

FreeBSD 勉強会

PC以外で動く FreeBSD

佐藤 広生 <hrs@FreeBSD.org>

東京工業大学/ FreeBSD Project

2012/11/2

講師紹介

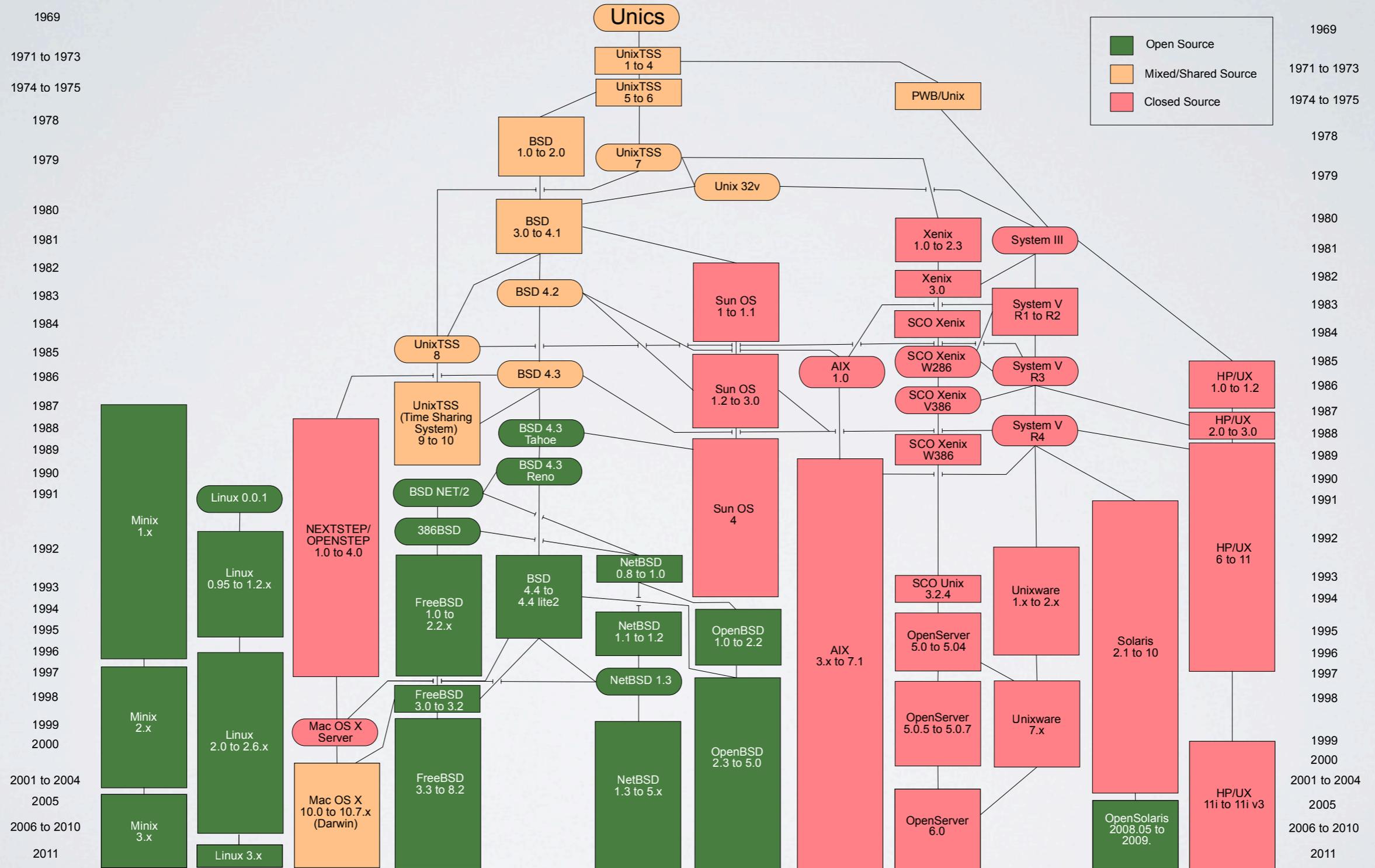
佐藤 広生 <hrs@FreeBSD.org>

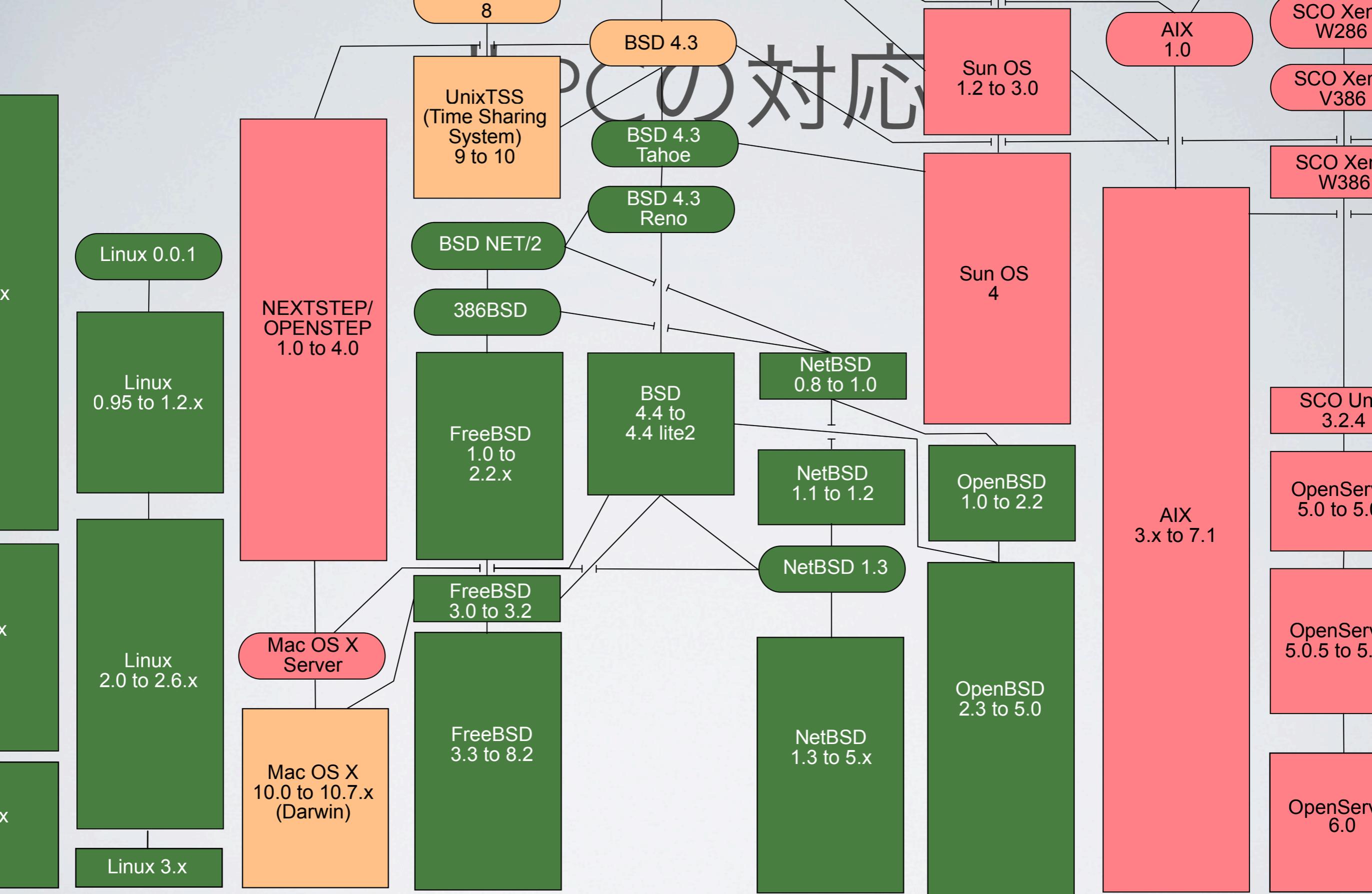
- ▶ *BSD関連のプロジェクトで10年くらい色々やっています
 - ▶ 何でも屋：カーネル開発・ユーザランド開発・文書翻訳・サーバ提供などなど
 - ▶ FreeBSD コアチームメンバ(2006 年から4期目)、リリースエンジニア
(commit 比率は src/ports/doc で 1:1:1 くらい)
 - ▶ AsiaBSDCon 主宰
 - ▶ 技術的なご相談や講演・執筆依頼は hrs@allbsd.org まで

お話すること

- ▶ **PC以外で動くFreeBSDの現状**
 - ▶ Intel iAPx86 系アーキテクチャ (IA-32, Intel 64)以外で動作するもの
 - ▶ pc98, ia64, sparc64, powerpc, powerpc64, arm, mips
- ▶ **小型機器で使ってみよう**
 - ▶ 技術的な話よりも実物をいじってみる!
 - ▶ EPSON NP11 (Intel Atom)
 - ▶ OpenBlockS A6 (ARM9E)
 - ▶ DreamPlug (ARM9E, 頒布)
 - ▶ TL-WR1043ND (MIPS32, 頒布)

非PCの対応





非PCの対応

- ▶ BSD 系 OS の出発点は 4.3BSD Net/2 → 386BSD
- ▶ 4.3BSD 近辺は DEC VAX 全盛期
 - ▶ 1977年
 - ▶ 32-bit CISC
 - ▶ でっかい (1.2m x 1.5m)



非PCの対応

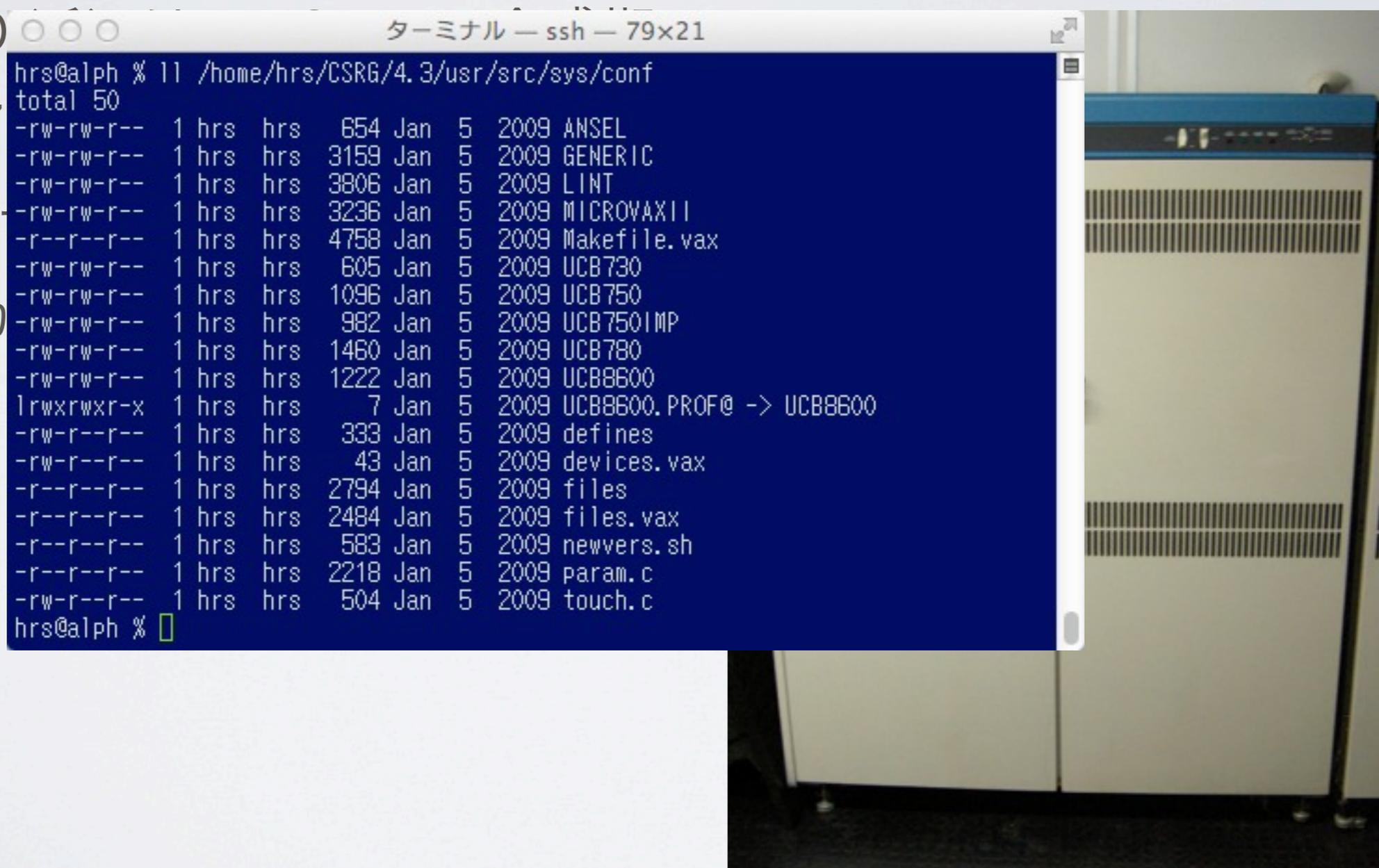
- ▶ BSD系OSの出発点は4.3BSD Net/2 → 386BSD

- ▶ 4.3BSD

- ▶ 1977

- ▶ 32-bit

- ▶ でっか



非PCの対応

- ▶ FreeBSD は i386 PC での動作にフォーカス (当初)
 - ▶ 4.X: DEC alpha (2000/3)
 - ▶ 5.X: NEC PC98x1, AMD64, Intel 64, Sun UltraSPARC (2003/1)
 - ▶ 7.X: PowerPC(32-bit), DEC alpha phase-out
 - ▶ 9.X: PowerPC(64-bit), ARM, MIPS(32-bit)
- ▶ FreeBSD/xxx と表記 (たとえば FreeBSD/arm)

非PCの対応

- ▶ FreeBSD は i386 PC での動作にフォーカス (当初)
 - ▶ 4.X: DEC alpha (2000/3)
 - ▶ 5.X: NEC PC98x1, AMD64, Intel 64, Sun UltraSPARC (2003/1)
 - ▶ 7.X: PowerPC(32-bit), DEC alpha phase-out
 - ▶ 9.X: PowerPC(64-bit), ARM, MIPS(32-bit)
- ▶ FreeBSD/xxx と表記 (たとえば FreeBSD/arm)
- ▶ Tier-1: リリースとパッケージを作成、セキュリティ勧告あり
- ▶ Tier-2: リリースは作らないがコンパイル可能な状態を維持
- ▶ i386, amd64 以外は基本的に Tier-2 扱い

使えるの？

- ▶ ① リリースがあって、ほぼ使えるもの
 - ▶ FreeBSD/i386, amd64, pc98, ia64, sparc64
- ▶ ② リリースがあるけれど、落とし穴もたくさん
 - ▶ FreeBSD/powerpc, powerpc64
- ▶ ③ リリースがないので、自分でつくらないとダメ
 - ▶ FreeBSD/arm, mips

ほぼ使えるもの

▶ i386, amd64

- ▶ 最もサポートが手厚く、性能が低下しないように配慮
- ▶ 強いて言うなら Intel や Supermicro 等、協力関係にあるベンダのハードウェアがトラブル少なし

▶ pc98, ia64, sparc64

- ▶ FreeBSD を読み込ませるための操作方法を知っておくことが必要
- ▶ 一度起動さえできれば、後の操作は変わらない
- ▶ MP対応もちゃんとある (pc98 以外)

ほぼ使えるもの

- ▶ 入手性の高い機器の動作実績 (sparc64 編)
 - ▶ UltraSPARC II, III, IV, V
 - ▶ Sun Ultra, Netra, Blade 1x0, Enterprise
 - ▶ Sun Blade xx00, Fire
 - ▶ 富士通 PRIMEPOWER x50
 - ▶ SPARC ラップトップでも動く
 - ▶ 起動
 - ▶ OpenFirmware (IEEE 1275) からUFSパーティションのローダを読む。ラベルは SMI VTOC8 が使える (<2TB)
 - ▶ CD-ROM やネットワークから起動可能

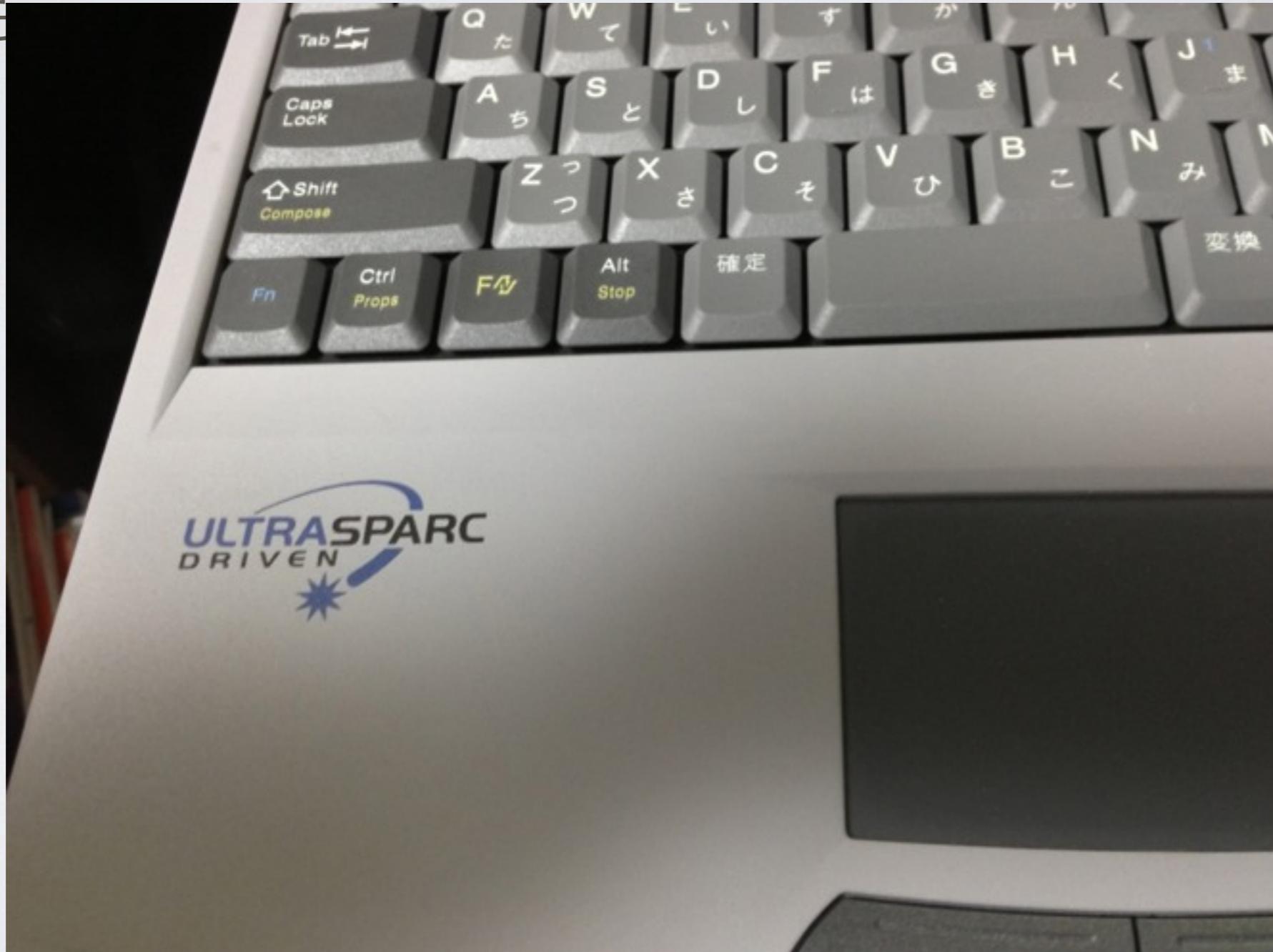
ほぼ使えるもの

▶ デモ



ほぼ使えるもの

▶ デモ



ほぼ使えるもの

- ▶ 入手性の高い機器の動作実績 (ia64 編)
 - ▶ itanium, itanium2 (Marced, Madison, Deerfield あたり)
 - ▶ HP rx2600, zx6000 (itanium, itanium2)
 - ▶ SGI Altix 350
 - ▶ 起動
 - ▶ EFI shell からFATパーティションに置いたローダを読む
 - ▶ PXE でネットワーク起動

ほぼ使えるもの

▶ デモ



注意しないと厳しいもの

▶ powerpc, powerpc64

- ▶ 実際のターゲットは G4/G5 Mac。MP 対応。
- ▶ リリースのCDROMイメージはちゃんと提供されており、導入は比較的容易
- ▶ 組み込み用 powerpc チップ(Freescale MPC8572など)は開発されているものの、ソースツリーには入っていない
- ▶ パッケージが少ない (コンパイルできないものがある)

注意しないと厳しいもの

- ▶ 入手性の高い機器の動作実績 (powerpc 編)
 - ▶ IBM PowerPC 7400 (Xserve G4)
 - ▶ IBM PowerPC 970FX (Xserve G5)
 - ▶ PPC Mac の大半が動く (Intel Mac は amd64 で動く)
- ▶ 起動
 - ▶ OpenFirmware (IEEE 1275) から APM(Apple Partition Map)パーティションに置いたローダを読む (ただしMacのOF実装はかなり適当)
 - ▶ ネットワーク起動も可能

注意しないと厳しいもの

▶ デモ



いばらの道

▶ arm, mips

- ▶ リリースが存在しない。起動の方法は機器によって異なるため、統一的な配布イメージが存在しない。
- ▶ カーネルとユーザランドをコンパイルして、そこから起動させるための手順が手動で必要になる
- ▶ 周辺機能を組み込んだCPU（SoCと呼ぶことが多い）がほとんどで、PCのようにハードウェアを自動検出する機能が乏しいため、対象機器の構造を把握した上でカーネル設定ファイルをつくる必要がある

いばらの道

▶ ARM 基礎知識

- ▶ ARM ファミリという分類:

ARM9TDMI, ARM9E, Cortex-A8...

- ▶ アーキテクチャ (命令セット) のバージョン:

ARMv4T, ARMv5TEJ, ARMv6....

- ▶ 組み合わせさせて「ARMコア」と呼ぶリファレンス実装

ARM9 + ARMv5TEH = ARM926EJ-S

- ▶ 別のメーカーが、このコア and/or アーキテクチャのライセンス提供を受けてCPUをつくる

Marvell 88F6281 = ARM9 + ARMv5TE

いばらの道

- ▶ 入手性の高い機器の動作実績 (arm 編)
 - ▶ ARM9 (ARMv5TE)
 - ▶ Marvell SoC: Orion, Kirkwood
 - ▶ OpenBlockS A6 (88F6282)
 - ▶ DreamPlug (88F6281)
 - ▶ OpenRD (88F6281)
 - ▶ Atmel AT91
 - ▶ 比較的安定して動作



いばらの道

- ▶ 入手性の高い機器の動作実績 (arm 編)
 - ▶ ARM11 (ARMv6)
 - ▶ Raspberry Pi (ARM1176JZFS)
 - ▶ Cortex-A8 (ARMv7)
 - ▶ BeagleBone (TI AM3359)
 - ▶ ~~Cortex-A9 (ARMv7)~~
 - ▶ ~~Marvell ARMADA XP~~
- ▶ 最近少しずつ拡充中



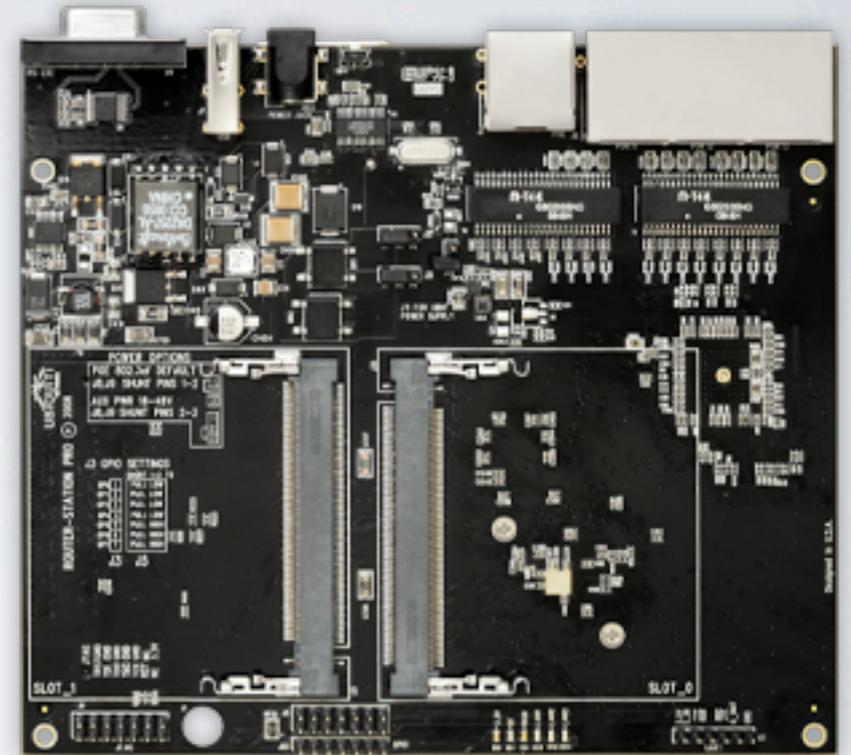
いばらの道

▶ MIPS 基礎知識

- ▶ アーキテクチャ分類 (昔の MIPS I, II, III,...)
MIPS32 24K, 1004K,
MIPS64 5K, 20K,
- ▶ ライセンスを提供してメーカーがつくったチップ名
Atheros AR71xx, AR724x, AR913x (MIPS32 24K)
Cavium OCTEON (MIPS64, MP)
NetLogic XLR/XLS (MIPS64, MP), Broadcom に買収
- ▶ 巷の無線LANルータによく使われている
- ▶ OpenWRT プロジェクト

いばらの道

- ▶ 入手性の高い機器の動作実績 (mips 編)
 - ▶ Atheros AR71xx
 - ▶ Ubiquiti Router Station Pro
 - ▶ TP-Link TL-WR1043ND



組み込み機器の起動

▶ PC と違うところ

- ▶ キーボードやスクリーンといった入出力デバイスがない
- ▶ 記憶装置もいろいろ：
SD カード、フラッシュメモリ、HDD、...

▶ 電源を入れてから起動するまでに働くプログラム

- ▶ PCはBIOSやEFIと呼ばれるIPLがあり、みんな一緒
- ▶ PC以外はいろいろある
 - ▶ OpenFirmware
 - ▶ U-Boot
 - ▶ RedBoot

組み込み機器の起動

- ▶ まず知らなければならないこと
 - ▶ IPLの種類
 - ▶ 入出力デバイスはどこだろうか？
 - ▶ ほとんどの場合、シリアルコンソールを使う
 - ▶ 文字でやりとり
 - ▶ OSはどこに入っているだろうか？
- ▶ FreeBSDが対応しているアーキテクチャであれば、上記の情報さえ分かれば、試すことまでは少なくともできる。
- ▶ 必要な情報は、えらいひとに聞きましょう。

小型機器でやってみよう

▶ EPSON NP-11

- ▶ 単なる小型のPC です。
- ▶ 「PCと全然違うつくりのやつはちょっと分からないけど、小さいのが欲しい」という方に。
- ▶ USB メモリを使ってインストールできます。
- ▶ 現行はNP30Sという型番。
AMD C-60, 2GB MEM, 250GB
HDD で 3万円くらい



小型機器でやってみよう

▶ カーネルとユーザランドプログラムの再構築

```
⌘ cd /usr/src
```

```
⌘ make -j4 buildworld
```

/usr/src の内容を使ってコンパイラをつくる

そのコンパイラでユーザランドをコンパイルする

コンパイル結果は /usr/obj に格納される (OBJDIR)

```
⌘ make -j4 buildkernel
```

/usr/src/sys の内容を使ってカーネルをつくる

コンパイラは、buildworld でつくったものが使われる

コンパイル結果は /usr/obj/ に格納される

```
⌘ make installkernel DESTDIR=/
```

/usr/obj にあるカーネルのコンパイル結果を、\$DESTDIR/boot/kernel へコピー

```
⌘ make installworld DESTDIR=/
```

/usr/obj にあるユーザランドのコンパイル結果を、\$DESTDIR/ へコピー

小型機器でやってみよう

- ▶ OpenBlockS A6 (ARM)

- ▶ |
- ▶ ○
- ▶ し
- ▶ 示



り) を

小型機器でやってみよう

- ▶ **OpenBlockS A6 (ARM)**
 - ▶ IPL は U-Boot を使っている
 - ▶ 0x9000000 にカーネル（ELFではなく生バイナリ）をロードすれば動く
 - ▶ USB メモリにFreeBSDシステムを入れて起動させることにチャレンジ

小型機器でやってみよう

- ▶ OpenBlockS A6 (ARM)

- ▶ IPI

- ▶ 0x

- 口

- ▶ US

- 起



リ) を

小型機器でやってみよう

▶ まずはコンパイル

```
% cd /usr/src  
% make TARGET_ARCH=arm TARGET=arm -j4 buildworld  
% make TARGET_ARCH=arm TARGET=arm -j4 buildkernel KERNCONF=OPENBLOCKS_A6
```

▶ PCでクロスコンパイルする

- ▶ TARGET_ARCH=arm と TARGET=arm を指定すれば OK
- ▶ コンパイル結果は、ARMマシンで動作するものになる
- ▶ OBJDIR は /usr/obj/arm.arm/usr/src/sys/... になる
- ▶ PCにインストールしちゃだめ
- ▶ 注：OPENBLOCKS_A6 というカーネル設定ファイルはまだツリーに入っていません。

試したい方は、後ほどダウンロード URL を紹介します。

小型機器でやってみよう

- ▶ コンパイル結果を一時ディレクトリにインストール

```
% cd /usr/src  
% make TARGET_ARCH=arm TARGET=arm installworld distribution DESTDIR=/armroot  
% cp /usr/obj/arm.arm/usr/src/sys/OPENBLOCKS_A6/kernel.bin /armroot
```

- ▶ /armroot にユーザランド一式
 - ▶ installworld と distribution を使う
- ▶ /armroot/kernel.bin に、カーネル（生バイナリ）
 - ▶ 手動でコピー

/armroot の内容を USB メモリに入れる

小型機器でやってみよう

▶ USB メモリにコピー(/dev/da2の場合)

```
# cd /armroot
# dd if=/dev/zero of=/dev/da2 bs=512 count=10

# gpart create -s mbr da2
# gpart add -s 32m -t \!14 da2
# newfs_msdos -F 16 /dev/da2s1
# mount -t msdosfs /dev/da2s1 /mnt
# cp /armroot/kernel.bin /mnt
# umount /mnt

# gpart add -t freebsd da2
# newfs -U /dev/da2s2
# mount /dev/da2s2 /mnt
# tar cf - . | tar xf - -C /mnt
# umount /mnt
```

- ▶ FAT 領域に kernel.bin を置く
- ▶ UFS 領域にその他のユーザランドを置く
- ▶ ユーザランドはコピー前に調整しよう

小型機器でやってみよう

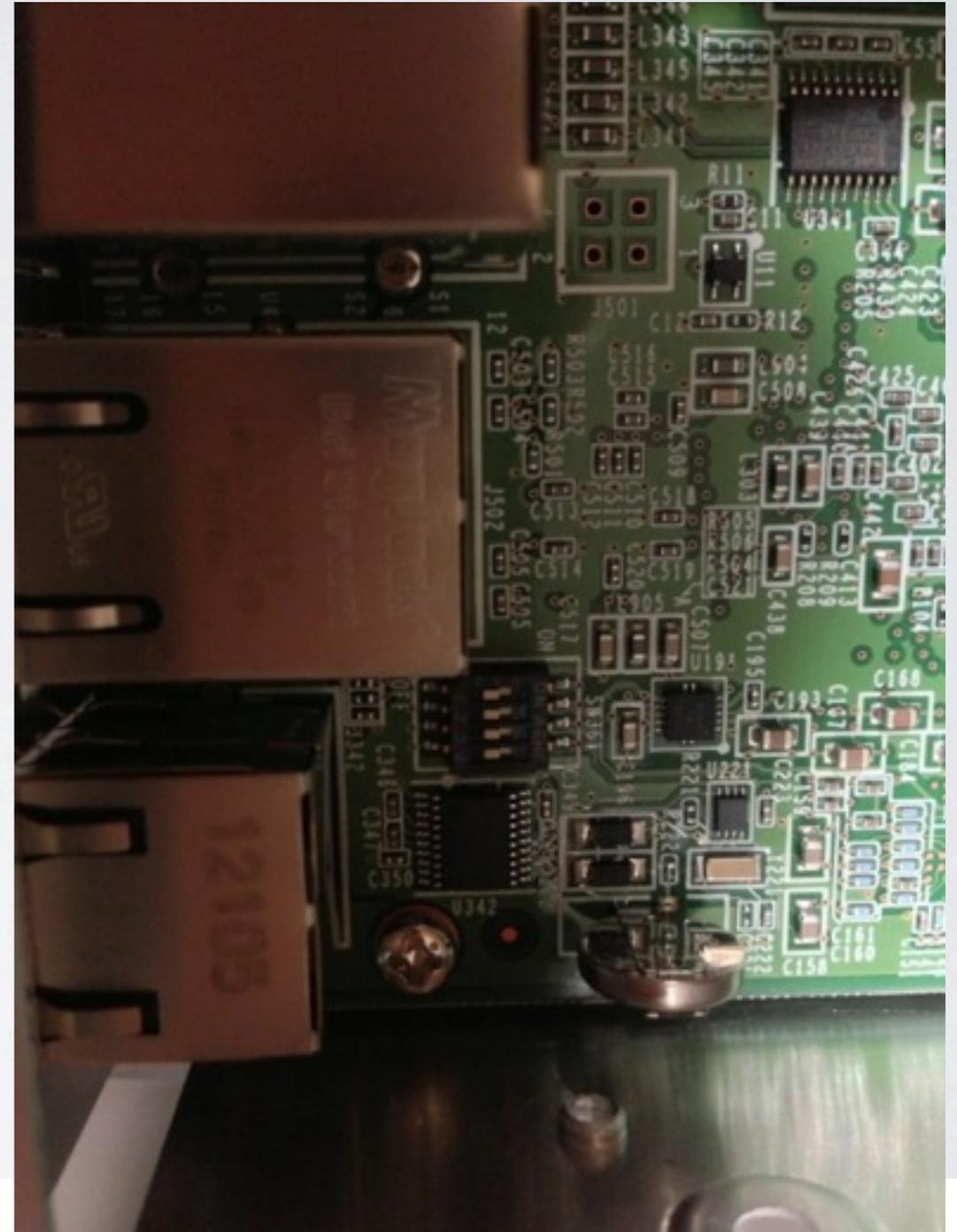
- ▶ **起動だ!**
 - ▶ シリアルコンソールを接続
 - ▶ /etc/remote に
115200bpsの定義を追加
 - ▶ 本体の DIP SW 1 を ON に



```
ucom1x:dv=/dev/cuaU0:br#115200:pa=none:
```

小型機器でやってみよう

- ▶ **起動だ!**
 - ▶ シリアルコンソールを接続
 - ▶ /etc/remote に
115200bpsの定義を追加
 - ▶ 本体の DIP SW 1 を ON に



```
ucom1x:dv=/dev/cuaU0:br#115200:pa=none:
```

小型機器でやってみよう

▶ 起動だ!

```
# tip ucom1x
U-Boot 1.1.4 (Jul  3 2012 - 15:39:15)
Plat'Home version: 1.0.0 (Base: Marvell version: 3.5.9)

U-Boot code: 00600000 -> 0067FFF0  BSS: -> 006CF140

Soc: 88F6282 A1CPU running @ 600Mhz L2 running @ 300Mhz
SysClock = 300Mhz , TClock = 200Mhz

DRAM (DDR2) CAS Latency = 5 tRP = 5 tRAS = 14 tRCD=5
DRAM CS[0] base 0x00000000  size 512MB
DRAM Total size 512MB  16bit width

.....

CPU : Marvell Feroceon (Rev 1)

.....

Run Boot Selection Script

*** SW 1 IS ON ***

*** SW 2 IS OFF, ENTER COMMAND PROMPT ***

openblocks>>
```

小型機器でやってみよう

▶ 起動だ!

- ▶ これで U-boot プロンプトで操作できるようになる
- ▶ CLI で操作できる BIOS のようなものだと思って良い
- ▶ ここで、USB メモリからカーネルを読み込む
コマンドを入力する

小型機器でやってみよう

▶ 起動だ!

```
openblocks>> go 0x00900000
## Starting application at 0x00900000 ...
KDB: debugger backends: ddb
KDB: current backend: ddb
Copyright (c) 1992-2012 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
    The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 10.0-CURRENT #164 r239381:239508M: Wed Aug 22 04:00:26 JST 2012

....
```

- ▶ U-boot のコマンドは、setenv bootcmd で保存できる
- ▶ setenv bootcmd “usb start; fatload usb 0”

小型機器でやってみよう

- ▶ **OpenBlockS A6 のスペック**
 - ▶ 88F6282 600MHz 動作
 - ▶ 512MB RAM
 - ▶ GbE x 1
 - ▶ SATA II インタフェース
 - ▶ USB x 1
 - ▶ GPIO (8-bit)
 - ▶ RS232C (RJ-45) x 1
 - ▶ 消費電力 6W

小型機器でやってみよう

- ▶ **DreamPlug (ARM)**
 - ▶ ほぼ OpenBlockS A6 と同じ
 - ▶ SD カードスロット、GbE x 2, eSATA, USB Audio、無線LANチップが載っている
 - ▶ SD カードからの起動にチャレンジ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ USBで接続できるUARTアダプタが添付されているので、それを使う。



小型機器でやってみよう

```
# tip ucom1x
U-Boot 2011.06 (Oct 15 2011 - 02:02:08)
Marvell-DreamPlug

SoC:   Kirkwood 88F6281_A0
DRAM:  512 MiB
SF:    Detected MX25L1606 with page size 256, total 1 MiB
In:    serial
Out:   serial
Err:   serial
Net:   egiga0, egiga1
88E1121 Initialized on egiga0
88E1121 Initialized on egiga1
Hit any key to stop autoboot:  0
```

- ▶ Hit any key で素早くキーを押して止める
- ▶ SD カードスロットは USB で接続されているので、手順は OpenBlockS A6 と同じ

小型機器でやってみよう

```
>> usb start
(Re)start USB...
USB:   Register 10011 NbrPorts 1
USB EHCI 1.00
scanning bus for devices... 6 USB Device(s) found
      scanning bus for storage devices... 2 Storage Device(s) found
>> fatload usb 1 0x00900000 kernel.bin
reading kernel.bin

5791928 bytes read
## Starting application at 0x00900000 ...

>> go 0x00900000
KDB: debugger backends: ddb
KDB: current backend: ddb
Copyright (c) 1992-2012 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
      The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 10.0-CURRENT #19 r239213:239381M: Sun Aug 19 21:07:10 JST 2012
      hrs@deva.allbsd.org:/usr/obj/arm.arm/usr/src.head/sys/DREAMPLUG arm
```

小型機器でやってみよう

▶ DreamPlug のスペック

- ▶ 88F6281 1.2GHz 動作
- ▶ 512MB RAM
- ▶ GbE x 2
- ▶ eSATA インタフェース
- ▶ SDカードスロット
- ▶ USB x 2
- ▶ Bluetooth, 無線LAN (残念ながらドライバはなし)
- ▶ 消費電力 15W

小型機器でやってみよう

- ▶ **TP-Link TL-WR1043 ND (MIPS)**
 - ▶ ルータ機能付き無線LANアクセスポイント（中国製）
 - ▶ 海外ではポピュラーですが、国内では売っていません
 - ▶ 802.11n 対応、USB端子にストレージ等を接続できる
今時の典型的な構成
 - ▶ 実は Linux が入ってます。
けしからん。
 - ▶ FreeBSD に置き換えてしまえ！



小型機器でやってみよう

- ▶ TP-Link TL-WR1043 ND (MIPS)
 - ▶ 中身は？
 - ▶ Atheros AR9132 (MIPS32) 600MHz 動作
 - ▶ 32MB RAM
 - ▶ GbE x 2, USB x 1
 - ▶ U-Boot を使っている



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



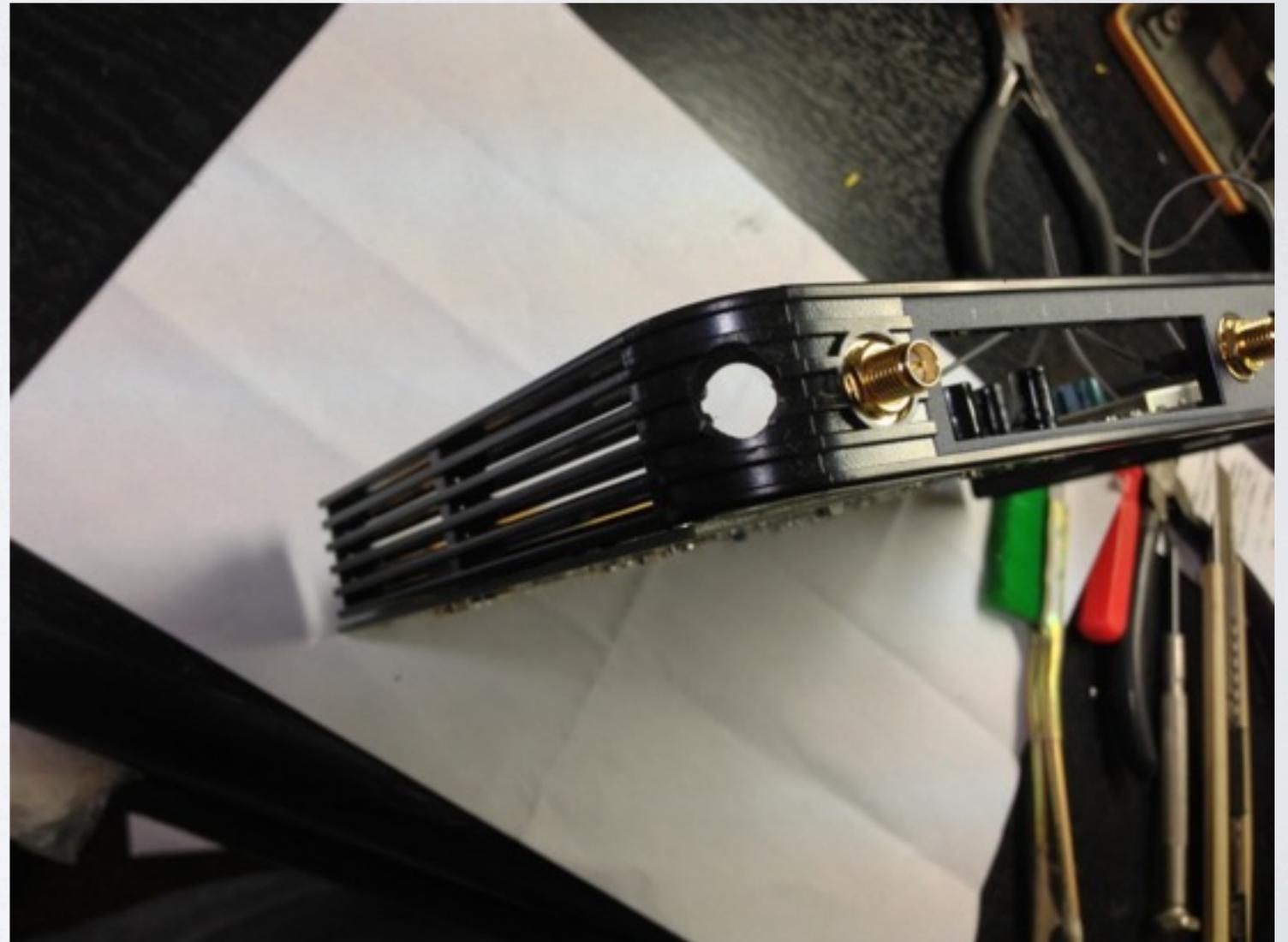
小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



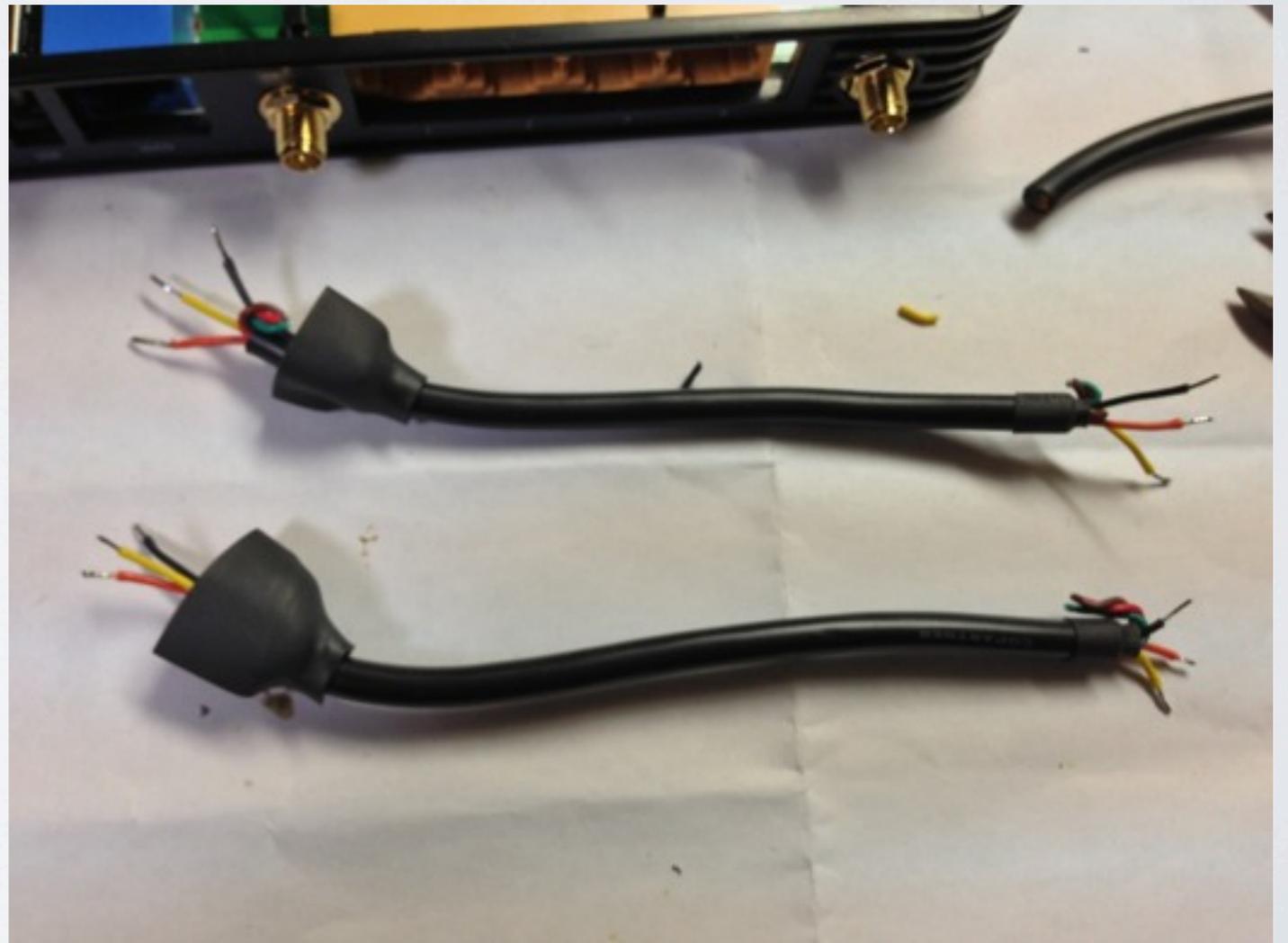
小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ そもそもそんな端子は付いてない
 - ▶ 付けてしまえ



小型機器でやってみよう

- ▶ シリアルコンソール
 - ▶ コネクタは3.5mmステレオ端子を流用
 - ▶ 3.3V系のUSB-Serial 変換ケーブルが使える



小型機器でやってみよう

```
# tip ucom1x
U-Boot 1.1.4 (Apr 28 2011 - 15:55:09)

AP83 (ar9100) U-boot 0.0.11
DRAM:
sri
32 MB
id read 0x100000ff
flash size 8MB, sector count = 128
Flash: 8 MB
Using default environment

In:      serial
Out:     serial
Err:     serial
Net:     ag7100_enet_initialize...
No valid address in Flash. Using fixed address
: cfg1 0xf cfg2 0x7114
eth0: 00:03:7f:09:0b:ad
eth0 up
eth0
Autobooting in 1 seconds
```

▶ やったぜ！

小型機器でやってみよう

```
# tip ucom1x
U-Boot 1.1.4 (Apr 28 2011 - 15:55:09)

AP83 (ar9100) U-boot 0.0.11
DRAM:
sri
32 MB
id read 0x100000ff
flash size 8MB, sector count = 128
Flash: 8 MB
Using default environment

In:      serial
Out:     serial
Err:     serial
Net:     ag7100_enet_initialize...
No valid address in Flash. Using fixed address
: cfg1 0xf cfg2 0x7114
eth0: 00:03:7f:09:0b:ad
eth0 up
eth0
Autobooting in 1 seconds
```

- ▶ Autobooting のところで素早く “t!p<CR>” と入れる！

小型機器でやってみよう

- ▶ システムはどうやってつくるの？
 - ▶ ちょっと大変
 - ▶ freebsd-wifi-build というスクリプトにまとめてある

```
% svn checkout svn://svn.freebsd.org/base/head src
% svn checkout http://freebsd-wifi-build.googlecode.com/svn/trunk/ build
% cd build/programs/mktplinkfw
% make
# make install
# ../../build/bin/build tl-wr1043nd buildworld buildkernel
# mkdir /tftpboot
# ../../build/bin/build tl-wr1043nd installworld installkernel
distribution mfsroot fsimage tplink
```

小型機器でやってみよう

▶ U-boot の操作

- ▶ 記憶装置がないので、フラッシュメモリに書き込む

```
>> setenv ipaddr 192.168.0.22
>> setenv serverip 192.168.0.10
>> erase 0xbf020000 +7c0000
>> tftpboot 0x81000000 TP-WN1043ND.factory.bin
>> cp.b 0x81000000 0xbf020000 0x7c0000
>> bootm 0xbf020000
```

- ▶ tftp でファイルシステムイメージを転送するので準備

```
# /etc/inetd.conf
tftp    dgram    udp        wait root /usr/libexec/tftpd tftpd -l -s /tftpboot
```

```
# /etc/hosts.allow
tftpd : 192.168.0.0/255.255.0.0 : allow
```

```
# /etc/rc.conf
inetd_enable="YES"
```

小型機器でやってみよう

```
# tip ucom1x
KDB: debugger backends: ddb
KDB: current backend: ddb
Copyright (c) 1992-2012 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
    The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 10.0-CURRENT #0 r239381:239508M: Thu Jan  1 09:00:00 JST 1970
    hrs@deva.allbsd.org:/usr/obj/mipseb/mips.mips/usr/src.head.mips/sys/TP-
WN1043ND mips

...

FreeBSD/mips (freebsd-wifi-build) (ttyu0)

login:
```

▶ わーい

小型機器でやってみよう

```
# ifconfig
arge0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> metric 0 mtu
1500
    ether 90:f6:52:3e:cd:fc
    inet6 fe80::92f6:52ff:fe3e:cdfc%arge0 prefixlen 64 scopeid 0x6
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    media: Ethernet 1000baseT <full-duplex>
    status: active
arge1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 90:f6:52:3e:cd:fd
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    media: Ethernet 100baseTX <full-duplex>
    status: active
ath0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 2290
    ether 00:19:e0:66:66:68
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    media: IEEE 802.11 Wireless Ethernet autoselect (autoselect)
    status: no carrier
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 9e:ed:03:3a:39:df
    inet 192.168.1.20 netmask 0xfffff00 broadcast 192.168.1.255
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
    maxage 20 holdcnt 6 proto rstp maxaddr 2000 timeout 1200
    root id 00:00:00:00:00:00 priority 32768 ifcost 0 port 0
    member: arge0 flags=143<LEARNING,DISCOVER,AUTOEDGE,AUTOPTP>
        ifmaxaddr 0 port 6 priority 128 path cost 2000000
```

小型機器でやってみよう

- ▶ **環境は構築スクリプトでカスタマイズ**
 - ▶ 容量が制限されているので、すべてのシステムは入っていない
 - ▶ `freebsd-wifi-build` のスクリプトは単純なので、入れたたいコマンドや `/etc/rc.conf` を変えるのは容易
 - ▶ `hostapd` を動かすと、802.11n 対応 AP としてちゃんと動作します
 - ▶ 注意：技適証明のない海外の無線LAN製品を国内で使うと電波法違反になる場合があります。

まとめ

- ▶ FreeBSD は PC 以外でもそこそこ動きます。
商用製品も存在します。
- ▶ 組み込みは、起動の部分が一番難しい。
それ以外はPCで使う場合と大差はありません。

まとめ

- ▶ FreeBSD は PC 以外でもそこそこ動きます。
商用製品も存在します。
- ▶ 組み込みは、起動の部分が一番難しい。
それ以外はPCで使う場合と大差はありません。
- ▶ ARM や MIPS の FreeBSD は、クロスコンパイルで構築できます。フレームワークは NetBSD と比べるとだいぶ遅れていますが、これから拡充予定です。
- ▶ 実際に動く小型機器の入手はそれほど難しくありません。
(FreeBSD を入れる難易度はさまざまですが...)
PCで設定する知識があるなら、それを応用して家庭用のルータや単機能サーバを自分で構築できます。

次回は

- ▶ 前々回、あまりにやっつけ仕事だった ZFS の話を、ちゃんとやります

お知らせ 1 : ABC2013

- ▶ AsiaBSDCon 2013 を、3/14-17 で開催します。
- ▶ 場所は前回と同じ、東京理科大学（JR飯田橋駅）です
- ▶ 基本的に英語での開催ですが、日本語での催しも企画しています。詳しくは <http://www.asiabsdcon.org> まで。

お知らせ 2 : ML

- ▶ FreeBSDに関する疑問や、勉強会の内容のフォローアップ等のやりとりができるメーリングリストをつくりました。
- ▶ もし興味のある方がいらっしゃったら、参加してみてください。
- ▶ 今回の組み込み機器に関するアップデートも、そのへんに投げます。
- ▶ <http://lists.allbsd.org> にアクセスして、freebsd-ja というメーリングリストに入会申請を出してください。手順に従って操作すれば、自動登録されます。

おしまい

- ▶ 質問はありますか？