

# FreeBSDでMirakurunを動かす

2019/01/28

色々コメントを貰いドキュメント構成を変更。情報追記。

2017/05/21

FreeBSD 11.0 RELEASEで無事導入までできたのでその際のメモ書き。

Mirakurun 2.6.1 はnode6サポートを終了しnode8を使われたしUpdateすると当然パッチあてたファイルも書き換わるから再度修正がいる。

## Mirakurunって何？

DVR Tuner Server Service for Chinachu Air. <https://github.com/Chinachu/Mirakurun>

Chinachu Airが何ぞや、という話は置いておく。

Mirakurunはデジタル放送チューナーデバイスを管理するサービス。recpt1等物理層直叩きなプログラムを隠蔽し、HTTPのREST API経由で操作可能としている。面倒なチューナーの管理を不要とできるので、このサービスに登録したチューナーデバイスは複数のプログラムやサービスで共用する事が可能になる。

このサービスからEPG情報の取得や放送のストリーミング視聴ができるし、効率的なチューナーの利用がされるので、人によっては録画可能な番組が増えるかもしれない。

例えばTOKYO MXの地上波デジタル放送だとマルチチャンネルと称しMX1,MX2の2放送が1つのチャンネルで送信されている。

チューナーが2つある環境で同時間帯のMX1,MX2の放送を受信したい場合、

- recpt1コマンド等単体の録画プログラムを使う場合MX1,MX2それぞれ用にチューナーが2つ使用されることになる。同時間帯の他の放送を録画できないし、リアルタイム視聴する事もできない。
- Mirakurun(のサービス)を使う場合、1つのチューナーで受信が行われMX1,MX2にデータが分離されて放送データ提供が行われる。空きチューナーで同時間帯の他の放送を録画したり、リアルタイム視聴する事もできる。

という恩恵がある。

## 前提

ここの説明ではPT2/PT3 が稼働していて、recpt1で録画が可能な状態であることが前提。

先に、いくつかディレクトリを作る。もしかすると/usr/local/var は既に存在しているかもしれない。ディレクトリ/usr/local/var/log はパーミッションをいじっておかないとMirakurunがブツクサ言うてくるのでとりあえずの変更。Chinachuを導入した時にChinachuがブツブツ...

```
# mkdir /usr/local/etc/mirakurun
# mkdir /usr/local/var
# mkdir /usr/local/var/log
```

```
# chmod guo+rwx /usr/local/var/log
```

追記 2019/01/28

また私の環境下では不要だったようなのですが、[@HackerMaskee](#)さんからの情報で、

```
# mkdir -p /usr/local/var/db/mirakurun
```

を実行しておく必要があったとの事です。

## インストール

Node.js配下で動作するサービスなので、それらも含めて前提ソフトウェアを先に導入しておく  
Mirakurunの前提環境のバージョン情報から、以下を導入したPortsでもpkgでも好きな方から導入すればいい。

また、こちら [Mirakurun](#) を [jail](#) の中で動かす。でjailに隔離する方法があるので必要な場合はこちらをご確認ください。

### 前提ソフトウェアの導入

- [www/node6](#) [www/node8](#)
- [www/npm3](#) [www/npm5](#)
- [shells/bash](#)
- [arib-b25-stream-test](#)

#### **www/node8, www/npm5**

Portsなりpkgなりで導入する。

#### **shells/bash**

bashインストール後は以下のシンボリックリンクを張っておく。どうも /usr/local/bin/bash ではなく /bin/bash が前提らしい。

```
ln -s /usr/local/bin/bash /bin/bash
```

#### **arib-b25-stream-test** インストール

こちらは[Mirakurun](#) を [FreeBSD](#) にインストールする(2018-12-12版)の手順そのまま。

```
fetch
https://registry.npmjs.org/arib-b25-stream-test/-/arib-b25-stream-test-0...
tar xf arib-b25-stream-test-0.2.9.tgz
```

```
mv package arib-b25-stream-test
cd arib-b25-stream-test
mv package.json package.json.orig
sed -e 's/"linux"/"linux","freebsd"/g' -e 's/make /make CC=cc CXX=c++ /g'
package.json.orig | jq . > package.json
npm install . -g --unsafe
```

## Mirakurunのインストール

今回はユーザ mirakurun を作ってホームディレクトリ下にgitでリポジトリをクローンしている。また、2017-05-19 のコミットで番組情報が集積されない不具合が修正されているので古いソースを取得しなかった人は取り直し。

```
# cd /home/mirakurun
# git clone https://github.com/Chinachu/Mirakurun/
# cd Mirakurun
```

## サポートOSにFreeBSDを追加

### package.json.diff

```
*** package.json.org      2017-05-20 04:56:44.531365000 +0900
--- package.json          2017-05-20 04:56:44.532775000 +0900
*****
*** 92,97 ****
    "os": [
      "linux",
      "darwin",
!     "win32"
    ]
  }
--- 92,98 ----
    "os": [
      "linux",
      "darwin",
!     "win32",
!     "freebsd"
    ]
  }
```

## Server.config に Listenアドレスのパラメタを追加

オリジナルは、利用可能なすべてのnicのすべてのアドレスでListenするようになっている。

NIC1枚でIPを1つ割り当てをしている様な普通の環境なら問題ない。  
が、うちはNIC2枚かつ各NICで複数IPアドレスを振っているのでもちと困る。

### config.ts.diff

```
*** src/Mirakurun/config.ts.org      2017-05-20 04:58:10.676229000
+0900
--- src/Mirakurun/config.ts        2017-05-20 04:58:10.677490000 +0900
*****
*** 24,29 ****
--- 24,30 -----

    // as Remote Server
    readonly port?: number;
+   readonly addr?: string;

    /** `true` to disable IPv6 listening */
    readonly disableIPv6?: boolean;
```

### Server.ts.diff

```
*** src/Mirakurun/Server.ts.org      2017-05-20 04:59:34.098231000
+0900
--- src/Mirakurun/Server.ts        2017-05-20 04:59:34.098925000 +0900
*****
*** 44,54 ****
        }

        if (serverConfig.port) {
!         addresses = [
!             ...addresses,
!             ...system.getPrivateIPv4Addresses(),
!             "127.0.0.1"
!         ];

        if (serverConfig.disableIPv6 !== true) {
--- 44,62 -----
        }

        if (serverConfig.port) {
!         if (serverConfig.addr) {
!             addresses = [
!                 ...addresses,
!                 serverConfig.addr,
!                 "127.0.0.1"
!             ];
!         } else {
!             addresses = [
!                 ...addresses,
!                 ...system.getPrivateIPv4Addresses(),
!                 "127.0.0.1"
!             ];
```



```
port: 40772
addr: 192.168.1.100
```

## tuners.yml

この定義は

- チューナーはPT2が1枚で地上波デジタル放送受信だけ使用。
- recpt1コマンドをチューナーアクセスに使う。-b25オプションで常時スクランブル解除する。
- デコーダーは未指定。

デバイス名はFreeBSDでのデバイス名でLinuxのそれとは違うので注意。

```
- name: PT2-T1
  types:
    - GR
  command: recpt1 --b25 --device /dev/ptx0.t0 <channel> - -
  decoder: ~
  isDisabled: false

- name: PT2-T2
  types:
    - GR
  command: recpt1 --b25 --device /dev/ptx0.t1 <channel> - -
  decoder: ~
  isDisabled: false
```

次の定義は

- チューナーはPT2が1枚で地上波デジタル放送受信だけ使用。
- recpt1コマンドをチューナーアクセスに使う。
- デコーダーに arib-b25-stream-test を利用する。

```
- name: PT2-T1
  types:
    - GR
  command: recpt1 --device /dev/ptx0.t0 <channel> - -
  decoder: arib-b25-stream-test
  isDisabled: false

- name: PT2-T2
  types:
    - GR
  command: recpt1 --device /dev/ptx0.t1 <channel> - -
  decoder: arib-b25-stream-test
  isDisabled: false
```

## channels.yml

自分の地域での受信環境に合わせる。私の場合は以下だけ追加。

```
- name: TVK
  type: GR
  channel: '18'
  isDisabled: false
```

## Mirakurun起動

引数無しだとヘルプが表示される。

```
# mirakurun

Usage: mirakurun <command> ...

<command>:

config server      Edit server configuration.
config tuners      Edit tuner configuration.
config channels    Edit channels configuration.

log                Stream logs.
log --help         Show usage for log stream.

status             Show status of services.
start              Start services.
stop               Stop services.
restart            Restart services.

version            Version info.
help               Output this information.

#
```

以下のコマンドで起動する。

```
# mirakurun start
```

コンソール上でログを見る場合は以下のコマンドで。

```
mirakurun log server
```

追記 2019/01/28

私の環境下では発生しなかったのですが、@HackerMaskeeさんからの情報で、

```
# mirakurun start
[PM2][ERROR] script not found : /home/mirakurun/Mirakurun/mirakurun-server
```

```
#
```

とエラーが発生して起動せず、次の手順が必要になったとのこと。

```
# cd /usr/local/lib/node_modules/mirakurun
# pm2 start processes.json
```

## 録画クライアント

クライアント rivarun をインストールする。

```
# npm install rivarun -g
```

引数無しでヘルプ表示。

```
# rivarun

Usage:
rivarun [--b25] [--mirakurun host:port] [--priority priority] [--sid SID] [-
-ch type/channel] rectime destfile
Remarks:
* if rectime is `-', records indefinitely.
* if destfile is `-', stdout is used for output.
* if `--sid` option specified, will ignore `--ch` option.
Options:
--b25:                Send decode request
--mirakurun host:port: Specify Mirakurun hostname and portnumber
--priority priority:  Specify client priority (default=0)
--sid SID:            Specify SID number
--ch type/channel    Specify channeltype and channel
                    type = GR | BS | CS | SKY
--help:              Show this help
--version:           Show version
--list:              Show channel list

#
```

ほぼ recpt1と同じように利用できる。

```
# recpt1 --b25 --sid hd 18 180 ch18.ts ← recpt1で18チャンネルを180秒間ファイル
ch18.tsへ記録する
# rivarun --b25 --ch GR/18 180 ch18.ts ← rivarunで18チャンネルを180秒間ファイル
ch18.tsへ記録する
# rivarun --b25 --mirakurun 10.0.0.100:40772 --ch GR/18 180 ch18.ts ← IPア
ドレス10.0.0.100で稼働しているMirakurunで18チャンネルを180秒間ファイル
ch18.tsへ記録する
```

Mirakurunのtuners.ymlの設定でdecoderオプションを指定できているならrivarunの- b25オプションは意味あり□ decoder未指定の場合は、recpt1の-b25 オプションでスクランブル解除を行うことになるか

と。  
Mirakurunが稼働しているマシンがリモートにあるのなら `-mirakurun` オプションでMirakurunのサービスを指定できる。

## ストリーミング

recpt1のようにVLCメディアプレイヤーでストリーミング視聴もできる。以下は192.168.1.20:40772で稼働しているMirakurunに接続しチャンネル16をストリーミングする例。



サービスIDを指定しないため、チャンネルによってはワンセグがストリーミングされるかもしれない。その場合にはサービスIDを明示する。



サービスIDはMirakurunのAPIで確認ができる。

```
192.168.1.20:40772/api/channels
1:
  type: "GR"
  channel: "16"
  name: "TOKYO MX"
  services:
    0:
      id: 3239123608
      serviceId: 23608
      networkId: 32391
      name: "TOKYO MX1"
    1:
      id: 3239123609
      serviceId: 23609
      networkId: 32391
      name: "TOKYO MX1"
    2:
      id: 3239123610
      serviceId: 23610
      networkId: 32391
      name: "TOKYO MX2"
    3:
      id: 3239123992
      serviceId: 23992
      networkId: 32391
      name: "MXワンセグ1"
    4:
      id: 3239123993
      serviceId: 23993
      networkId: 32391
      name: "MXワンセグ2"
```

以下のrivarunコマンドでAPIから取得する事もできる。

```
# rivarun --list | nkf
```

rivarunの表示はutf-8で行われるのでnkfで文字化けを抑止している　　うちの環境はeuc-jpなので。

またはChinachu γを導入するのもよい。 <https://github.com/Chinachu/Chinachu>

Chinachu γならブラウザでストリーミング視聴出来たり、予約録画や番組表を参照できる。

[技術資料](#), [FreeBSD](#), [PT1](#), [PT2](#), [PT3](#), [Mirakurun](#), [rivarun](#)

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/freebsd/ptx/ptx-002>

Last update: **2024/11/01 16:25**

