

# Embedding FreeBSD, KyivBSD'12

Aleksandr Rybalko

ZRouter.org

28 сентября 2012 г.

KyivBSD'12

# Complete picture (almost)

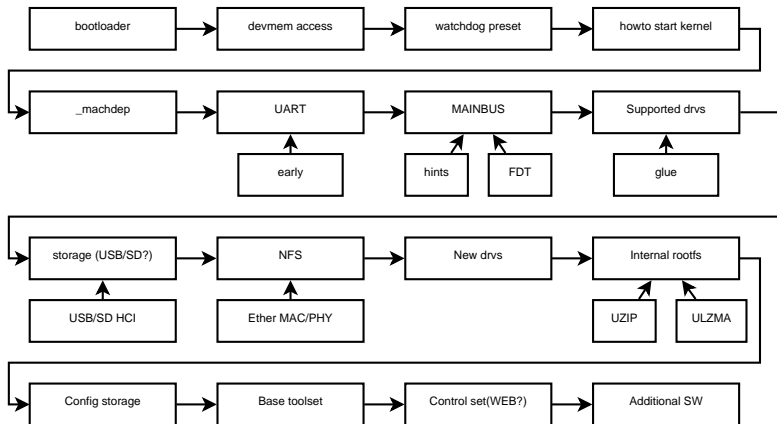


Рис.: It's how I do that

# bootloader

## bootloader

- REDBOOT
- U-Boot
- CFE
- ...

# Accessing devices memory

What we need?

- Memory write
- Memory read
- Memory dump

What if we don't have any? Try to:

- Make second loader (U-Boot)
- Make some loader application.
- Skip to «Ways to start kernel» :-)

# Enable watchdog

Enable watchdog and preset it to maximum value.  
Then you will not require to have direct access to device and/or remote controlled power switch.

# Ways to start kernel

Find all possible ways how to start kernel - helps for development.

U-Boot example:

`go 0x00001000` - pass control to address 0x00001000.

`bootm 0x00000000` - boot OS packed in U-Boot image loaded at 0x00000000.

Find standard way to start kernel - helps for usage by users.

Most U-Boot devices do `bootm` on address in Flash memory right after U-Boot + his Environment block:

`0xbf000000-0xbf040000` - U-Boot

`0xbf040000-0xbf050000` - U-Boot Environment

`0xbf050000-0xbfxx0000` - OS kernel image packed with **mkimage** tool

# Chip family startup code

We have code for platform start (ARM/MIPS/PPC/etc.), now we want our own point to start.

Copy one of already done `_machdep.c` files for that platform :-)

# UART

We need UART, basic support of serial controller much more easy than many other type of text communications.

But please remember, at early stage we:

- don't have DELAY(or it don't work)
- we still not have interrupts (controller not initialized)
- and we still don't have correct virtual bus address (we define it)



# Main system bus

## MAINBUS

- autodetect - good to have :-)
- hints - easy but too simple
- FDT - complicated but powerful

# Already supported devices

- Search source tree to find, maybe we already have such driver.
- Maybe some modification.
- Maybe split driver into driver itself and bus glue.
- Maybe we can just add bus glue.

# Storage support

## Storage

- USB - require USB HCI (uhci/ohci/ehci/xhci/dwcotg - cover 95% devices)
- SD - require SDHCI (less standard, but may be same for most SoCs of that vendor)

# Network storage support

NFS (Network drivers required)

MAC (find || write new)

PHY support (easy)

switch (ask #bsdmins or #zrouter)

# Time for new drivers

New drivers implementation

Hints: Make drivers system/arch independent, SoC vendors are lazy and device cores may migrate to SoCs with different arch or even different vendors.

# Root FS inside box

## Internal rootfs

- cd9660 with gz, GEOM\_UZIP
- cd9660 with lzma, GEOM\_ULZMA
- nandfs like
- something for NOR
- Filesystem compression ???
- ...

# How can we pack that all into image?

- mdconfig + dd/tar
- makefs
- makefs + mtree
- makefs + mtree (tuned to NetBSD one)

# Footshooting or updating rootfs on working system

- two rootfs - require more flash
- mfs rootfs - require more ram
- geom alias (/dev/map/upgrade) + geom intersection
- How to partially preload shared library ???
- how can we notify system to do refresh of FS metadata ???



# Where to store configs?

Config storage:

- etc.tar.gz
- FS mounted in Read-Write mode
- network-autoconfig - device can get everything configured from network (DHCP, PPP, SNMP, etc.)

# What can fit into our small flash?

Base toolset:

- standard base
- reduced base (only few required tools)
- bsdbox
- /bin/sh "applets" ???

# Reduce userland size

- limit library symbols (objcopy excluding unused)
- cut off unused(for embedded) code in apps
- alternative tools set
- tools embedded into /bin/sh

# How we will control it?

## Control set

- shell access + editors
- WebUI
- SNMP
- ...

# We need more!

Additional software (ports cross-building)

# Контакты

Спасибо за внимание.

Александр Рыбалко

<ray@ddteam.net>

# TOC

- 1 Complete picture
- 2 bootloader
- 3 Accessing devices memory
- 4 Enable watchdog
- 5 Ways to start kernel
- 6 Chip family startup code
- 7 UART
- 8 Main system bus
- 9 Already supported devices
- 10 Storage support
- 11 Network storage support
- 12 Time for new drivers
- 13 Root FS inside box
  - How can we pack that all into image?
  - Footshooting or updating rootfs on working system
- 14 Where to store configs?