

KiNOKO-DAQ Technical Tips

- [デバイスドライバをシステム起動時に自動で組み込む](#)
- [共有メモリセグメントの最大サイズを変更する](#)

デバイスドライバをシステム起動時に自動で組み込む

Linux 起動時に camdrv などのデバイスドライバを自動でを組み込む方法について解説します。これにより、再起動するたびに毎回 make install を実行しなくてもすむようになります。ここでは、camdrv を例として説明しますが、cam を vme に変えればそのまま vmedrv にも適用できます(ただし、vmedrv は、**ドライバインストール時に VME クレートの電源が入っている必要がある**ことに注意してください)。

ここに書いてある内容は、ディストリビューションやそのバージョンにより若干異なっている可能性があります。ここでは RedHat Linux および Debian GNU/Linux を念頭に置いていますが、詳細についてはそれぞれのディストリビューションのドキュメントを参照してください。

[簡単な方法]

/etc/rc.d/rc.local ファイル(Debian では /etc/rc.local)の末尾に以下の内容を追加します。

```
cd /usr/local/camdrv/Linux2.4_GC77pci
make install
cd -
```

ここで、先頭行の /usr/local は、camdrv を実際にインストールしたディレクトリに置きかえてください。

メジャー番号を決定する

camdrv はキャラクタ型デバイスドライバと呼ばれるもので、システム中では1つの番号で識別されます。その番号がメジャー番号で、キャラクタ型デバイスドライバは他のキャラクタ型ドライバと重ならないメジャー番号を持たなければなりません。どのドライバがどの番号を持っているかは、/dev のリストを見れば分かります。

```
% ls -l /dev
crw----- 1 root  root  10, 134 Nov 26 22:28 apm_bios
crw----- 1 root  root  10,  3 Jul 29 1999 atibm
crw-rw---- 1 root  audio  14,  4 Oct  6 1999 audio
crw-rw---- 1 root  audio  14, 20 Oct  6 1999 audiol
brw-rw---- 1 root  disk   41,  0 Jul 29 1999 bpcd
lrwxrwxrwx 1 root  root    8 Oct  7 1999 cdrom -> /dev/hdc
brw-rw---- 1 root  disk   24,  0 Jul 29 1999 cdu535
lrwxrwxrwx 1 root  root    4 Oct  7 1999 console -> tty0
lrwxrwxrwx 1 root  root   11 Oct  7 1999 core -> /proc/kcore
crw-rw---- 1 root  audio  14,  3 Oct  6 1999 dsp
crw-rw---- 1 root  audio  14, 19 Oct  6 1999 dsp1
crw--w--w  1 root  tty    29,  0 Jul 29 1999 fb0
crw--w--w  1 root  tty    29,  1 Jul 29 1999 fb0autodetect
```

(以下延々と続く)

%

ここで、一文字目が c になっているものがキャラクタ型ドライバです。そして、カンマの前の数字(一行目では10)が、そのドライバに割り当てられているメジャー番号です。このリストを全部見て、空いていそうなところを探します。64番以降はあまり使われていないはずですが、ただし、127番からは降順で自動割り当てに使われることが多いので、その周辺も避けます。また、カーネルバージョンによって異なる上限値がある(少なくとも127以上のはず)ので、あまり大きい値は使えません。ここでは、空いていて使われそうにない番号87を使うことにします。

デバイスエントリを作成する

- camdrv の場合

```
# rm -f /dev/camdrv
# mknod /dev/camdrv c 87 0
# chmod 666 /dev/camdrv
```

ここで、87 は、上で決めたメジャー番号です。/dev のリストを表示させて、エントリが作成されていることを確認します。

```
# ls -l /dev/camdrv
crw-rw-rw- 1 root root 87, 0 May 6 20:21 /dev/camdrv
```

- vmedrv の場合

```
# rm -f /dev/vmedrv*
# mknod /dev/vmedrv c 87 0
# mknod /dev/vmedrv16d16 c 87 1
# mknod /dev/vmedrv16d32 c 87 2
# mknod /dev/vmedrv24d16 c 87 3
# mknod /dev/vmedrv24d32 c 87 4
# mknod /dev/vmedrv32d16 c 87 5
# mknod /dev/vmedrv32d32 c 87 6
# mknod /dev/vmedrv24d16dma c 87 7
# mknod /dev/vmedrv24d32dma c 87 8
# mknod /dev/vmedrv32d16dma c 87 9
# mknod /dev/vmedrv32d32dma c 87 10
# chmod 666 /dev/vmedrv*
```

ここで、87 は、上で決めたメジャー番号です。/dev のリストを表示させて、エントリが作成されていることを確認します。

```
# ls -l /dev/camdrv
crw-rw-rw- 1 root root 87, 0 May 6 20:21 /dev/vmedrv
crw-rw-rw- 1 root root 87, 1 May 6 20:21 /dev/vmedrv16d16
crw-rw-rw- 1 root root 87, 2 May 6 20:21 /dev/vmedrv16d32
crw-rw-rw- 1 root root 87, 3 May 6 20:21 /dev/vmedrv24d16
crw-rw-rw- 1 root root 87, 4 May 6 20:21 /dev/vmedrv24d32
(以下省略)
```

ドライバを規定の場所に置く

modprobe によって検索される場所にドライバをコピーします。その場所は、システムにより異なります。詳しくは、マニュアルページ modprobe(1)などを参照してください。ここでは、/lib/modules/2.2.14 が検索パスだとして話を進めます。

```
# cp camdrv.o /lib/modules/2.2.14/misc/camdrv.o
```

モジュール設定ファイルを編集する

まず、/lib/modules/2.2.14/ (ドライバを置いたモジュール検索パス)にある modules.dep に以下の行を追加します。

```
/lib/modules/2.2.14/misc/camdrv.o:
```

次に、/etc/modules.conf (このファイルがなければ、/etc/conf.modules) に以下の内容を追加します (RedHat の場合)。ioport および irq の値は自分の設定に合わせてください (PCI版のコントローラを用いている場合、ioport および irq の記述は不要です)。

```
options camdrv camdrv_major=87 ioport=0x00e0 irq=5
```

Debian の場合、ディレクトリ /etc/modutils に camdrv というファイルを作って、上記の内容を記述し、update-modules を実行します。

```
# cat > /etc/modutils/camdrv
options camdrv camdrv_major=87 ioport=0x00e0 irq=5
^D
# update-modules
```

ここまでの作業で、modprobe によるロードができるようになっているはずですが、以下のように確認してください。

```
# /sbin/modprobe camdrv
# cat /proc/devices | grep camdrv
87 camdrv
# /sbin/modprobe --remove camdrv
```

起動スクリプトを追加する

[簡単な方法]

- RedHat の場合

/etc/rc.d/rc.local ファイルの末尾に以下の内容を追加します。

```
action "Installing camdrv..." /sbin/modprobe camdrv
```

- Debian の場合

/etc/rc.local ファイルの末尾に以下の内容を追加します。

```
/sbin/modprobe camdrv
```

これによりこの章に書かれている内容は省略できます。

/etc/rc.d/init.d (Debian では /etc/init.d) に以下の内容を書いた camdrv というファイルを作成します。

```
#!/bin/sh
#
# Init file for camdrv
#

start() {
    /sbin/modprobe camdrv
    touch /var/lock/subsys/camdrv
}

stop() {
    /sbin/modprobe --remove camdrv
    rm -f /var/lock/subsys/camdrv
}

restart() {
    stop
    start
}

status() {
    if [ -f /var/lock/subsys/camdrv ] ; then
        echo "camdrv is installed"
    else
        echo "camdrv is not installed"
    fi
}

case "$1" in
    start)
        start
    ;;
```

```

;;
stop)
    stop
;;
restart|reload|force-reload)
    restart
;;
status)
    status
;;
*)
    echo "Usage: $0 {start|stop|restart|reload|force-reload|status}"
    exit 1
esac

exit $?

```

次に、このスクリプトを実行可能に設定し、各ランレベルに対応する起動スクリプトディレクトリからリンクを張ります。

- RedHat の場合

```

# chmod 755 /etc/rc.d/init.d/camdrv
# ln -s /etc/rc.d/init.d/camdrv /etc/rc.d/rc2.d/S98camdrv
# ln -s /etc/rc.d/init.d/camdrv /etc/rc.d/rc3.d/S98camdrv
# ln -s /etc/rc.d/init.d/camdrv /etc/rc.d/rc5.d/S98camdrv
# ln -s /etc/rc.d/init.d/camdrv /etc/rc.d/rc0.d/K02camdrv
# ln -s /etc/rc.d/init.d/camdrv /etc/rc.d/rc6.d/K02camdrv

```

- Debian の場合

```

# chmod 755 /etc/init.d/camdrv
# ln -s /etc/init.d/camdrv /etc/rc2.d/S98camdrv
# ln -s /etc/init.d/camdrv /etc/rc3.d/S98camdrv
# ln -s /etc/init.d/camdrv /etc/rc4.d/S98camdrv
# ln -s /etc/init.d/camdrv /etc/rc5.d/S98camdrv
# ln -s /etc/init.d/camdrv /etc/rc0.d/K02camdrv
# ln -s /etc/init.d/camdrv /etc/rc6.d/K02camdrv

```

ここで、リンク先となっている `/etc/rc2.d` などの中の数字(この例では 2) は、ランレベルを示します。各ランレベルは、`/etc/inittab` ファイルに定義されています。以下は、RedHat の `/etc/inittab` の例です(抜粋)。

```

# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#

id:3:initdefault:

```

ここで、`id:3:initdefault` が、デフォルトランレベルが 3 であることを示しています。この値は Debian では 2 になっているかもしれませんが、`xdm` などのグラフィカルログインを使っていれば 5 になっていると思います。

それぞれのランレベルに対応する `rc*.d` ディレクトリに `S98camdrv` という名前でシンボリックリンクを作成すれば、そのランレベルで `camdrv` がインストールされます。また、`K02camdrv` という名前でシンボリックリンクを作成すると、そのランレベルで `camdrv` がアンインストールされるようになります。上の例では、ランレベル 0 および 6 が終了に対応するので `K02camdrv` を、それ以外のランレベルに対して `S98camdrv` を作成しています。

この類の処理を自動化するユーティリティとして、RedHat には `ntsysv` や `chkconfig` などが、Debian には `update-rc.d` があります。

再起動と確認

```
# reboot
:
:
# cat /proc/devices | grep camdrv
87 camdrv
```

共有メモリセグメントの最大サイズを変更する

Linux 2.2/2.4

現在の共有メモリセグメントの最大サイズを知る

```
# cat /proc/sys/kernel/shmmax
33554432 (Byte)
#
```

共有メモリセグメントの最大サイズを変更する

```
# echo 134217728 > /proc/sys/kernel/shmmax
#
```

[mailto: sanshiro@awa.tohoku.ac.jp](mailto:sanshiro@awa.tohoku.ac.jp)