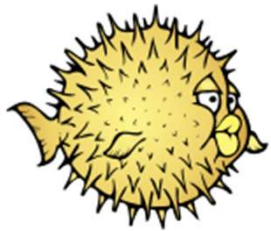


河豚板使ってみた2



OpenBSD

塩路 和彦 shioji@shioji.org



ChatGPT

LOGOS.ADAM96.COM

前回(2025年1月)

- 余っているNote PCに河豚板をインストール
- USBからbootできずに悩んだり
- 購入したUSBメモリが怪しかったり
- いつもの癖で再起動して設定が消えたり

- rsyncでバックアップしようとしたが途中で力尽きました
- その後のお話です

今回のNote PC

- 現在メインで使用しているNote PCの前にメインで使用していたもの
- メインPCが何らかの理由で使用できない場合に仕事で使用するためインストールされているWindows10は削除できない
- 河豚板ならUSBだけで使用できるので問題なし！

ThinkPad X260

CPU Core i7(第6世代)

メモリ 16GB

SSD 2TB(Windows10で使用)

USBメモリに河豚板をインストール

- USBメモリは手元にあった16GBの小さなUSBメモリを使用
- Windows11でUSBへの書き込みはrufusを使用
- fuguita.orgから最新のimgファイルをダウンロード
 - fuguita-7.8-amd64-202511171.img
- このUSBメモリに河豚板の環境を構築



タイムゾーンの変更

- OpenBSDではハードウェア時間(RTC)はUTC(協定世界時)に設定されている
- 河豚板ガイドを参考に/etc/localtimeのシンボリックリンクを変更し、時間のずれを合わせると、Windowsで立ち上げた時に時間がずれる
- ChatGPTに相談するとfuguita側で調整するのは困難
 - TZ=UTCで上書きしたり、独自tzdataを作成したり
- 最終的にWindowsでもRTCをUTCに設定することで解決
 - レジストリの編集は必要

rsyncでのバックアップ

- 今回インストールしたUSBメモリに保存領域を作ってバックアップするのには挫折
 - そのため今回は小さめのUSB(16GB)を使用
- 新しいUSBメモリを用意してそこにバックアップデータを保存することに
 - バックアップ用のUSBは256GBのものを用意
- 新しいUSBメモリを指すと「/dev/sd2」と認識された
 - /var/log/messagesを見ることで確認できる

USBメモリをmount

- `fdisk -i sd2`
- `disklabel -E sd2`
 - `sd2> a [Enter]`、その後はほぼデフォルトでOK
 - 最後に`w`、`q`で書き込み
- `newfs sd2a`
- `mkdir /ram/opt`
- `mount /dev/sd2a /ram/opt`
- これで新しいUSBメモリに書き込みができるように

再起動時に自動マウント

- /etc/fuguita/fstab.tailを作成
- /dev/sd2a /ram/opt ffs rw 0 1と記載
- usbfadmでsyncしたあとrebootしてマウントされるか確認
 - syncするのを忘れると設定が全て消えてしまう

rsyncでバックアップ

- 認識されたUSBメモリにrsyncを利用してバックアップする
- Epsilon.shioji.jpの/home/shioji/backupを/ram/opt/Epsilon/backupにバックアップすることに
- rsyncはEpsilonで実行するためEpsilonからgammaにパスワードなしでログインできるようにsshの設定をする
- 具体的にはEpsilonの共通鍵をgammaに登録

rsyncでバックアップ2

- Epsilonの./ssh/id_rsa.pubをgammaにコピー
 - id_rsa.pubがない場合はssh-keygen -t rsaで作成する
- scp -P 3522 ./ssh/id_rsa.pub gamma:/home/shioji
- gamma上で \$ cat id_rsa.pub >> .ssh/authorized_keys
- Epsilonから \$ ssh -p 3522 gamma でパスワードなしで接続されることを確認
- \$ rsync -az --delete -e "ssh -p 3522" /home/shioji/backup gamma:/ram/opt/Epsilon で確認
- 問題なければcronに登録

SAGによるモニタリング

- System Activity Grapher (SAG)をインストール
- 河豚板ガイドをみてinstall.shをダウンロード
- # sh ./install.shで基本的にはデフォルトでOK
- 河豚板ガイドのとおり、最後のpostgproc.shのみ編集
 - exit 0をコメント
 - sleep 60とその次の行を非コメント化
- データは/var/www/htdocs/sag/に保存される

次はWeb server

- デフォルトでOpenHTTPDはインストールされている
- 設定ファイルは/etc/httpd.conf、とりあえず動かすだけなら以下でOK

```
server "default" {  
    listen on * port 80  
    root "/htdocs"  
}
```

- # rcctl enable httpdで有効化したあと、# rcctl start httpdで起動

httpsでアクセスするためLet's encryptの設定

- OpenBSDではacme-clientで簡単に使える？
- まずは/etc/acme-client.confを編集し# acme-client -v shioji.jpをするがエラーが出て先に進まず
- 最終的にこれでOKに
- 今はfullchain.pemを使う？

```
authority letsencrypt {  
    api url https://acme-v02.api.letsencrypt.org/directory  
    account key "/etc/acme/letsencrypt.key"  
}  
  
domain shioji.jp {  
    alternative names { www.shioji.jp }  
    domain key "/etc/ssl/private/shioji.jp.key"  
    domain certificate "/etc/ssl/shioji.jp.crt"  
    sign with letsencrypt  
}
```

httpsでアクセスするためLet's encryptの設定2

- # acme-client -v shioji.jp
- でうまくファイルが作られることを確認
- 次にhttpd.confを編集し、httpsでアクセスできるように
- 最初ルータの443を空けるの忘れていて悩んだ
- httpd.confを作成するにあたりChatGPTに聞いたがかなり難渋

次はNamazuによる検索に挑戦

- Namazuのインストールはpkg_addでOK
- インデックスファイルを作るのに難渋
- 以前作ったMLをmhonarcでhtml化したものでインデックスを作ろうとするがうまくいかない
- ファイルの漢字コードがShift-JISのため？
- nkfでUTF-8にしてもエラー？
- 結局うまく行かず、とりあえず以前作ったインデックスファイルを流用することに

まずはCGIが使えるように

- CGIを使うためにはslowcgiを有効化する必要あり
- /etc/httpd.confにCGIを使うため次を追記

```
location "/cgi-bin/" {  
    root "/"  
    fastcgi socket "/run/slowcgi.sock"  
}
```

- まずはCGIが使用できるかテスト

test.cgi

- まずtest.cgiとして以下の様なファイルを作成しcgi-binに保存

```
#!/bin/sh  
  
echo "Content-Type: text/plain"  
echo  
echo "CGI OK"
```

- 当初エラーが出て思うような結果が得られず
- いろいろ調べるとOpenHTTPD(?)ではchroot /var/wwwされるため/bin/shが見つからないことが原因
- cp /bin/sh /var/www/bin/として/var/www/bin/shをコピー

つぎにnamazu.cgiが使えるように

- ChatGPTに訪ねるとその前にperlが使えるようにする必要ありと
- `#!/usr/bin/perl`が使えるように`/usr/bin/perl`を`/var/www/bin`にコピー
- しかしperlを使うためには`/usr/lib/libperl.so.26.o`や`/usr/lib/libm.so.10.1`などもコピーする必要あり
- さらに`/usr/local/libdata/perl5/site_perl`や`/usr/libdata/perl5/strict.pm`などのコピーもする必要あり
- 河豚板では`/usr/libdata/perl5`以下のファイルは`/fuguita/usr/libdata/perl5`とリンクしているため実際は`cp /fuguita/usr/libdata/perl5/* /var/www/usr/libdata/perl5/`でコピー

何とかperlも使えるようになった

- しかしnamazuで検索するにはnamazu.cgiが実行できればいいらしい
- namazu.cgiはPerlのスクリプトではなくバイナリファイル
- 結局# ldd namazu.cgiで表示されるlibnmz.so.0.1やlibintl.so.8.1などを/var/www/usr/local/lib等にコピーすればperlが動かなくても大丈夫みたい
- 次はブラウザからnmz.cgiを呼び出すがインデックスファイルが見つからないと
- nmz.cgi内でNMZ_INDEXを指定するがうまくいかない

時間切れ

- いろいろ試した結果/etc/namazu/namazurcを
/var/www/etc/namazu/namazurcにコピーしここでインデックスファイル
のあるディレクトリを指定すると何とかできた
- が、以前の環境で作ったインデックスファイルのため検索はされるが、
元のファイル(メール)を読もうとするとエラーに
- やっぱり今の状況でインデックスファイルを作成する必要ありそう
- と、ここで時間切れ

次の目標

- 何とかML検索ができるようにしたい
 - もうちょっとだと思うけど...
- あとはmnewsを使いたい
- こっちもいろいろしたが、まずMakefileが現在のGNU makeではない昔のmake形式になっていてmakeが通らない
- あと画面制御(?)もtermcap/terminfoからncursesに?
- ChatGPTに聞きながらいろいろ試してmnewsがコンパイルできたが全く動かない



おまけ

fuguitaのアップデート

- 今回最初にインストールしたもの
 - fuguita-7.8-amd64-202511171.img
- いろいろ設定しているうちに202512111がリリース
- 河豚板ガイドを参考にアップデートをしてみよう

fuguitaのアップデート2

- 最初は失敗例
- wgetをpkg_addでインストールしたあと<https://jp1.dl.fuguita.org>からFuguita-7.8-amd64-20251211.iso.gzをダウンロード
- usbfdmでsyncしたあと再起動
 - サイズが大きい(約400MB)ので結構時間がかかる
- モード0で起動
- Fuguita-...がない
- モード0でFuguitaをgetする必要あり

fuguitaのアップデート3

- モード0で起動してFuguIta...をgetしたあとは簡単
- 河豚板ガイドラインに沿ってfiupdate 202512111でアップデート
- 再起動していつものモード3で設定を読み込み
- # cat /usr/fuguita/versionで7.8-amd64-202512111になっていることを確認

今回の感想

- 設定ファイル等ChatGPTに聞きながらしたが、かなり振り回された
- まずはマニュアルを読むか、実際に同じような設定をした人が書いたページを探した方が確実？
- あまりFugulItaの話じゃなくて済みません