

続・誰にでも使える Unix 講座

第 4 回

「core を探せ」

安岡孝一

```

root : yasuoka くん、yasuoka くん。
yasuoka : 何ですか？
root : core 消しといてね。
yasuoka : え？ core って何ですか？
root : プログラムのゲロ。

(間)

root : ま、とにかく login してよ。
yasuoka : はい。

login: yasuoka (ぼこ)
Password: (ばたばたばたばたばたばたぼこ)
Last login: Tue Mar 20 12:27:22 on console
MoonOS 1.1 (GINKAKU) Mon Jan 1 11:11:11 GMT+0900 1990
#####

                DEAREST WALTER, WITH HARD-BOILED EGG.

#####
~% █

root : えっとね、~yasuoka/src/cbanner に大きな core があるから、消しとい
てほしいんだけど…。
yasuoka : src/cbanner ですか？

~% cd src/cbanner (ぼこ)
~/src/cbanner% ls (ぼこ)
Makefile          cbanner.c          cbanner.o          patchfile
cbanner            cbanner.c~         core
~/src/cbanner% █

```

```

あ、この core ってファイルですか？
root : そう。
yasuoka : これ、何かまずいファイルなんですか？
root : 大きさを見てごらん。
yasuoka : 大きさ？

~/src/cbanner% ls -l (ぼこ)
total 139
-rw-r--r--  1 yasuoka      108 Mar 18 18:49 Makefile
-rwxr-xr-x  1 yasuoka    24576 Mar 20 17:34 cbanner
-rw-r--r--  1 yasuoka    12436 Mar 20 15:32 cbanner.c
-rw-r--r--  1 yasuoka    12370 Mar 20 15:32 cbanner.c~
-rw-r--r--  1 yasuoka    14388 Mar 20 17:33 cbanner.o
-rw-r--r--  1 yasuoka   2122554 Mar 20 18:16 core
-rw-r--r--  1 yasuoka      968 Mar 20 14:14 patchfile
~/src/cbanner% █

うわ、でかい。
root : だから、消してね。
yasuoka : はい。

~/src/cbanner% rm core (ぼこ)
rm: remove core? y (ぼこ)
~/src/cbanner% ls (ぼこ)
Makefile          cbanner.c          cbanner.o
cbanner            cbanner.c~         patchfile
~/src/cbanner% █

でも、この core ってファイル、どうして出来たんでしょう？
root : cbanner のプログラムにミスがあって途中で異常終了したか、そうでなけ
ればコントロール\でプログラムを強制終了させたか、そのどっちかだろ
う。
yasuoka : どうしてそういう時に core ってファイルが出来るんですか？
root : ようするにゲロだよ。異常終了した時にプログラムがどういう状態だっ
たか、全て core の中に記録しておくんだ。原因を調べたい人のためにね。
yasuoka : ふーん。でも root さん、僕のディレクトリのここに core ってファイルが
あるって、どうしてわかったんですか？
root : 皆んなのホームディレクトリがそろそろ満杯になってきたからね。見える

```

範囲のディレクトリだけ、find で探したんだよ。

yasuoka : find って？

find ディレクトリ名 条件

ディレクトリ中の全ファイルに対して、条件をチェックする。ディレクトリ名は複数書いてもよい。

条件は以下の通り。

-name 文字列	ファイル名が文字列と一致するとき真 (文字列にワイルドカード使用可)
-type d	ディレクトリのとき真
-type f	ディレクトリでないとき真
-perm 3 桁の 8 進数	モードが「3 桁の 8 進数」のとき真
-user ユーザ ID	持ち主が「ユーザ ID」のとき真
-size 数	512 バイト単位で「数」ブロックの大きさなら真
-links 数	リンク数が「数」のとき真
-atime 数	最終読み出し時刻が「数」日前のとき真
-mtime 数	最終書き込み時刻が「数」日前のとき真
-newer ファイル名	最終書き込み時刻が「ファイル名」以後なら真
-exec コマンド ';'	実行の結果、エグジットステータスが 0 なら真
-ok コマンド ';'	標準入力から y が入力されたなら -exec と同様 (それ以外が入力されれば偽となり実行しない)
-print	ファイル名を標準出力に出力する、常に真
! 条件	条件が偽のとき真
条件 条件	前の条件が真なら、後の条件をチェックする
条件 -o 条件	いずれかの条件が真なら真
'(' 条件 ')'	条件が真なら真、-o のおよぶ範囲を限定する

「数」として、「-数」が指定された場合は「数」未満なら真、「+数」が指定された場合は「数」より大きければ真。「コマンド」中の文字列に「;」が指定された場合は、チェック中のファイル名に置き換えられる。

root : 指定したディレクトリの中にある全てファイルのうち、条件にあったものだけ抜き出すコマンドだよ。ちょっとやってみせようか？

```
~/src/cbanner% find ~/src -name Makefile -print (ぼこ)
/home/yasuoka/src/xor/Makefile
/home/yasuoka/src/kbanner/Makefile
/home/yasuoka/src/lfh/Makefile
```

```
/home/yasuoka/src/cbanner/Makefile
~/src/cbanner% █
```

~/src の中で、ファイル名が Makefile のものだけ標準出力に出力してみたけど、わかるかい？

yasuoka : 出てるのはわかりますけど、find ~/src -name Makefile -print が、どうしてそういう意味になるのかがわかりません。

root : そうか、じゃまずは、find の基本形からやろう。

```
~/src/cbanner% pushd .. (ぼこ)
~/src ~/src/cbanner
~/src% ls (ぼこ)
cbanner kbanner lfh      xor
~/src% find . -print (ぼこ)
.
./xor
./xor/main.c
./xor/Makefile
./xor/pattern.c
./xor/README
./xor/score.c
./xor/mkscene.c
./kbanner
./kbanner/Makefile
./kbanner/README
./kbanner/font.h
./kbanner/font1.c
./kbanner/font2.c
./kbanner/font3.c
./kbanner/kbanner.c
./lfh
./lfh/main.c
./lfh/sub.c
./lfh/Makefile
./lfh/op.c
./cbanner
```

```
./cbanner/cbanner.c
./cbanner/Makefile
./cbanner/patchfile
./cbanner/cbanner.c~
./cbanner/cbanner.o
./cbanner/cbanner
~/src% █
```

これが find の基本。find . -print で、カレントディレクトリの中にあるディレクトリ名とファイル名を、全部出力する。というよりは、カレントディレクトリの中にあるもの全部に対して、それぞれ -print をおこなったのが、これだ。

yasuoka : はい。

root : で、これにさらに条件を加えて

```
~/src% find . -name '*r' -print (ぼこ)
./xor
./kbanner
./cbanner
./cbanner/cbanner
~/src% █
```

r で終わるものだけ -print したのが、これ。カレントディレクトリの中にあるもの全部のうち、-name '\*r' が真になったものだけ、-print がおこなわれる。

yasuoka : それを使って core とか Makefile とかを探したんですね。

root : そう。さらに

```
~/src% !:0- -type f !$ (ぼこ)
find . -name '*r' -type f -print
./cbanner/cbanner
~/src% █
```

とすれば、r で終わるものの中で、ディレクトリじゃないものだけが出力される。

yasuoka : -name '\*r' と -type f の両方が真になったものだけ、-print ですか？

root : 正確に言うと、-name '\*r' が真になったものだけ -type f をチェックして、それも真になったものだけ -print をおこなってるんだ。結果的には同

じ意味だけど、find の条件のチェックは前から順におこなわれるってことは、知っておいた方がいいな。

```
~/src% !-2 -type f -print (ぼこ)
find . -name '*r' -print -type f -print
./xor
./kbanner
./cbanner
./cbanner/cbanner
./cbanner/cbanner
~/src% █
```

こんな風に -print を 2 回書くと、よくわかるだろ？

yasuoka : はい、だいたいのところは。

root : それから find では、ファイルの大きさに条件をつけたり、ファイルの日付に条件をつけたりすることもできる。例えば

```
~/src% find . -size +500 -print (ぼこ)
./kbanner/font1.c
./kbanner/font2.c
./kbanner/font3.c
~/src% █
```

で、大きさが 256001 バイト以上のものだけ選び出せる。

```
~/src% ^-print^-exec wc '{}' ';' ^ (ぼこ)
find . -size +500 -exec wc '{}' ';'
4940 4942 289120 ./kbanner/font1.c
4940 4942 289120 ./kbanner/font2.c
4750 4800 278000 ./kbanner/font3.c
~/src% █
```

ほらね。

yasuoka : どうして -size +500 が 256001 バイト以上って意味になるんですか？

root : 単に -size 500 と書くと、512 バイト単位でちょうど 500 ブロック、つまり 255489 バイト ~ 256000 バイトって意味になるんだ。255488 バイトだったら 499 ブロックで済むし、256001 バイトだったら 501 ブロック要るからね。で、-size +500 は、-size 500 より大きなものを意味する。逆が -size -500 で、255488 バイト以下のものだ。

yasuoka : その後の `-exec wc '{}'` ;' ってのは?  
root : `wc` を実行して、ファイルの行数と単語数と文字数を調べてるんだよ。  
`-exec` では `'{}'` がチェック中のファイル名に置き換えられるからね。こんな芸当もできる。あ、最後の `';` ;' は絶対忘れないようにね。  
yasuoka : はい。  
root : そんなところかな。  
~/src% `popd (ぼこ)`  
~/src/cbanner  
~/src/cbanner% █  
yasuoka : さっき root さん、日付にも条件がつけられるって言ってませんでした?  
root : あ、そうだったね。

```
~/src/cbanner% ls -l (ぼこ)
total 67
-rw-r--r--  1 yasuoka      108 Mar 18 18:49 Makefile
-rwxr-xr-x  1 yasuoka    24576 Mar 20 17:34 cbanner
-rw-r--r--  1 yasuoka    12436 Mar 20 15:32 cbanner.c
-rw-r--r--  1 yasuoka    12370 Mar 20 15:32 cbanner.c~
-rw-r--r--  1 yasuoka    14388 Mar 20 17:33 cbanner.o
-rw-r--r--  1 yasuoka      968 Mar 20 14:14 patchfile
~/src/cbanner% █
```

うーん

```
~/src/cbanner% find . -mtime 1 -print (ぼこ)
./cbanner.c
./patchfile
./cbanner.c~
~/src/cbanner% date (ぼこ)
Wed Mar 21 16:09:14 JST 1990
~/src/cbanner% █
```

これでわかるかな?

yasuoka : わかりません。  
root : 最終書き込み時刻が 24 時間前から 48 時間前までのものを、`-print` したんだよ。  
yasuoka : 最終書き込み時刻って?

root : ファイルに最後に書き込んだ時刻。 `ls -l` で表示されるだろ?

```
ls -lu ディレクトリ名
    ディレクトリ内のファイルの一覧を、最終読み出し時刻等とともに標準出力
    へ出力する。
ls -l ディレクトリ名
    ディレクトリ内のファイルの一覧を、最終書き込み時刻等とともに標準出力
    へ出力する。
ls -lc ディレクトリ名
    ディレクトリ内のファイルの一覧を、最終更新時刻等とともに標準出力に出
    力する。
ls -d ディレクトリ名
    ディレクトリ自身の情報を標準出力へ出力する。
いずれもディレクトリ名が省略された場合は、カレントディレクトリが指定された
    ものとみなす。
```

yasuoka : あ、ファイルの日付って 3 つもあるんですか。

```
~/src/cbanner% ls -lu (ぼこ)
total 67
-rw-r--r--  1 yasuoka      108 Mar 20 17:34 Makefile
-rwxr-xr-x  1 yasuoka    24576 Mar 20 18:16 cbanner
-rw-r--r--  1 yasuoka    12436 Mar 20 17:34 cbanner.c
-rw-r--r--  1 yasuoka    12370 Mar 20 15:37 cbanner.c~
-rw-r--r--  1 yasuoka    14388 Mar 20 17:34 cbanner.o
-rw-r--r--  1 yasuoka      968 Mar 20 15:32 patchfile
~/src/cbanner% ls -lc (ぼこ)
total 67
-rw-r--r--  1 yasuoka      108 Mar 18 18:49 Makefile
-rwxr-xr-x  1 yasuoka    24576 Mar 20 17:34 cbanner
-rw-r--r--  1 yasuoka    12436 Mar 20 15:32 cbanner.c
-rw-r--r--  1 yasuoka    12370 Mar 20 15:32 cbanner.c~
-rwxr--r--  1 yasuoka    14388 Mar 20 17:33 cbanner.o
-rw-r--r--  1 yasuoka      968 Mar 20 14:14 patchfile
~/src/cbanner% █
```

root : 最終更新時刻は `find` では使えないけどね。

yasuoka : 最終更新時刻って?

