

# シェルを使おう

## - 日々のシェル -

lilo-bk  
友國 哲男  
(tomokuni@netfort.gr.jp)

### 1.1 ファイル消去

`$ for file in *~; do rm -i $file; done ... ダメ。`

`$ for file in *~; do rm -i "$file"; done ... これでオーケー。`

カレントディレクトリ以下のファイルを消去は、、、2章で

### 1.2 ファイル名変換(大文字入り→小文字)

`$ for file in *; do mv $file `echo $file | tr 'A-Z' 'a-z';` done`

`$ for file in *; \  
> do if test `echo $file | tr -cs 'A-Z' '_' | tr -d '_' != x`; \  
> then mv $file `echo $file | tr 'A-Z' 'a-z';` fi; done`

`$ for file in *; \  
> do if test `echo $file | tr -cs 'A-Z' '_' | tr -d '_' != x`; \  
> -a !-f `echo $file | tr 'A-Z' 'a-z';`; \  
> then mv $file `echo $file | tr 'A-Z' 'a-z';` fi; done`

### 1.2 ファイル名変換(大文字入り→小文字) (2)

GNU 版 test なら `-e (exist)` が使えるので

`$ for file in *; \  
> do if test `echo $file | tr -cs 'A-Z' '_' | tr -d '_' != x`; \  
> -a !-e `echo $file | tr 'A-Z' 'a-z';`; \  
> then mv $file `echo $file | tr 'A-Z' 'a-z';` fi; done`

### 2.1 改行コード変換 (1)

□UNIX → Mac

`$ cat file.unix | tr '\n' '\r' > file.mac`

□Mac → UNIX

`$ cat file.mac | tr '\r' '\n' > file.unix`

### 2.1 改行コード変換 (2)

□DOS → UNIX

`$ cat file.dos | tr -d '\r' > file.unix`

□DOS → Mac

`$ cat file.dos | tr -d '\n' > file.mac`

### 2.1 改行コード変換 (3)

□UNIX → DOS

`$ cat file.unix | awk 'BEGIN{RS="\n";ORS="\r\n"}{print $0}' > file.dos  
$ cat file.unix | perl -ne 's/\r/\n/g; print;' > file.dos`

□Mac → DOS

`$ cat file.mac | awk 'BEGIN{RS="\r";ORS="\r\n"}{print $0}' > file.dos  
$ cat file.unix | perl -ne 's/\r/\n/g; print;' > file.dos`

\*\* RS: (Input) Record Separator  
\*\* ORS: Output Record Separator

### 2.2 拡張子変換

`$ for file in *.htm; do mv $file `basename $file htm`html; done`

bash/zsh なら  
`$ for file in *.htm; do mv $file ${file%*.htm}.html; done`

## 2.3 ファイル消去 (1)

```
$ find . -name '*~' -exec rm {} \;  
$ find . -name '*~' -exec rm -i {} \;
```

## 2.3 ファイル消去 (1)

```
$ find . -name '*~' | xargs rm -i ... ダメ。  
$ find . -name '*~' | xargs -p -r rm ... 空白文字入りファイルはダメ。  
$ find . -name '*~' | xargs -0 -p -r rm ... これでもちよっと。。。  
$ find . -name '*~' -print0 | xargs -p -r rm ... やっぱりダメ。  
$ find . -name '*~' -print0 | xargs -0 -p -r rm ... やっと正解。
```

## 2.3 ファイル消去 (2)

実は GNU 版 find を使えば簡単。^^;

```
$ find . -name '*~' -ok rm -i {} \;
```

## 2.4 ファイルのバックアップ (1)

昨日増えた分を表示

```
$ find Mail -daystart -ctime 1 -print
```

これをバックアップ

```
$ find Mail -daystart -ctime 1 -print | \  
> tar -T - -czf Backup/Mail'date '+%Y%m%d'.tar.gz
```

## 2.4 ファイルのバックアップ (2)

しかしこれではディレクトリが入ってしまうので、

```
$ find Mail -daystart -type f -ctime 1 -print | \  
> tar -T - -czf Backup/Mail'date '+%Y%m%d'.tar.gz
```

空白文字入りファイル対策

```
$ find Mail -daystart -type f -ctime 1 -print0 | \  
> tar --null -T - -czf Backup/Mail'date '+%Y%m%d'.tar.gz
```

## 2.4 ファイルのバックアップ (3)

ログファイルを除く処理も入れる

```
$ find Mail -path 'Mail/procmail.log' -prune \  
> -o -daystart -type f -ctime 1 -print0 | \  
> tar --null -T - -czf Backup/Mail'date '+%Y%m%d'.tar.gz
```

一週間分は、、、

```
$ find Mail -daystart -ctime +0 -ctime -8 ...
```

## 3.1 while と read を使った例

リダイレクトを用いて、

```
$ while read file; do mv ${file} ${file}.bak; done < filelist
```

パイプを使用すると、

```
$ cat filelist | while read file; do mv ${file} ${file}.bak; done
```

## 3.2 for を使った例

```
$ for file in *.c; \  
> do (diff ../orig/${file} ${file} >/dev/null \  
> && echo ${file} is not changed.) \  
> || echo ${file} is changed. ; done | tee diff.log
```

## Acknowledgement

参考にしたもの

Bourne Shell 自習テキスト (<http://www.tsden.org/takamiti/shText/>)  
プロフェッショナルシェルプログラミング (アスキー, ISBN4-7561-1632-9)  
井上さんのページ (<http://www.ainet.or.jp/~inoue/memo/shell.html>)

確認に使わせていただいたシェル

しらいさん作の FD shell (<hp.vector.co.jp/authors/VA012337/soft/fd/fd2.html>)

## 最後に

ご静聴、ありがとうございました。 m(\_o\_)m

次回もご期待ください。^\_^;

おしまい。