ ls -l /dev -> koji je major i minor broj
 - ➔
 - ➔ mknod /linux-kernel-labs/src/busybox-1.21.1/_install/dev/tty1 c 4 1
 - ➔ mknod /linux-kernel-labs/src/busybox-1.21.1/_install/dev/tty2 c 4 2
 - ➔ mknod /linux-kernel-labs/src/busybox-1.21.1/_install/dev/tty3 c 4 3
 - ➔ mknod /linux-kernel-labs/src/busybox-1.21.1/_install/dev/tty4 c 4 4
 - ➔ mknod /linux-kernel-labs/src/busybox-1.21.1/_install/dev/ttyAMA0 c 204 64
 - ➔ mknod /linux-kernel-labs/src/busybox-1.21.1/_install/dev/console c 5 1
-

ДОДАВАЊЕ PROC И SYS СИСТЕМА ДАТОТЕКА

Направите директоријуме /etc/init.d, /proc и /sys у оквиру rfs-a. На proc директоријум маунтовати proc систем датотека, а на sys директоријум sysfs. Аутоматизовати моунтовање ових система датотека на старту система тако што ћете направити и на адекватан начин попунити датотеку /etc/init.d/rcS. То је скрипта коју треба да покрене први кориснички програм (скрипта), /etc/inittab, који такође треба направити и попунити са:

```
::sysinit:/etc/init.d/rcS
ttyAMA0::askfirst:/bin/sh
```

- ➔ cd _install -> rfs /linux-kernel-labs/src/busybox-1.21.1/_install
 - ➔ mkdir proc
 - ➔ mkdir sys
 - ➔ mount -t sysfs sys sys
 - ➔ mount -t proc proc proc
 - ➔
 - ➔ mkdir etc
 - ➔ cd etc
 - ➔ mkdir init.d
 - ➔ cd init.d
 - ➔ sudo gedit rcS
 - mount -t proc none /proc
 - mount -t sysfs none /sys
 - ako se trazi web sprega
 - /usr/sbin/httpd -h /www/ &
 - Provera: http://ip_rpi
 - ➔ cd ..
 - ➔ sudo gedit inittab
 - ::sysinit:/etc/init.d/rcS
 - ttyAMA0::askfirst:/bin/sh
-

ГЕНЕРИСАЊЕ INITRAMFS

Искористите генерисани rfs као изворне датотеке за генерисање initramfs-a (CONFIG_INITRAMFS_SOURCE) и поново преведите кернел. Претходно је потребно преименовати linuxrc у init, у коренском директоријуму rfs-a. Покрените initramfs на RPI.

Потребно је изменити bootargs варијаблу у u-boot-у тако да у њој остану само параметри rw и console.

По покретању initramfs-а додајте све што је потребно да се оствари иста функционалност као и када је rfs био на мрежи.

- ➔ cd /linux-kernel-labs/src/linux
- ➔ make xconfig
- ➔ pronaci *Initramfs source file(s)* i dodati putanju rfs (/linux-kernel-labs/src/busybox-1.21.1/_install)
- ➔ sacuvati izmene
- ➔ cd /linux-kernel-labs/src/busybox-1.21.1/_install/
- ➔ mv linuxrc init
- ➔ make -j4 //ponovo prevesti kernel

- ➔ U U-boot-u:
- ➔ izmeniti bootargs (editenv bootargs) tako da останu samo parametri rw i console

- ➔ ako ne bude radilo, ponoviti korak iz konfiguracije busyboxa od komande ls -l /dev za sve neophodne datoteke