NetBSD/sandpoint を KURO-BOX/HG（玄箱 /HG）を使って、実際に試してみた。そのめも。

はじめにのはじめに

NetBSD 7 で試す場合の注意

この文書は、NetBSD 6 で試したものなので、NetBSD 7 で試す場合は、以下の注意があります。

手元で試している限りでは、NetBSD 7.0 Release の altboot.bin ではうまく boot できなかったり、boot にとても時間がかかるようです。NetBSD 6 のものだとうまく動いています。NetBSD 6.1.5 の altboot.bin は、http://ftp.NetBSD.org/pub/NetBSD/NetBSD-6.1.5/sandpoint/installation/ にあります。

NetBSD 7 で試す場合の補足説明

NetBSD 7 にも disklabel コマンドの -B オプションは入らなかったので、以下の、NetBSD current の disklabel コマンドを用意する方法は（残念ながら）有効です。

はじめに

前史

これまで、LinkStation にて、「玄箱で NetBSD を動かす」版（かわうち版）の NetBSD/sandpoint を使ってきて、特に困ったことはなく使えていた。

さて、某所で、KURO-BOX/HG が、disk 付で格安で売られていて、うっかり買ってしまった。買ったはいいものの、どうしようか考えていたら、めったと流れない port-sandpoint ML で、"Announcing KuroBox support" というメールを見てしまった。ついに NetBSD 本家でサポートされたのかー。

そして、そのメールを見たその日に、自宅に KURO-BOX/HG が届いた。

これはやるしかないよね。

しばらくして、"NAS support ready for upcoming releases" というメールが ML で流れた。そして、インストールの方法のページが公開された。

しかし、時間があまりつくれなくて、ほったらかしにしていたら、"All supported NAS models for 6.0 " というアナウンスが出たり、dmesg が出たりした。

さぁやるか

2012 年 10 月 18 日に NetBSD 6.0 が出た。6.0 が出たのであれば、やらざるをえない。（OSC 2012 Tokyo/Fall に間に合うかな、と思っていたら、間に合わなかった。）

改めて、インストールの方法のページを見ると、シリアルコンソールを出したり、u-boot 化したり、かわうち版と比べると、ハードルが高い。

結論として、シリアルコンソール化もせず、u-boot 化もしないでインストールする一番簡単な方法は、"NetBSD/sandpoint 6.0 on kurobox" 等に書かれているとおり、かわうち版でも使っていた、「ブートローダー (v3)」と「なんちゃってブートセレクタ（タイプ２）」を流用する方法が、一番簡単と

思われる。（たぶん、大半の国内ユーザはそうしていると思われる。）

しかし、同じことしてみてもしょうがない (?) ので、ここでは、本家のインストール方法に近い形で行うこととする。

方針

本家のインストールの方法のページのとおり、u-boot 化する。

Linux 環境を残しておけば、シリアルコンソールは使わないようにできそうなので、そうすることにする。

注意

一般

機種によっては、本家のインストール解説にある情報ではうまくいかないかもしれない。適宜、各種玄箱 hack の情報、特に、玄箱そのものの知識（EM モードの変更方法や通常モードへの戻り方、デフォルトの IP アドレス等）、そして、U-Boot 化と、他ディストリビューション化（Debian 化とか）の情報を調べて、予備知識をつけよう。

たとえば、root のパスワードは、KURO-BOX と KURO-BOX/HG とでは違うし、LinkStation は、HDD はそのままでは mount できない。

Genbako kernel collection さんのサイト移転

Genbako kernel collection さんのサイトが移転している。このページを書き始めた 2012 年には http://www.genbako.com/ だったが、現在は http://genbako.vodapone.com/ のようだ。

世のたくさんの Web ページ（NetBSD 本家のリンクも含む）は古いほうを向いているので、ちうい。

以下の文中のリンクは、全て新しい方に向けてある。

U-Boot 化 & Debian 化

Debian 化

Genbako kernel collection さんの install-new-kernel.sjis.txt に書かれているとおりにやる。（「HG の場合はこれも必要」のところまで。）

ようは、玄箱うぉううぉう♪さんの、install_debian_standalone.txt にしたがって、Debian 化し、そのあと、install-new-kernel.sjis.txt に書かれている不具合修正を行う、ということである。

（今にして思えば、http://www.genbako.com/debian-2.6.17.3/ を最初から使ってもよかったと思うが、もともと知られた手順どおりに行った。2013 年 10 月に確認した限りでは、玄箱うぉううぉう♪さんは既に消失しており、Genbako kernel collection さんの配布物を使う以外にない。）

（2017/7/20 注：玄箱うぉううぉう♪さんのコンテンツは復活されており、install-new-kernel.sjis.txt

の不具合を修正した（ぽい）もの含めて、配布を再開されている。https://dsk.jp/kurobox/）

sarge(3.1) 化

■ /etc/apt/sources.list の変更と upgrade

上記までの状態では、woody(3.0) なので、sarge 化する。基本的には、install-new-kernel.sjis.txt に書かれているとおりにやる。（「※カーネル変更」の前まで。）

/etc/apt/sources.list の変更については、install-new-kernel.sjis.txt のとおりに書いても動かない。sarge はずいぶん昔にサポートが打ちきられており、アーカイブサイトにしかない。よって、以下のとおりとする。

```
deb http://archive.debian.org/debian/ sarge main contrib
deb http://archive.debian.org/debian/ sarge-proposed-updates main contrib
```

■ /etc/init.d/halt の修正

/etc/init.d/halt の修正は、PATH 行の直後に書く。

■ /etc/kuroevtd/resetpress の修正

/dev/fl* は sarge 化で使えなくなるので、変更しておく。（ただ、試した限りでは、こうやっても、うまく動かない。）

```
#! /bin/sh
/usr/sbin/write_ng > /dev/mtdblock2
/sbin/shutdown -h now
```

■ /etc/fstab の修正

/dev/hda3 は NetBSD に割り当てるので、該当行を消す。（または、コメントアウトする。）

カーネル変更

kernelimage は、kernelimage-2.6.25.1-kuroHG.tgz を使った。install-new-kernel.sjis.txt のとおり、/boot に展開する。

リブート

これも、install-new-kernel.sjis.txt に書いてあるとおりにする。2.6.15 ではないけれども、sarge 化の際は必要みたい。もちろん、modules-2.6.25.1-kuroHG.tgz を使う。

ぼくの環境では、shutdown -r now や /etc/init.d/reboot すると、次の boot は、一切 telnetd（や別途インストールした sshd）が反応しないという症状になった。（なぜか、ping には反応するが、それ以外は一切ダメ。）困り果てたが、強制的に電源 off（= 電源ケーブルを引っこ抜く）して、再度起動させたら、telnetd や sshd は反応するようになった。

この現象は毎回おきる。つまり、一度電源を落としたり、再起動したりすると、次に Linux を起動するときは、ネットワークの応答がなくなり、強制的に電源 off して、再度起動すると、復活する。

Flash ROM のデバイスファイル作成

これもやっておく。sarge 化によって、/dev/fl* が使えなくなるのだ。install-new-kernel.sjis.txt の「参

考　FLASH を使おう」のとおりである。

U-Boot 化

Genbako kernel collection さんに引き続きお世話になる。

■ U-Boot 化された kernelimage

http://www.genbako.com/uImage/ からダウンロードしようとすると思うが、2.6.25.1 はない。実は、kernelimage-2.6.25.1-kuroHG.tgz の中にすでに入っていたのだ。よって、もうインストールは終わっているはず。

```
# ls -l /boot/uImage
-rw-r--r--  1 root root 1422555 May  4  2008 /boot/uImage
```

よって、別途ダウンロードする必要はない。ただし、シンボリックリンクをつくらないと、U-Boot で boot しないので、それだけはつくる必要がある。

```
# cd /boot
# ln -s uImage vmlinux.UBoot
```

■ U-Boot の書き込み

http://www.genbako.com/u-boot_loader/ から、U-Boot 化されたファームウェアをダウンロードしてくる。注意書きをちゃんと読むこと。

適当なところにダウンロードできたら、md5sum で MD5 の値をとって、ダウンロード時に壊れなかったかどうかを確認する。.md5 のファイルも一緒にダウンロードして、同じ dir に保存しておいて、

```
# md5sum -c u-boot-hg.flash.md5
```

とするのが簡単。

次に、いよいよ書き込み。（これを失敗すると、でかい文鎮ができる。）

まず、EM モードで起動する。select_mode.txt のとおり、でいいはず。

telnet はとても遅いので、プロンプトが帰ってこなくても、待つこと。（POWER が点滅していなければ、EM モードか、通常モードか、どっちかで上がっているはず。）

EM モードで上がっていたら、login して、u-boot-hg.flash.bin を保存してある内蔵 HDD を mount。そのあと、注意書きどおり、

```
# cat u-boot-hg.flash.bin > /dev/fl2
```

しましょう。（dd を使うと死にます。たぶん。）

これで、readme.txt のとおり、netcat(= nc) で待ち受けておけば、u-boot のコンソールがリモートからとれるようになったはず。（ちなみに、NetBSD の場合、netcat は pkgsrc の net/netcat にあるので、

インストールしておく。Linux の各ディストリビューションの場合は、オリジナルの netcat（= OpenBSD 版 netcat ではない）をインストールしないと、readme.txt どおりのコマンドラインオプションでは動かない。）

KURO-BOX/HG の電源をいったん落とす。

```
# /usr/bin/write_ok > /dev/fl3 （エラーが出ますが、きにしない）
# shutdown -h now
```

再度起動して、待ち受けていた netcat に u-boot のコンソールが出ているかどうかを確認する。問題なければ、再度電源を落とす。

NetBSD 化

準備

■はじめに

KURO-BOX/HG をバラして、ディスクを取り出して、中身が読み書きできる環境として、NetBSD/i386 か NetBSD/amd64 の環境があることを前提とする。

ここでは、NetBSD/i386 をインストールした Netbook（MSI Wind Notebook U100）を用意した。（以下作業 PC と呼ぶ。）

仮想環境（Virtual Box 等）でも、できると思うけど、今回は実機があるので、ここでは考えない。

■バイトオーダーの問題

対象とする KURO-BOX/HG は PPC 系の CPU なので、ビックエンディアンなのだそうだ。対して、i386 はリトルエンディアンである。具体的に何が問題になるのかというと、

- ディスクラベル (disklabel)
  - *BSD の世界で使われるパーティション情報。PC や Linux の世界で使われる MBR パーティションとは別にあるもの。
- ファイルシステム

がある。つまり、NetBSD/i386 上でディスクラベルを書いたりファイルシステムを作ったりして作業すると、KURO-BOX/HG 上の NetBSD/sandpoint（やブートローダである altboot）では理解できなくなってしまう。

ファイルシステムの作成（フォーマット）そのものは、newfs(8) には、-B (byte-order) オプションがあり、問題なさそうだ。しかし、それで作成したパーティションは、NetBSD/i386 では mount できないので、カーネルにそれをマウントできるオプションを追加する必要がある。

また、ディスクラベルの方は、NetBSD/i386 -current の disklabel コマンドには、-B (byte-order) オプションがあるため、それを用意することにする。

■手順の確認

以下のとおり。

1. NetBSD/i386 のカスタムカーネルを作成し、それを使って作業 PC を (再) 起動する。
2. NetBSD/i386 -current の disklabel コマンドを用意する。
3. 作業 PC 上で KURO-BOX/HG のディスクの MBR パーティションをいじって、NetBSD の MBR パーティションを作成する。
4. 作業 PC 上で KURO-BOX/HG のディスクにディスクラベルを書く。
5. 作業 PC 上で KURO-BOX/HG のディスク（の NetBSD でつかうところ）をフォーマットする。
    - -B (byte-order) オプションを使って、ビックエンディアンにする。
6. 作業 PC 上で KURO-BOX/HG のディスク（の NetBSD でつかうところ）をマウントし、NetBSD/sandpoint をインストールする。
7. 作業 PC 上で KURO-BOX/HG のディスク（の NetBSD でつかうところ）をマウントし、NetBSD/sandpoint の初期設定をする。
8. 作業 PC 上で KURO-BOX/HG のディスクのディスクラベルが NetBSD/sandpoint で扱えるように、NetBSD/i386 -current の disklabel で書き直す。
    - -B (byte-order) オプションを使って、ビックエンディアンにする。
9. KURO-BOX/HG にディスクを取り付け、Linux 側をいったん起動し、altboot を Linux 側の /boot/altboot.bin に置く。
10. NetBSD が起動できるか試す。問題なければ、常時起動するようにする。

NetBSD/i386 のカスタムカーネルを作成し、それを使って作業 PC を (再) 起動する

NetBSD 6 系のソースコードが手元にある人はそれを使ってもよい。

ソースコードが手元に無い人で、あなたが、6 系をお使いなら、自分が使っているもののソースコードをとってこればよい。6.0 なら、

http://ftp.jaist.ac.jp/pub/NetBSD/NetBSD-6.0/source/sets/

などから、とってくる。xsrc.tar.gz 以外のものをとりあえずとってきて、展開しよう。デフォルトでは、/ から展開すると、/usr/src に展開される。

展開できたら、カスタムカーネルを作る。まず、コンフィグファイルを作成する。/usr/src/sys/i386/conf に移動して、GENERIC をコピーして、編集する。

```
# cd /usr/src/sys/arch/i386/conf
# cp -p GENERIC FFS_EI
```

FFS_EI（という名前でなくてももちろんよい）を以下のように編集する。

```
--- GENERIC  2012-10-05 00:35:16.000000000 +0900
+++ FFS_EI  2013-10-15 01:52:41.000000000 +0900
@@ -184,7 +184,7 @@
 # File system options
 options   QUOTA    # legacy UFS quotas
 options   QUOTA2    # new, in-filesystem UFS quotas
-#options   FFS_EI    # FFS Endian Independent support
+options   FFS_EI    # FFS Endian Independent support
 options   WAPBL    # File system journaling support
 #options   UFS_DIRHASH  # UFS Large Directory Hashing - Experimental
 options   NFSSERVER  # Network File System server
```

ようは、FFS_EI の行の先頭の # を取るだけ。

編集できたら、コンパイルする。

```
# cd /usr/src
# ./build.sh -m i386 tools
# ./build.sh -m i386 kernel=FFS_EI
```

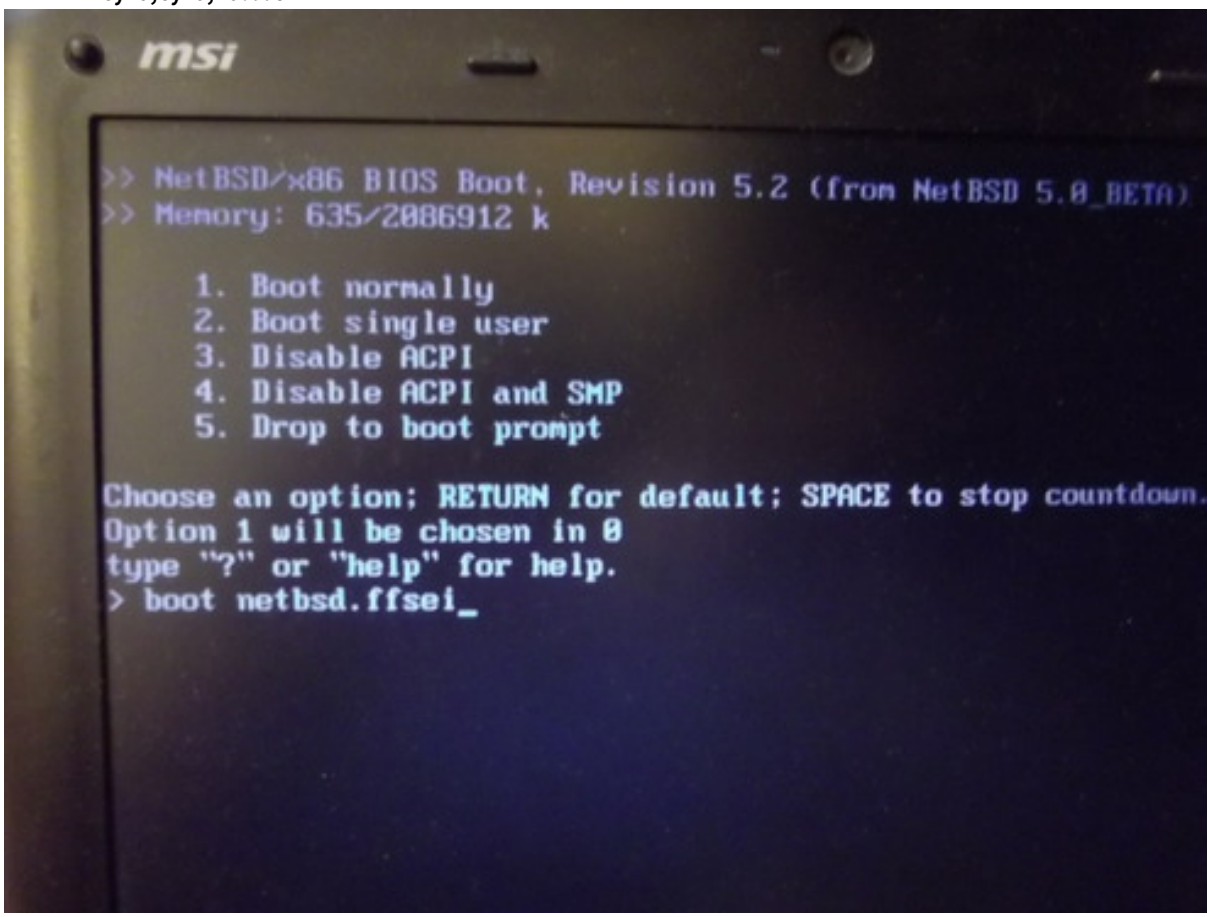
kernel= の後はコンフィグファイル名。

しばし待つ（非力な MSI Wind Notebook U100 だと、1 時間ちょっとくらい）と、コンパイルが終わる。

```
===> Kernels built from FFS_EI:
  /usr/src/sys/arch/i386/compile/obj.i386/FFS_EI/netbsd
```

と、できたカーネルの場所が出ているはずなので、それをつかって、再起動する。

```
# cp /usr/src/sys/arch/i386/compile/obj.i386/FFS_EI/netbsd /netbsd.ffsei
# sync;sync;reboot
```



起動の途中で、"Drop boot prompt"（この例では、5）を入力し、そのあとのプロンプトで、"boot netbsd.ffsei" と入力するだけである。

万一、うまく起動できなかったりしたときは、いったん電源を停止して、再度起動すれば、いままでどおりのカーネルで起動してくるから大丈夫。

NetBSD/i386 -current の disklabel コマンドを用意する

残念ながら、現行（NetBSD 6.1）の disklabel コマンドは、バイトオーダーを変えることはできないが、2013 年 5 月以降の NetBSD -current（開発先端）では、newfs コマンドと同じ、-B (byte-order) オプションが実装されている。これを使うことにする。

まず、cvs で最新のソースコードをとってくる。

```
# cd /var/tmp
# cvs -d anoncvs@anoncvs.netbsd.org:/cvsroot co src/sbin/disklabel
```

とってこれたら、これをコンパイルする。

```
# cd src/sbin/disklabel
# make USETOOLS=no cleandir dependall
```

コンパイルが通れば、無事、disklabel コマンドができているはず。

作業 PC 上で KURO-BOX/HG のディスクの MBR パーティションをいじって、NetBSD の MBR パーティションを作成する

KURO-BOX/HG をバラして、ディスクをとりだし、作業用 PC に PATA-USB 変換経由で接続。sd1 で見えた。

fdisk を使って、一番後ろにあるデータ領域用の Linux パーティション (Linux 用語では hda3) を NetBSD の MBR パーティションとする。要点は以下の通り、

1. fdisk -u sd1 で起動。
2. "Do you want to change our idea of what BIOS thinks? [n]" はなにも入力しない（単に Enter）
3. "Which partition do you want to change?: [none]" は、2 と入力
   - NetBSD の fdisk では、先頭から、0, 1, 2, 3。
4. sysid は、169 を入力
5. start と size と bootmenu はそのまま（単に Enter）
6. "The bootselect code is not installed, do you want to install it now? [n]" もなにも入力しない（単に Enter）
7. 入力が戻ってくるので、2 番が NetBSD になっていることを確認。"Which partition do you want to change?: [none]" にはなにも入力しない（単に Enter）
8. "Update the bootcode from /usr/mdec/mbr? [n]" もなにも入力しない（単に Enter）
9. "We haven't written the MBR back to disk yet. This is your last chance." とでて、もう一度、MBR パーティションの情報が出て、"Should we write new partition table? [n]" には、y と入力する。
   - もし、ここで、止めておきたければ、n を入力すれば、何も変更されない。（よって、何度でも試すことができる。）

結果、こんな感じ。

```
% fdisk sd1
Disk: /dev/rsd1d
NetBSD disklabel disk geometry:
cylinders: 238475, heads: 64, sectors/track: 32 (2048 sectors/cylinder)
total sectors: 488397168

BIOS disk geometry:
cylinders: 1024, heads: 255, sectors/track: 63 (16065 sectors/cylinder)
total sectors: 488397168
```

```
Partition table:
0: Linux native (sysid 131)
    start 63, size 4192902 (2047 MB, Cyls 0-260)
        PBR is not bootable: All bytes are identical (0x00)
1: Linux swap or Prime or Solaris (sysid 130)
    start 4192965, size 514080 (251 MB, Cyls 261-292)
        PBR is not bootable: All bytes are identical (0x00)
2: NetBSD (sysid 169)
    start 4707045, size 483690123 (236177 MB, Cyls 293-30401/80/63)
        PBR is not bootable: All bytes are identical (0x00)
3: <UNUSED>
No active partition.
Drive serial number: 2023426387 (0x789b0953)
```

作業 PC 上で KURO-BOX/HG のディスクにディスクラベルを書く

次に、disklabel を書く。下記のようにすることにした。

- a -> NetBSD（システム領域）
- b -> swap（Linux の swap と兼用 , MBR パーティションの 1）
- c （使えない）
- d （使わない）
- e -> Linux（システム領域 , MBR パーティションの 0）
- f -> NetBSD（データ領域）

つまり、

- MBR パーティションの 0 -> e
- MBR パーティションの 1 -> b
- MBR パーティションの 2 -> a と f

にする、ということである。

これにより、disk の先頭から、e -> b -> a -> f の順番である。

b パーティションは、MBR パーティションの 1 の先頭 8 セクタだけ使わないようにする。（当然、サイズも 8 小さくなる。）これは、Linux が swap と認識するためのデータがかかれている部分を消さないようにするため。

MBR パーティションの 2 は、a を 1GB として、あとは f にする。

disklabel の設定は、disklabel コマンドを使う。

```
# disklabel -i -I sd1
Enter '?' for help
partition> ?
?       print this menu
C       make partitions contiguous
E       print disk label and current partition table
I       change label information
L       list all known file system types
N       name the label
P       print current partition table
Q       quit
R       rounding (c)ylinders (s)ectors
W       write the current partition table
[a-p]   define named partition
```

最初に、OS が適当にみつくろってくれる disklabel を見ておく。

```
partition> P
7 partitions:
#        size    offset     fstype [fsize bsize cpg/sgs]
 c: 483690123   4707045     unused      0     0        # (Cyl.   2298*- 238475*)
 d: 488397168         0     unused      0     0        # (Cyl.      0 - 238475*)
 e:   4192902        63 Linux Ext2      0     0        # (Cyl.      0*-   2047*)
 f:    514080   4192965       swap                    # (Cyl.   2047*-   2298*)
 g: 483690123   4707045     4.2BSD      0     0     0  # (Cyl.   2298*- 238475*)
```

- MBR パーティションの 0 が、e になっているようだ。
- MBR パーティションの 1 が、f になっているようだ。
- MBR パーティションの 2 が、g になっているようだ。

では、変更していく。まずは、g と f を消す。サイズに 0 を入力すれば、消える。

```
partition> g
Filesystem type [?] [4.2BSD]:
Start offset ('x' to start after partition 'x') [2298.36181640625c, 4707045s, 2298.36181640625M]: 0
Partition size ('$' for all remaining) [236176.81787109375c, 483690123s, 236176.81787109375M]: 0
partition> f
Filesystem type [?] [swap]:
Start offset ('x' to start after partition 'x') [2047.34619140625c, 4192965s, 2047.34619140625M]: 0
Partition size ('$' for all remaining) [251.015625c, 514080s, 251.015625M]: 0
partition> P
7 partitions:
#        size    offset     fstype [fsize bsize cpg/sgs]
 c: 483690123   4707045     unused      0     0        # (Cyl.   2298*- 238475*)
 d: 488397168         0     unused      0     0        # (Cyl.      0 - 238475*)
 e:   4192902        63 Linux Ext2      0     0        # (Cyl.      0*-   2047*)
```

消せたので、まずは、MBR パーティションの 1 を b にしよう。

```
partition> b
Filesystem type [?] [unused]: ?
Supported file system types:
        4.1BSD          EFS             MINIX FSv3      UDF
        4.2BSD          Eighth Edition  MSDOS           unknown
        4.4LFS          FILECORE        NiLFS           unused
        ADOS            HFS             NTFS            Version 6
        Apple UFS       HPFS            RAID            Version 7
        boot            ISO9660         swap            vinum
        ccd             jfs             System V
        cgd             Linux Ext2      SysVBFS
Filesystem type [?] [unused]: swap
Start offset ('x' to start after partition 'x') [0c, 0s, 0M]: 4192973
Partition size ('$' for all remaining) [0c, 0s, 0M]: 514072
 b:    514072   4192973       swap                    # (Cyl.   2047*-   2298*)
partition> P
7 partitions:
#        size    offset     fstype [fsize bsize cpg/sgs]
 b:    514072   4192973       swap                    # (Cyl.   2047*-   2298*)
 c: 483690123   4707045     unused      0     0        # (Cyl.   2298*- 238475*)
 d: 488397168         0     unused      0     0        # (Cyl.      0 - 238475*)
 e:   4192902        63 Linux Ext2      0     0        # (Cyl.      0*-   2047*)
```

次に、a と f を作る。

```
partition> a
Filesystem type [?] [unused]: 4.2BSD
Start offset ('x' to start after partition 'x') [0c, 0s, 0M]: b
Partition size ('$' for all remaining) [0c, 0s, 0M]: 1024M
 a:   2097152   4707045     4.2BSD      0     0     0  # (Cyl.   2298*-   3322*)
partition> f
Filesystem type [?] [swap]: 4.2BSD
Start offset ('x' to start after partition 'x') [0c, 0s, 0M]: a
Partition size ('$' for all remaining) [0c, 0s, 0M]: $
 f: 481592971   6804197     4.2BSD      0     0     0  # (Cyl.   3322*- 238475*)
partition> P
```

```
7 partitions:
#        size    offset     fstype [fsize bsize cpg/sgs]
 a:   2097152   4707045     4.2BSD      0     0     0  # (Cyl.   2298*-   3322*)
 b:    514072   4192973       swap                     # (Cyl.   2047*-   2298*)
 c: 483690123   4707045     unused      0     0        # (Cyl.   2298*- 238475*)
 d: 488397168         0     unused      0     0        # (Cyl.      0 - 238475*)
 e:   4192902        63 Linux Ext2      0     0        # (Cyl.      0*-   2047*)
 f: 481592971   6804197     4.2BSD      0     0     0  # (Cyl.   3322*- 238475*)
```

問題ないようなので、disklabel を作成しよう。

```
partition> ?
?       print this menu
C       make partitions contiguous
E       print disk label and current partition table
I       change label information
L       list all known file system types
N       name the label
P       print current partition table
Q       quit
R       rounding (c)ylinders (s)ectors
W       write the current partition table
[a-p]   define named partition
partition> W
Label disk [n]? y
Label written
partition> Q
```

（もし不安な時は、）"Label disk [n]" のところで、n と入力すれば、一切変更は反映されない。

こんな感じになった。

```
% disklabel -r sd1
# /dev/rsd1d:
type: SCSI
disk: OOBB-55RDAO
label: fictitious
flags:
bytes/sector: 512
sectors/track: 32
tracks/cylinder: 64
sectors/cylinder: 2048
cylinders: 238475
total sectors: 488397168
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # microseconds
track-to-track seek: 0  # microseconds
drivedata: 0

7 partitions:
#        size    offset     fstype [fsize bsize cpg/sgs]
 a:   2097152   4707045     4.2BSD      0     0     0  # (Cyl.   2298*-   3322*)
 b:    514072   4192973       swap                     # (Cyl.   2047*-   2298*)
 c: 483690123   4707045     unused      0     0        # (Cyl.   2298*- 238475*)
 d: 488397168         0     unused      0     0        # (Cyl.      0 - 238475*)
 e:   4192902        63 Linux Ext2      0     0        # (Cyl.      0*-   2047*)
 f: 481592971   6804197     4.2BSD      0     0     0  # (Cyl.   3322*- 238475*)
```

作業 PC 上で KURO-BOX/HG のディスク（の NetBSD でつかうところ）をフォーマットする

NetBSD パーティションは、フツーに newfs すればよい。-B (byte-order) オプションを忘れないように。

```
# newfs -B be /dev/rsd1a
/dev/rsd1a: 1024.0MB (2097152 sectors) block size 16384, fragment size 2048
        using 6 cylinder groups of 170.67MB, 10923 blks, 21504 inodes.
```

```
  super-block backups (for fsck_ffs -b #) at:
  32, 349568, 699104, 1048640, 1398176, 1747712,
  # newfs -B be -O 2 /dev/rsd1f
  /dev/rsd1f: 235152.8MB (481592968 sectors) block size 32768, fragment size 4096
        using 317 cylinder groups of 741.81MB, 23738 blks, 46848 inodes.
  super-block backups (for fsck_ffs -b #) at:
  192, 1519424, 3038656, 4557888, 6077120, 7596352, 9115584, 10634816, 12154048, 13673280, 15192512,
16711744,

.....................................................................................................
```

-O 2 は、FFSv2(UFS2) にするオプション。

作業 PC 上で KURO-BOX/HG のディスク（の NetBSD でつかうところ）をマウントし、NetBSD/sandpoint をインストールする

■ sets の展開

適当なミラーサイト（例：ftp://ftp.jaist.ac.jp/pub/NetBSD/NetBSD-6.0/sandpoint/binary/sets/）から取得する。x で始まる .tgz と、kern-GENERIC.tgz は必要ない。

とってきたら、壊れていないかどうか確認する。NetBSD 上だと、cksum で確認できる。

```
  % ls
   MD5                    base.tgz              etc.tgz               kern-KUROBOX.tgz    misc.tgz
tests.tgz
   SHA512                comp.tgz             games.tgz             man.tgz               modules.tgz
text.tgz
  % cat MD5 SHA512 | cksum -c
  (MD5) xbase.tgz: FAILED
  (MD5) xcomp.tgz: FAILED
  (MD5) xetc.tgz: FAILED
  (MD5) xfont.tgz: FAILED
  (MD5) xserver.tgz: FAILED
  (SHA512) xbase.tgz: FAILED
  (SHA512) xcomp.tgz: FAILED
  (SHA512) xetc.tgz: FAILED
  (SHA512) xfont.tgz: FAILED
  (SHA512) xserver.tgz: FAILED
```

無いファイルはエラーになるのは問題ない。展開する。

```
  % su -
  # mount /dev/sd1a /mnt
  # cd /path/to
  # for h in *.tgz
  > do
  > echo -n "$h: ";tar xzpehf $h -C /mnt; echo $?
  > done
  base.tgz: 0
  comp.tgz: 0
  etc.tgz: 0
  games.tgz: 0
  kern-KUROBOX.tgz: 0
  man.tgz: 0
  misc.tgz: 0
  modules.tgz: 0
  tests.tgz: 0
  text.tgz: 0
```

/kern /proc /data を作っておく。

```
  # cd /mnt
  # mkdir kern proc data
```

あとで、使うので、altboot.bin をコピーしておく。

```
# cp -p /mnt/usr/mdec/altboot.bin /var/tmp
# cd /
# umount /mnt
```

作業 PC 上で KURO-BOX/HG のディスク（の NetBSD でつかうところ）をマウントし、NetBSD/sandpoint の初期設定をする。

事前に設定しておく。（詳細は割愛。）

■ /etc/fstab

```
/dev/wd0a               /       ffs     rw,log          1 1
/dev/wd0b               none    swap    sw              0 0
/dev/wd0f               /data   ffs     rw,log          1 1
kernfs          /kern   kernfs  rw
ptyfs           /dev/pts        ptyfs   rw
procfs          /proc   procfs  rw
```

■ /etc/ifconfig.re0

```
inet IP あどれす netmask さぶねっとますく
inet 192.168.11.150 netmask 255.255.255.0 alias
```

固定 IP をつけておく。

KURO-BOX/HG のネットワークインターフェースは、re になる。INSTALL.txt の "NetBSD/sandpoint System Requirements and Supported Devices" 参照。

■ /etc/rc.conf

- rc_configured=YES
- hostname= なにかつけとくとよい
- defaultroute= デフォルトゲートウェイの IP アドレス
- sshd=YES

■ /etc/ssh/sshd_config

root でログインできるように、PermitRootLogin yes しておく。（あとで、no に戻すこと。）

■オプション

- /etc/passwd
  - vipw -d /mnt で設定できる。ハッシュ化された文字列は、pwhash コマンドで作ることができる。
  - ログイン後、最初に root のパスワードをつけるのなら、なにもしなくてもよい。（このままだと、root のパスワードは空のままである。）
- /etc/resolv.conf
  - DNS の設定。あとからしてもよい。
- /etc/hosts
  - 必要に応じてどうぞ。
- /etc/localtime
  - あとからした方がらく。
- /etc/ttys
  - シリアルコンソール化をあとからしたくなった時のために、"115200" のところを、"57600" にしておく。
- /etc/inetd.conf

・必要に応じてどうぞ。
・telnet を開ける場合は、-a valid がついていると、SRA login に対応していない telnet クライアントだと、telnet できないのでちうい。{Net,Free,DragonFly }BSD は OK。OpenBSD や Linux の各ディストリビューション (Netkit telnet) は NG。Windows だと RLogin というターミナルエミュレータはいけるらしい。

終わったら、最後に、MAKEDEV して、umount する。

```
# cd /mnt/dev
# ./MAKEDEV all
# cd /
# umount /mnt
```

作業 PC 上で KURO-BOX/HG のディスクのディスクラベルが NetBSD/sandpoint で扱えるように、NetBSD/i386 -current の disklabel で書き直す

まず、disklabel を書き出す。

```
# disklabel -r sd1 > /tmp/sd1_label
```

次に、書き出したラベルを編集し、c パーティションを消し、d パーティションを c パーティションにする。また、label: 行の " fictitious" を消す。

```
# cp /tmp/sd1_label /tmp/sd1_label.new
# vi /tmp/sd1_label.new
# cat /tmp/sd1_label.new
# /dev/rsd1d:
type: SCSI
disk: OOBB-55RDA0
label:
flags:
bytes/sector: 512
sectors/track: 32
tracks/cylinder: 64
sectors/cylinder: 2048
cylinders: 238475
total sectors: 488397168
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # microseconds
track-to-track seek: 0  # microseconds
drivedata: 0

7 partitions:
#        size    offset     fstype [fsize bsize cpg/sgs]
 a:   2097152   4707045     4.2BSD      0     0     0  # (Cyl.   2298*-   3322*)
 b:    514072   4192973       swap                    # (Cyl.   2047*-   2298*)
 c: 488397168         0     unused      0     0        # (Cyl.      0 - 238475*)
 e:   4192902        63 Linux Ext2      0     0        # (Cyl.      0*-   2047*)
 f: 481592971   6804197     4.2BSD      0     0     0  # (Cyl.   3322*- 238475*)
```

次に、ディスクラベルを消す。

```
# disklabel -D sd1
# disklabel -r sd1
disklabel: could not read existing label
```

次に、NetBSD -current の disklabel で書く。

```
# /var/tmp/src/sbin/disklabel/disklabel -B be -R sd1 /tmp/sd1_label.new
disklabel: changing le byteorder to be: Undefined error: 0
```

エラーが出るが、これで問題ないみたい。

読めるかどうか確認する。

```
# /var/tmp/src/sbin/disklabel/disklabel -B be -r sd1
```

読めないことも確認する。

```
# disklabel -r sd1
disklabel: ignoring byteswapped label at offset 512 from sector 4707045
disklabel: could not read existing label
```

KURO-BOX/HG にディスクを取り付け、Linux 側をいったん起動し、altboot を Linux 側の /boot/altboot.bin に置く

作業 PC の /var/tmp/altboot.bin にとったあったはずなので、コピーしておく。

Linux 側の /boot/altboot.bin に保存する。

NetBSD が起動できるか試す。問題なければ、常時起動するようにする

u-boot のリモートコンソールをとって作業する。

作業前に、設定をメモっておく。（出力がくっつく場合は、何度か試すとよい。）

```
=> printenv
printenv
bootargs=root=/dev/hda1
bootcmd=run bootcmd1
nfsboot=bootp;run nfsargs;bootm
bootdelay=10
baudrate=57600
autoload=no
stdin=nc
stdout=nc
stderr=nc
ipaddr=192.168.11.150
netmask=255.255.255.0
serverip=192.168.11.149
ncip=192.168.11.149
netretry=no
nc=setenv stdin nc;setenv stdout nc;setenv stderr nc
ser=setenv stdin serial;setenv stdout serial;setenv stderr serial
ldaddr=800000
hdpart=0:1
hdfile=boot/vmlinux.UBoot
hdload=echo Loading ${hdpart}:${hdfile};ext2load ide ${hdpart} ${ldaddr} ${hdfile}
boothd=setenv bootargs root=/dev/hda1;bootm ${ldaddr}
hdboot=run hdload boothd
flboot=setenv bootargs root=/dev/hda1;bootm ffc00000
emboot=setenv bootargs root=/dev/ram0;bootm ffc00000
      nfsargs=setenv      bootargs      root=/dev/nfs      rw      nfsroot=${serverip}:${rootpath}
ip=${ipaddr}:${serverip}:${gatewayip}:${netmask}:{hostame}:ff
bootretry=30
bootcmd1=run hdboot;run flboot
bootcmd2=run flboot
bootcmd3=run emboot
 writeng=protect off fff70000 fff7ffff;era fff70000 fff7ffff;mw.l 800000 4e474e47 1;cp.b 800000
fff70000 4
 writeok=protect off fff70000 fff7ffff;era fff70000 fff7ffff;mw.l 800000 4f4b4f4b 1;cp.b 800000
fff70000 4
ubpart=0:3
ubfile=share/u-boot/u-boot-hg.flash.bin
ubload=echo Loading ${ubpart}:${ubfile};ext2load ide ${ubpart} ${ldaddr} ${ubfile}
ubsaddr=fff00000
ubeaddr=fff2ffff
```

```
   ubflash=protect off ${ubsaddr} ${ubeaddr};era ${ubsaddr} ${ubeaddr};cp.b ${ldaddr} ${ubsaddr}
${filesize};cmp.b ${ldadr ${ubsdr}${ileiz}
  upgrade=run ubload ubflash
  ethact=RTL8169#0

  Environment size: 1476/65532 bytes
```

altboot.bin で起動するように設定を変更する。

```
=> ide device
ide device

IDE device 0: Model: WDC WD2500BB-55RDA0 Firm: 20.00K20 Ser#: WD-WCANKF661037
            Type: Hard Disk
            Supports 48-bit addressing
            Capacity: 238475.1 MB = 232.8 GB (488397168 x 512)
=> ide part
ide part

Partition Map for IDE device 0  --   Partition Type: DOS

Partition     Start Sector     Num Sectors     Type
    1                63          4192902      83
    2           4192965           514080      82
    3           4707045        483690123      a9
=> ext2ls ide 0:1 /
ext2ls ide 0:1 /
<DIR>       4096 .
<DIR>       4096 ..
<DIR>       4096 lost+found
<DIR>       4096 bin
<DIR>       4096 boot
<DIR>       4096 cdrom
<DIR>      20480 dev
<DIR>       4096 etc
<DIR>       4096 floppy
<DIR>       4096 home
<DIR>       4096 initrd
<DIR>       4096 lib
<DIR>       4096 mnt
<DIR>       4096 opt
<DIR>       4096 proc
<DIR>       4096 root
<DIR>       4096 sbin
<DIR>       4096 tmp
<DIR>       4096 usr
<DIR>       4096 var
<DIR>       4096 sys
=> ext2ls ide 0:1 /boot
<DIR>       4096 .
<DIR>       4096 ..
           30427 .config
          695026 System.map
         3047558 vmlinux.bin
         1422555 uImage
<SYM>           6 vmlinux.UBoot
           75924 altboot.bin
```

ちゃんと u-boot から HDD は見えているみたいだ。試しに NetBSD を boot してみる。

```
=> ext2load ide 0:1 1000000 boot/altboot.bin

75924 bytes read
=> go 1000000 wd0a:netbsd
## Starting application at 0x01000000 ...
（このあと何も出てこなくなる。これで正しい。シリアルコンソールを出している人はそっちに出ているはず。）
```

あとは、設定した IP アドレスに、作業 PC から ping して応答があれば、root で ssh ログインできるはずである。

問題なければ、常時起動するようにする。

せっかくなので、

```
=> setenv debboothd setenv bootargs root=/dev/hda1\;bootm \${ldaddr}
  => setenv debboot setenv ldaddr 800000\;setenv hdfile boot/vmlinux.UBoot\;setenv boothd
\${debboothd}
=> setenv nbboothd go \${ldaddr} wd0a:netbsd
=> setenv nbboot setenv ldaddr 1000000\;setenv hdfile boot/altboot.bin\;setenv boothd \${nbboothd}
```

としておけば、Linux に切り替えたい時、

```
=> run debboot
```

で、よくなる。また、NetBSD は、

```
=> run nbboot
```

で、よくなる。

u-boot 上で手動で起動する時は、run hdboot である。再起動は reset。

自動起動するほうを、設定の上、保存しよう。

```
=> setenv debboothd setenv bootargs root=/dev/hda1\;bootm \${ldaddr}
  => setenv debboot setenv ldaddr 800000\;setenv hdfile boot/vmlinux.UBoot\;setenv boothd
\${debboothd}
=> setenv nbboothd go \${ldaddr} wd0a:netbsd
=> setenv nbboot setenv ldaddr 1000000\;setenv hdfile boot/altboot.bin\;setenv boothd \${nbboothd}
=> run nbboot
=> saveenv
```

再起動して、自動起動するか確認して、おしまい！

```
=> reset
```

Linux パーティションに altboot.bin を置いたことで、（本家が採用している）u-boot のなかに altboot.bin を保存する必要ななくなった。

苦労したところ

バイトオーダーの問題については、ディスクラベルについては、NetBSD/sandpoint のツールチェーンを用意し、その中にある disklabel コマンド（nbdisklabel-sandpoint）でなんとかなると思っていた。

実際、バイトオーダーは、ビックエンディアンで読み書きできているようで、問題なさげだった。

ところが、とてもとてもとても残念なことに、nbdisklabel-sandpoint で書くと、（理由はよくわからなかったが、）ディスクの先頭にディスクラベルを書く。悲しいことに、altboot は全く理解してくれない。

（NetBSD/i386 の）disklabel コマンドを使うと、ちゃんと、NetBSD パーティションの先頭に書いて

くれるのだが、バイドオーダーは、リトルエンディアンになる。もちろん、これも altboot は全く理解してくれない。

ここで、ずっと詰んでしまい、しばらく放置していた。

ところが、2013 年の 5 月になって、NetBSD -currnt（開発先端）で、newfs の -B オプションと同じオプションが disklabel コマンドにもついた。（以下は、main.c の cvs log。）

```
date: 2013/05/03 16:05:12;  author: matt;  state: Exp;  lines: +276 -26
Make disklabel a MI tool.  It will use MACHINE/MACHINE_ARCH to determine
the disklabel params as well as allowing command-line options of -M <machine>
and -B {le,be} to specify MACHINE and byteorder to be used.
```

これに気づいたのが、2013 年の 9 月くらいで、確認したところ、NetBSD パーティションの先頭に書いてくれるようだったので、テストしてみて、ようやく完成した。

もしかして：シリアルコンソールのはんだ付けをする方がかんたんだったかも（笑）。

その他

時刻のズレについて

NetBSD では、altboot の中で必要な設定が行われている。しかし、この値がよくないので、ntpd を動かしていても間に合わないくらい、結果的にずれが生じることになる。Genbako kernel collection さんのカーネルは、（arch/ppc/platforms/linkstation.c に）手当てがしてあって、問題が少なくなるようになっている。これと同じことをすればよいようだ。具体的には、sys/arch/sandpoint/stand/altboot/brdsetup.c を以下のように変更する。

```
--- brdsetup.c.orig  2013-09-29 22:31:09.000000000 +0900
+++ brdsetup.c  2015-10-18 00:52:35.000000000 +0900
@@ -653,9 +653,9 @@
 {

   if (PCI_VENDOR(pcicfgread(pcimaketag(0, 11, 0), PCI_ID_REG)) == 0x10ec)
-    brd->extclk = 32768000; /* decr 2457600Hz */
+    brd->extclk = 32522240; /* {32.768MHz*(100% - 0.75%)} * 4 / 4 */
   else
-    brd->extclk = 32521333; /* decr 2439100Hz */
+    brd->extclk = 24391680; /* {24.576MHz*(100% - 0.75%)} * 3 / 4 */
 }

 void
```

こうすることで、KURO-BOX/HG では、かなりましになった。

Perl がコンパイルできない

pkgsrc から、あれこれソフトを入れようとすると、Perl がコンパイルできないことでつまづく。これは、swap が足らないせい。swap パーティションを大きくすることはインストール後だと難しいけど、swap ファイルをつくって、それを使うようにすることは難しくない。swapctl のマニュアルをみてやってみよう。

ちなみに、自分は 2GB くらいの swap ファイルを足しています。（なお、ファイルはどんなファイ

ルでもいいので、/dev/zero とかを使って、つくるとよいです。dd if=/dev/zero of=swapfile bs=1024k count=2048 とか。）

りんくしゅ

あれこれ参考になります。

- Buffalo LinkStation Installation Guide: 本家のインストールガイド。
- Kuro-Box: かわうち版なきあと、もっともよく情報がまとまっているページ。
- [NetBSD] 玄箱 HG Update (2): 本家のとおりのやり方でインストールしてみえる例。
- 5 分で分かる Raspberry Pi で NetBSD/evbarm: このあと、シリアルを出そうとされるのであれば、ケーブル側はこのやり方がオススメです。