

なんちゃってブートセクタの説明

注意：このページは古い情報が書かれています。ご本家NetBSD/sandpointには*絶対に*適用しないようにしてください。

LinuxからNetBSDを起動する/しないを選択できるブートセクタ（もどき）を作成しました。

特長

- Linuxのブートシーケンス中に埋め込む形で使用します。
- リードオンリーマウント状態でnbloaderをinsmodしますので、かなり安全にNetBSDをブートできます。
- AVRの制御を自前で行っていますので、起動後に勝手にリブートしてしまうことが（たぶん）有りません。
- シリアルコンソールが有れば、キー入力によりLinux/NetBSDを選択できます。
- シリアルコンソールが無くとも、デフォルト値を保存しておくことでリブートによるOSの切替が可能です。
- 実験カーネル等、不安定なカーネルのために一時的な起動の切り替え(今回はNetBSDを起動するが、次回はLinuxを起動する、など)が可能です。
- Debian化キットのカーネルロードと共存可能です。

『なんちゃって』の理由

Linuxを選んでも特に何もしていないため、ブートセクタとは呼ぶのはおこがましいからです。

使い方

1. NetBSDカーネルの準備

NetBSDカーネルは、サンプルカーネルがそのまま使用可能です。
ダウンロードしたカーネル(netbsd.bin)は、Linux側の/bootにコピーしておきます。

2. なんちゃってブートセクタのファイルのコピー

(ここの内容はmake installで行っているものです。)
kuro_boot.sel.shをLinux側の/etc/init.d/以下に、kuro_boot.confを/etc以下にコピーします。
(kuro_boot.sel.shのパーミッションをチェックし、実行可であることを確認してください。)
また、/etc/rcS.d/にて、
ln -s ../init.d/kuro_boot.sel.sh S15kuro_boot.sel.sh
を実行してシンボリックリンクをしておきます。

3.nbloader_v2.oのコピー

nbloader_v2.oは/lib/modules/2.4.17_mv121/kernel/nbloader_v2.o(玄箱/HG)、または
/lib/modules/2.4.17_kuro-box/kernel/nbloader_v2.o(初代玄箱)としてコピーしておきます。

4.設定

/etc/kuro_boot.confを必要に応じて修正します。
このファイルでは永続的な設定を行います。
一時的な設定を行うためには、このファイルを/etc/kuro_boot.tmpとしてコピーして、こちらを修正します。
内容は以下の通りです。(必要に応じてコメントを外して修正して下さい)

```

# kuro_bootsel config file (2005-11-07)
#
# "bootdefault"
# 1: Linux, 2-5: See below
#bootdefault=1

#
# "timeout"
# prompt timeout (in secs)
#timeout=3

#
# "AVR"
# AVR port name
#AVR=/dev/AVR00

# Menu entry No.2
#
menu2="NetBSD(HDD:wd0)"
loader2=/lib/modules/`/bin/uname -r`/kernel/nbloader_v2.o
kernel2=/boot/netbsd.bin
option2="bootdev=0xa0000000"

# Menu entry No.3
#
#menu3="NetBSD(EM-boot)"
#loader3=/lib/modules/`/bin/uname -r`/kernel/nbloader_v2.o
#kernel3=/boot/emergency.bin
#option3=

# Menu entry No.4
#
#menu4="NetBSD(USB:sd0)"
#loader4=/lib/modules/`/bin/uname -r`/kernel/nbloader_v2.o
#kernel4=/boot/netbsd.bin
#option4="bootdev=0xa0000004"

# Menu entry No.5
#
#menu5="Linux(New-kernel)"
#loader5=/lib/modules/`/bin/uname -r`/kernel/loader.o
#kernel5=/boot/linux.bin
#option5="cmdline= \"root=/dev/hda1 \ " "

```

bootdefault

デフォルトのOSを指定します。1ならLinuxで、2~5は指定したものとなります。無指定時は1(Linux)です。

timeout

コンソールでのOS選択のプロンプト時の入力タイムアウト時間(秒)です。デフォルトは3秒です。入力が無くこの時間経過すると、デフォルトのOS(bootdefaultで指定)が起動します。

AVR

Linux側から見えるAVRポートのデバイス名です。デフォルトは/dev/AVR00です。

間違えるとハマりますので注意して設定して下さい。

menu[2345]

メニューに表示する文字列です。

loader[2345]

loader.oやnbloader_v2.oのある場所（フルパス）です。
kernel[2345]
LinuxやNetBSDのカーネルの名前（フルパス）です。
option[2345]
ブートローダへ渡す引数を記述します。

5.起動

設定が終わったらおもむろにリブートします。シリアルコンソールが有ると、ブート途中で次のような表示が出るはずで

Select the OS: (default is 1)

1: Linux, 2: NetBSD(HDD:wd0)

boot>

ここで1を押すとそのままLinuxが起動し、2を押すとNetBSDが起動します。3秒間放置するとデフォルト(この例ではLinux)が起動します。

NetBSDの不安定カーネル/ユーザランドの起動のためのTips

実験のため不安定なカーネルを起動する場合、シリアルコンソールが無いと毎度その不安定なカーネルのブートに行ってしまう、ハマってしまいます。

これを回避するためには、/etc/kuro_boot.tmpを作成し、NetBSDをブートするように設定します。/etc/kuro_boot.conf側はLinuxを起動するように設定しておきます。

kuro_boot.tmpはkuro_boot.confよりも優先して読み込まれます。また、一旦起動すると削除されま

す。このため、次に起動した場合はkuro_boot.confが使用される形になります。

これで、たとえNetBSDのブートではまっても、おもむろに電源を切り再投入することでLinux側が起動する、と言う形にすることが出来ます。（実験の度に毎度設定は面倒でしょうが、安全のためです。）

カーネルが不安定で無くとも、ユーザランドの設定がおかしい際でもこれが有効だと思います。シリアルコンソールが無い場合にはこれをうまく活用してください。

その他

たぶんバグがあります。また、/etc/rcS.d/の入れる場所については議論の分かれる所だと思います。

/etc/rcS.d/の入れる場所によっては、kuro_boot.tmpの削除が出来ない場合があります。

少なくともルートパーティションがリードライト可能になった後に入れることが必要です。

kuro_bootsel.shの中身(2005-11-07版)

ライセンスはBSDとします。

文字列表示のとき、ベル(^G)を鳴らしている所があります。コピペの際はご注意ください。

```
#!/bin/bash
#
# KURO-BOX boot selector (2005-11-07)
# (c) 2005 Eiji Kawauchi
#
# default values
timeout=3
bootdefault=1
AVR=/dev/AVR00
```

```

menu2="NetBSD(HDD:wd0a)"
loader2=/lib/modules/`/bin/uname -r`/kernel/nbloader_v2.o
kernel2=/boot/netbsd.bin
option2="bootdev=0xa0000000"

menu3=
loader3=
kernel3=
option3=

menu4=
loader4=
kernel4=
option4=

menu5=
loader5=
kernel5=
option5=

# These may override above
if [ -f /etc/default/kuro_boot.conf ]; then
    . /etc/default/kuro_boot.conf
fi
if [ -f /etc/kuro_boot.conf ]; then
    . /etc/kuro_boot.conf
fi
if [ -f /etc/kuro_boot.tmp ]; then
    . /etc/kuro_boot.tmp
    rm -f /etc/kuro_boot.tmp
fi

# functions
init_avr()
{
    # AVR Control
    /bin/stty -F $AVR cs8 parenb -parodd clocal 9600
    /bin/echo -n 'AAAAFFFFJJJJ>>>>VVVV>>>>ZZZZVVVVKKKK' > $AVR
}
boot_kernel()
{
    if [ "$1"x = "x ]; then
        #echo "Loader not specified."
        return 0;
    fi
    if [ ! -f $1 ]; then
        #echo "Loader does not exist."
        return 0;
    fi
    if [ "$2"x = "x ]; then
        echo "Kernel not spicified."
        return 0;
    fi
    if [ ! -f $2 ]; then
        echo "Kernel does not exist."
        return 0;
    fi
}

```

```

fi
init_avr;
/bin/sync; /bin/sync;
/bin/umount -a
/bin/mount -n -o remount,ro /
/sbin/inssmod $1 kernel=$2 $3
}

select_boot()
{
    menustr="1: Linux"
    if [ "$menu2"x != ""x ]; then
        menustr="$menustr, 2: $menu2"
    fi
    if [ "$menu3"x != ""x ]; then
        menustr="$menustr, 3: $menu3"
    fi
    if [ "$menu4"x != ""x ]; then
        menustr="$menustr, 4: $menu4"
    fi
    if [ "$menu5"x != ""x ]; then
        menustr="$menustr, 5: $menu5"
    fi
    /bin/echo "*****"
    /bin/echo "Select the OS: (default is $bootdefault)"
    /bin/echo $menustr "^G^G"
    read -t $timeout -n 1 -s -p "boot> " key
    if [ "$key"x = ""x ]; then
        key=$bootdefault
    fi
    /bin/echo $key
}

if [ "$1"x = "start"x ]; then
    select_boot
    case $key in
    2) boot_kernel "$loader2" "$kernel2" "$option2" ;;
    3) boot_kernel "$loader3" "$kernel3" "$option3" ;;
    4) boot_kernel "$loader4" "$kernel4" "$option4" ;;
    5) boot_kernel "$loader5" "$kernel5" "$option5" ;;
    *) ;;
    esac
fi

```

ご連絡先: E.Kawauchi atto gmail dotto com