

誰にでも使える Unix 講座

第 5 回

「.cshrc の世界」

安岡孝一

yasuoka : root さん、root さん！
root : 何だい？
yasuoka : ginkaku が返事しなくなっちゃったんです。
root : またかい？ どれどれ。
■
うーん。コントロール Q は押してみたかい？
yasuoka : はい。
root : コントロール S じゃないとすると。
who■
おや、字は出るね。
yasuoka : ええ。でもいつもの % が出ないんです。
root : ふうん。
who (ぼこ)
■
じゃあ、コントロール C だな。コントロール C っと。
(Interrupt -- one more to kill letter)
■
もう 1 回、コントロール C っと。
% ■
よし、うまくいった。
yasuoka : そのコントロール C って何ですか？
root : 実行中のプログラムに割り込みをかけるんだよ。ま、たいていのプログラムはこれで止まる。ただこれは BSD の場合で、System V ではデリートキーってことが多いね。

```
% history (ぼこ)
1 set history=10
2 alias rm 'rm -i'
3 Mail
4 ls
5 pushd Mail/Ochi
6 ls
7 mv ~/tmp1 from6
8 vi to6
9 Mail ochi to6
10 history
% ls (ぼこ)
from1 from3 from5 to1 to3 to5
from2 from4 from6 to2 to4 to6
% ■
```

見事だ。<を忘れてる。

yasuoka : え？
root : Mail ochi < to6 とすべきところを見事に <を忘れてる。
yasuoka : あ、ほんとだ。

```
% Mail ochi < to6 (ぼこ)
% ■
```

でも、どうして何も返事しなくなったんですか？

root : ファイル名の指定がないから、メールをキーボードから入力してくれるのをじっと待ってたんだね。たぶん、ホームディレクトリに dead.letter っていうのができてから、消しといた方がいいよ。

yasuoka : はい。
% pushd ~ (ぼこ)
~ ~/Mail/Ochi ~
% ls (ぼこ)
Mail dead.letter sign
% rm dead.letter (ぼこ)
rm: remove dead.letter? y (ぼこ)
% ■

ところで、今日はワイルドカードってのを教えてくれる約束じゃなかった

ですか？

root : ああ、そうだったね。じゃ、この前の~のおさらいも含めて、ファイル名の指定で手を抜く方法を教えよう。

~	ホームディレクトリ
~ユーザ ID	指定されたユーザのホームディレクトリ
*	ヌルを含めていかなる文字列ともマッチ
?	いかなる 1 文字ともマッチ
[複数の文字]	[] 中のいかなる 1 文字ともマッチ
[文字-文字]	2 つの文字の間のいかなる 1 文字ともマッチ
{文字列,文字列,...}	各文字列とマッチ

yasuoka : どうやって使うんですか？

root : ま、そうあせらずに。

```
% ls (ぼこ)
Mail  sign
% █
```

例えばここで cd Mail とする代わりに

```
% cd M* (ぼこ)
% pwd (ぼこ)
/home/yasuoka/Mail
% █
```

とすればいい。M*ってのは「Mから始まるものなら何でも」って意味なんだ。

yasuoka : へーえ。

root : 他にも

```
% cd (ぼこ)
% ls (ぼこ)
Mail  sign
% cat ??g? (ぼこ)
```

```
+-----+
! Koichi YASUOKA !
+-----+
```

% █

なんてのもできる。これは「4文字でなおかつ3文字目がgのものなら何でも」って意味だね。

yasuoka : なーかなか。

root : あるいは

```
% popd (ぼこ)
~/Mail/Ochi ~
% ls (ぼこ)
from1  from3  from5  to1    to3    to5
from2  from4  from6  to2    to4    to6
% file to[3-5] (ぼこ)
to3:   ascii text
to4:   ascii text
to5:   ascii text
% █
```

なんてのは、file to3 to4 to5と同じ意味だ。

yasuoka : すごい。

root : まあ、こうやって手を抜くわけだけど、ワイルドカードなんかがある風には展開されるかは、echoで見るとよくわかる。

```
echo 文字列
文字列を出力する。文字列は複数書いてもよい。
```

yasuoka : echo？

root : まあ、ちょっと見ててごらん。

```
% pushd .. (ぼこ)
~/Mail ~/Mail/Ochi ~
% ls (ぼこ)
Hama          Ochi          Takahash
% █
```

ここでたとえば cd * とすると

```
% cd * (ぼこ)
*: Ambiguous.
% █
```

って叱られちゃうんだけど、これはどうしてかな、って調べられる。

```
% echo * (ぼこ)
Hama Ochi Takahash
% █
```

yasuoka : どういうことですか？

```
root : echo * としたら、Hama Ochi Takahash と表示されたる？
yasuoka : はい。
root : これは、ここでは cd * は cd Hama Ochi Takahash という意味にとられるってことだ。
yasuoka : ふーん、そういう意味なんですか。
```

```
% echo *i (ぼこ)
Ochi
% cd *i (ぼこ)
% pwd (ぼこ)
/home/yasuoka/Mail/Ochi
% █
```

ところで、root さん。

```
root : 何だい？
yasuoka : login したときにいちいち set history=10 とか、alias rm 'rm -i' とかするの面倒くさいんですけど、何かいい方法ありませんか？
root : そうだな。 .cshrc にでも書いておいたら？
yasuoka : .cshrc って何ですか？
root : まずはホームディレクトリへ行ってごらん。
yasuoka : はい。
```

```
% cd (ぼこ)
% ls (ぼこ)
Mail    sign
% █
```

root : そこで ls -a.

yasuoka : ls -a ですか。

```
% ls -a (ぼこ)
.      ..      .cshrc  Mail    sign
% █
```

あれ、ファイルが増えた。

```
root : . から始まるファイルはいつもは見えないんだよ。
yasuoka : へーえ。
root : で、 .cshrc を cat してみてください。
yasuoka : はい。
```

```
% cat .cshrc (ぼこ)
set path=(/usr/local/bin /usr/ucb /bin /usr/bin .)
set prompt='% '
stty erase ^H
stty kill ^U
stty intr ^C
stty quit ^\
stty eof ^D
% █
```

何ですか？ これ。

```
root : yasuoka さんの ID を作った時に、僕が書いておいたものだよ。この .cshrc に書いてあるコマンドが、login した時に自動的に実行されるんだ。
yasuoka : へーえ。
root : もうちょっと詳しくいうと、login した時にはホームディレクトリの .cshrc と .login が、logout した時には .logout が、それぞれ実行される。
yasuoka : じゃあ、 .cshrc に set history=10 とか alias rm 'rm -i' とか書いておけば、いちいち手で実行しなくていいんですね。
root : そうそう。
yasuoka : じゃ、早速。
```

(間)

yasuoka : できました。

```
% cat .cshrc (ぼこ)
set path=(/usr/local/bin /usr/ucb /bin /usr/bin .)
set prompt='% '
set history=10
stty erase ^H
stty kill ^U
stty intr ^C
stty quit ^\
```

```
stty eof ^D
alias rm 'rm -i'
alias his 'history'
% █
```

これでいいんですか？

root : あ、his も加えたんだね。じゃあ、source .cshrc を実行してごらん。

```
source ファイル名
      ファイルに書かれたコマンドを実行する。
```

yasuoka : はい。

```
% source .cshrc (ぼこ)
% █
```

これでどうなったんですか？

root : .cshrc が実行されたから、his が alias されたはずだよ。

yasuoka : はあ。

```
% his (ぼこ)
29 cd *i
30 pwd
31 cd
32 ls
33 ls -a
34 cat .cshrc
35 vi .cshrc
36 cat .cshrc
37 source .cshrc
38 his
% █
```

ほんとだ、alias されてる。でもどうして.cshrcって名前なんですか？

root : て、言うतो？

yasuoka : いえ、login した時に.login、logout した時に.logout を実行するってのはわかるんですけど、どうしてそれ以外に.cshrcってのも実行するんですか？

root : うん、今まで黙ってたけど、それは yasuoka くんが C シェルを使ってるからだよ。

yasuoka : C シェル？

root : そう、csh と書いて C シェル。yasuoka くんが打ち込んだコマンドを、Unix 本体に渡してくれるコマンドインタプリタだよ。

yasuoka : へーえ。

root : Unix ではそういうコマンドインタプリタが login した時に立ち上がって、そのあとずっと面倒を見てくれるんだ。

yasuoka : C シェルの他にもコマンドインタプリタはあるんですか？

root : うん、いくつかね。でもまあ、C シェルがわりと使いやすいし、普及してるからね。

yasuoka : そうなんですか。

```
% cat .cshrc (ぼこ)
set path=(/usr/local/bin /usr/ucb /bin /usr/bin .)
set prompt='% '
set history=10
stty erase ^H
stty kill ^U
stty intr ^C
stty quit ^\
stty eof ^D
alias rm 'rm -i'
alias his 'history'
% █
```

ところで話は変わるんですけど、この.cshrc の中に書いてあるのは全部コマンドですよ。

root : ああ、そうだよ。

yasuoka : この set path=(/usr/local/bin /usr/ucb /bin /usr/bin .) っていうのは、どういう意味なんですか？

```
set path=(ディレクトリ名 ディレクトリ名 ...)
      コマンドパスを設定する。
set cdpath=(ディレクトリ名 ディレクトリ名 ...)
      cd 用のパスを設定する。
```

```
set prompt='文字列'  
    コマンド入力待ち時に表示する文字列を設定する。  
set history=数  
    history で残しておくコマンドの回数を設定する。  
set savehist=数  
    セッション終了時に次のセッションまで残しておくコマンドの回数を設定する  
    (残しておくコマンドは ~/.history ファイルに蓄えられる)。
```

```
root :   例えば yasuoka くんが ls を実行したとするだろ。  
yasuoka : はい。  
root :   でも実際には ls のプログラムの入ったファイルは /bin の中にあるんだ。  
yasuoka : そうなんですか。  
root :   そこで ls をどこに探しに行くかの一覧表がある。それがパスなんだ。この  
        場合は /usr/local/bin、 /usr/ucb、 /bin、 /usr/bin、 . の順に探すわけだ。  
yasuoka : . って何ですか？  
root :   カレントディレクトリ、つまりこのディレクトリって意味だよ。  
yasuoka : わかりました。  
root :   これとよく似たのが cd パスだ。こっちはよく set cdpath=(. ...) なんて  
        風に設定する。  
yasuoka : そうしておくとうどうなるんですか？  
root :   やってみせようか？  
        % set cdpath=(. ...) (ぼこ)  
        % pwd (ぼこ)  
        /home/yasuoka  
        % cd ochi (ぼこ)  
        /home/ochi  
        % pwd (ぼこ)  
        /home/ochi  
        % █  
        つまり、 cd とか pushd とかした時に 1 つ外のディレクトリまで探しに  
        行ってくれるから、結果的に隣のディレクトリに移れる。  
yasuoka : 便利ですね。で、また .cshrc に戻りますけど、この stty ってのは？  
root :   特殊キャラクタの指定だよ。
```

```
stty 機能 キャラクタ  
    特殊キャラクタの設定をおこなう。  
機能としては以下のものがある。  
    erase   1 文字消去  
    kill    1 行消去  
    intr    割り込み  
    quit    強制終了  
    eof     エンドオブファイル (入力終了)
```

```
yasuoka : 特殊キャラクタって？  
root :   コマンドを打ち間違えた時、いつもどうしてる？  
yasuoka : バックスペースで戻してます。  
root :   そのバックスペースが 1 文字消去だったのを設定してるのが、この stty  
        erase ^H だよ。  
yasuoka : え？  
root :   バックスペースはコントロール H と同じコードが出るんだ。そして stty で  
        は、^H はコントロール H を表してる。  
yasuoka : へーえ。するとこの stty intr ^C ってのは？  
root :   今日最初にやったら？ 実行中のプログラムに割り込みをかける特殊キャラ  
        クタだ。  
yasuoka : ああ、そういえば。コントロール C でしたね。  
root :   次の stty quit ^\\ は、コントロール \ でプログラムを強制終了させるっ  
        て意味だよ。  
yasuoka : コントロール \ だけ書き方が違うんですか？  
root :   まあね。他にはデリートが ^_、エスケープが ^\[ だ。  
yasuoka : へーえ。でも強制終了と割り込みってどう違うんですか？  
root :   強制終了の方が強くなってることになってる。でも core っていうファイルを残  
        すんで、注意が必要だよ。さて、もうそろそろ終わりに近づいてきたな。  
yasuoka : そうですか。お名残り惜しいですね。  
root :   じゃ、Unix で一番大事なコマンドを教えて、終わりにしよう。
```

```
man コマンド名  
    コマンドに関するマニュアルを出力する。
```

```
yasuoka : どうもありがとうございました。
```