

FreeBSD カスタマイズ

1 環境って？

たとえ同じ計算機を使う場合でも、人によっては使いにくいと感じる人もいれば使いやすいと感じる人もいる。また、「ここがどうしても気に入らん」とか、「ここがこうなればもっと使いやすいのに」とか思うこともよくある。そう思ったときに計算機を自分の使いやすいようにして、自分のやり方で計算機を使っていくことができる(できない OS もあるが)。

計算機を使う時の主な動作として、

- キーボード、マウスを使う
- 画面をみる
- アプリケーションプログラムを使う

が挙げられる。

この各動作を「自分が働きやすい」ようにすることを環境整備という。たとえば、

- コマンド名を自分の好きなものにしたい
- 壁紙を変えたい
- ウィンドウの色・デザインを変えたい
- ログイン時に自動でアプリを起動したい

なんか手始めに行う環境整備ではなからうか。

1.1 環境設定の前に

UNIX 系の OS では設定ファイルを書き換えることにより環境設定を行う。設定ファイルは自分のホームディレクトリで

```
\% ls -a
```

と打ち込んでコマンドを実行させると表示される。-a オプションをつけると通常表示されない、最初の文字が「。」(ドット)で始まる名前のファイル(ドットファイル)も表示される。このドットファイルを環境設定用のファイルのファイル名として使うというのが UNIX での約束になっている。

また、環境設定を行う際は、エディタでドットファイルを編集することになるので、エディタの使い方もマスターしておいてほしい。

1.2 ドットファイルの種類

ホームディレクトリにあるドットファイルのうち、変更する可能性があるのは主に以下のファイル。

- .cshrc** csh,tcsh を起動する際に読み込まれる
- .tcshrc** .cshrc よりも優先的に読み込まれる
- .xsession** xdm でログインするときに読み込まれる
- .steprc** ウインドウマネージャafterstep を起動する際に読み込まれる
- .emacs** Mule,Emacs を起動する際に読み込まれる
- .aliases** エイリアスを書いておく
- .forward** メールの転送先を書いておく
- .signature** メールの最後のシグネチャを書いておく
- .im/** IM(Internet Message) の設定ファイルがあるディレクトリ
- .xinitrc** startx で X を立ち上げるときに読み込まれる

1.3 現在の環境は？

いきなり環境設定を始める前に、現在の設定がどうなっているかを調べる方法についてまとめておく。

方法 1 「ls -a」コマンドでドットファイルを調べて、各ドットファイルを more や less 等のコマンドで見る。

方法 2 set,printenv,stty コマンド等で基本的な設定の値が表示されるのを確認する。

2 シェルの環境設定

2.1 シェルとは

OS の機能を実際に実現するプログラムのことをカーネルと呼ぶ。UNIX の場合、ユーザは直接カーネルのプログラムを実行するようにはなっていない。カーネルとユーザの間には、ユーザからの命令を受け取る作業を専門に行う機構が用意されており、さらにその命令を解釈してカーネルに渡して実行してもらう。

2.2 シェルの種類

シェルには何種類もあり、新 B4 の最初の設定ではログインシェルは tcsh になっている。

Bourn シェル UNIX に標準で備わっている。/bin/sh が実行形式

C シェル UNIX に標準で備わっている。/bin/csh が実行形式

tcsh csh の機能拡張版。/usr/local/bin/tcsh が実行形式

bash,zsh その他のシェル

シェルを変えるにはシェル名を打てば変わる。もし、ログインシェル自体を変えるときは

```
%chsh login name
```

とうち、出てきた画面を編集すればよい。

2.3 環境変数とシェル変数

シェルはそれ自身の動作を定義する変数を持っている。この変数はシェル変数と呼ばれるもので、シェルの中だけで使用される。この他に、UNIXシステムのコマンド全般に影響を与える変数として環境変数と呼ばれるものもある。なお、シェル変数は小文字、環境変数は大文字を用いるのが一般的である。

シェル変数の値を表示するには *set* コマンドを実行することにより、左側にシェル変数、右側に設定されている値が表示される。任意のシェル変数だけを参照するには

```
% echo $cwd
```

の様に、シェル変数の前にメタ・キャラクタ\$をつけて指定することで参照できる。

同様に、環境変数の値を表示するには *env*, *printenv*, *setenv* 等を用いる。任意の環境変数の値を表示したい場合もシェル変数と同様に行う。

2.4 .cshrc の設定

快適にシェルを使うために、*tcsh* ではドットファイル「.cshrc」を変更する。何も変更していなければ

```
#
source /usr/local/share/skel/.cshrc-skel
alias rm rm -i
alias cp cp -i
alias mv mv -i
alias rlogin slogin

if (-r $HOME/.aliases) then
    source $HOME/.aliases
endif
```

となっているはずである。「#」はコメントアウトである。2行目では別のファイル */usr/local/share/skel/.cshrc-skel* を読み込んでいる。

変更したい変数はたいていデフォルト設定ファイル中で設定されているが、ホームディレクトリの *.cshrc* で設定しておけば上書きされる。

こまかい設定などは

<http://www.db.is.kyushu-u.ac.jp/fuki/mame.html>

などに詳しい。

2.5 エイリアスの設定

普段良く使うコマンド (+オプション) は打つのに長すぎていやだ、いう場合は、エイリアス (別名) 機能を使うと便利。

```
alias kterm 'kterm -km euc -sb -fn tiny ' #ktermがスクロールするようになる
alias mew 'mule -f mew '#mewが起動
alias bif 'ssh -f minako xpbiff'
```

```
alias ls 'ls -F'
alias la 'ls -a'
alias ll 'ls -l'
```

```
alias lll 'll |less'
alias lsl 'ls |less'
alias lal 'la |less'
alias rm 'rm -i'
```

なお、「*alias*」というコマンドで、現在自分が設定しているエイリアスの一覧が見れる。ホームディレクトリの`.alias`というファイルがあればこれを読むようになっているのでファイルをつくり書き込んでおいた方が良い。

3 .forward

メールを転送させたいときに記述する。

```
\kubo
kubo@csce.kyushu-u.ac.jp
```

書き方を間違えると悲惨なことになるので注意して書くこと。携帯などに転送すると便利。

4 ウィンドウシステムの環境設定

4.1 ウィンドウシステム

*UNIX*では *X Window* とウィンドウマネージャを使って様々なウィンドウを同時に複数開いて作業ができるようになっている。しかも、その見た目やマウス操作、キー操作、色等、ほとんどの部分を自分好みに変えることができる。

Window Manager は何でも使用して良い。現在は *Afterstep* で統一しているがどう変更しても自由である。他に *fvwm*, *fvwm2*, *qwm*, *enlightenment* などなどたくさんそろっている。

4.2 Install

ウィンドウマネージャのインストールは
`/usr/ports/x11-wm/`
内に入っている。インストールしたいウィンドウマネージャのディレクトリに入って (例えば *fvwm2* 等)
`make;makeinstall`
とすればよい。次にこのウィンドウマネージャを起動するには今動作中のウィンドウマネージャを停止させ新しいウィンドウマネージャを起動させなければならない。

```
kubo@kago[~]%ps aux | grep enlightenment
kubo    306  0.1  4.4 7660 5480  v0  S    5:39PM  0:15.81 enlightenment
kubo@kago[~]%kill 306
kubo@kago[~]%afterstep &
```

これで *enlightenment* が停止され、*afterstep* が起動した。

4.3 X 起動時の設定ファイル

コンソール (計算機の前) からログインする場合は */.xinitrc* を読み込むようになっている。これは元になるものがすでに入っているのでコピーする。すでにある人はコピーする必要はない。

```
%cp /usr/X11R6/lib/X11/xinit/xinitrc /.xinitrc
```

.xinitrc の中には、ログイン時に起動しておきたいウィンドウマネージャやクライアントを記述している。例として、壁紙 (本当は *root window* と呼ぶ) を変えてみたり、時計を表示したり。

```
#!/bin/sh
# $XConsortium: xinitrc.cpp, v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

# merge in defaults and keymaps

if [ -f $sysresources ]; then
    xrdb -merge $sysresources
fi

if [ -f $sysmodmap ]; then
    xmodmap $sysmodmap
fi

if [ -f $userresources ]; then
    xrdb -merge $userresources
fi

if [ -f $usermodmap ]; then
    xmodmap $usermodmap
fi

# start some nice programs

afterstep &
```

```
xclock -geometry 50x50-1+1 &
kterm -sb -geometry 60x20+451+0 &
kterm -sb -geometry 60x15+451+355 &
exec kterm -sb -geometry 60x20+0+0 -name login
```

注意しておかないといけないのは、最後のログインシェル以外の記述には、行末に「&」が必要であること。*X Window* は最後に指定したプロセスが終了したときにログアウトするようになっている。よって、それ以前に起動したクライアントはバックグラウンドにまわして、次のクライアントを起動しなければならない。

4.4 .afterstep

アプリケーションの起動をショートカットキーで行なうと大変便利。*Windows* などでは複数のキーを組み合わせることが多いが、せっかく *AfterStep* を使っているのだから、思い切って一発起動にしよう。例えばルートウィンドウで「k」を押すと *kterm* が開く。

```
Key k R N Exec "kterm" exec kterm &
```

ただし、ちゃんとルートウィンドウを指定している状態でないと動かない。そんなときは他のキー (例えば「Alt+k」等) と指定したりすることで回避できる。

```
Key k A M Exec "kterm" exec kterm &
```

マウスを使わずにウィンドウを移動させたり、ウィンドウの大きさを変更できるのも便利。「Ctrl + カーソルキー」で仮想画面の移動、「Alt + カーソルキー」でウィンドウの移動、「Alt + Ctrl + カーソルキー」でウィンドウの大きさを変更ができる。これらはすでに記述されていたりするが自分好みにカスタマイズするのも良いだろう。

```
Key Left A C Scroll -100 +0
Key Right A C Scroll +100 +0
Key Up A C Scroll +0 -100
Key Down A C Scroll +0 +100
```

```
Key Left A M Function "Shade-or-Raise"
Key Right A M Function "Shade-or-Raise"
Key Up A M Function "Shade-or-Raise"
Key Down A M Function "Shade-or-Raise"
```

```
Key Left A CM Function "Resize-or-Raise"
Key Right A CM Function "Resize-or-Raise"
Key Up A CM Function "Resize-or-Raise"
Key Down A CM Function "Resize-or-Raise"
```

ここではキーバインドのみに関して記述しているが *afterstep* のいろいろな部分に関してカスタマイズは可能なので調べてみるとよい。

5 その他

5.1 エディタの環境設定

mule のキー操作も変更可能。 */.emacs* 中で *load* している各ファイルを良く見て変更すれば良い。しかし、*mule* の設定ファイルは *emacs lisp* という、特殊な言語で記述されているため、要注意。

5.2 gkrellm

現在のマシンの状態などが見れるツール。 *Ports* では */usr/ports/sysutils/gkrellm* に入っている。結構便利。

5.3 設定変更時の注意

以上、様々な設定箇所を述べたが、これらを一度に変更してしまったあとにログアウトして、次にはログインできないということがよくある。その場合どこが問題かも分からなくなってしまうので、少しずつ変更していくようにするべし。また、研究室の他の人のドットファイルを覗いてみて、まねしてみるのもいいかも。

5.4 参考

1. とりあえず *man <キーワード>*, コマンド *-help* してみる。
2. そちら辺の本 (*UNIX USER*, *UNIX MAGAZINE*) なんかを漁る
3. *goo* なんかで探すと親切な人による解説があるはず
4. 研究室の人に聞いてみる