

続・誰にでも使える Unix 講座

第 5 回

「親亀こけたら...」

安岡孝一

```

yasuoka : root さん、root さん。
root : あ、リンクの話だったね。
yasuoka : いえ、その前に一つ訊きたいことがあるんですけど。
root : 何だい？
yasuoka : ファイルの内容の引き算がしたいんです。具体的には
~/tmp% ls (ぼこ)
bkup  org
~/tmp% cat bkup (ぼこ)
./Makefile
./main.c
./sub.c
./sub2.c
./op.c
./Makefile~
~/tmp% cat org (ぼこ)
./main.c
./sub.c
./Makefile
./op.c
~/tmp% █
この bkup に入ってて、org に入っていない行、つまり
./sub2.c
./Makefile~
を、この 2 つから取り出したいんですけど。
root : 順番は、元のファイルの行の順番と違っててもいいのかい？
yasuoka : はい。別にどんな順番でもかまいません。

```

root : じゃ、sort して comm -23 だ。

```

comm -23 ファイル名 ファイル名
    前のファイルにあって、後のファイルにない行を、標準出力に出力する。
comm -12 ファイル名 ファイル名
    前のファイルと後のファイルの両方にある行を、標準出力に出力する。
    ファイルはそれぞれ sort されていなければならない。ファイル名として - が指定
    された場合は、標準入力を用いられる。

```

```

root : まずは sort bkup > bkup.sort と sort org > org.sort して。
yasuoka : はい。

```

```

~/tmp% sort bkup > bkup.sort (ぼこ)
~/tmp% !:gs/bkup/org/ (ぼこ)
sort org > org.sort
~/tmp% cat bkup.sort (ぼこ)
./Makefile
./Makefile~
./main.c
./op.c
./sub.c
./sub2.c
~/tmp% cat org.sort (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
~/tmp% █

```

```

root : 次に comm -23 bkup.sort org.sort だ。
yasuoka : comm -23 bkup.sort org.sort ですね。
~/tmp% comm -23 bkup.sort org.sort (ぼこ)
./Makefile~
./sub2.c
~/tmp% █

```

```

root : それが bkup にあって、org にない行だろう？
yasuoka : はい。うーん、なかなか。でも、sort してないといけないんですか？

```

```
root : うん。comm は入力 が sort されてることを前提に作られてるからね。
yasuoka : ふうん。
          ~/tmp% ^23^12^ (ぼこ)
          comm -12 bkup.sort org.sort
          ./Makefile
          ./main.c
          ./op.c
          ./sub.c
          ~/tmp% █
```

これが両方のファイルに入ってる行ですか？

```
root : そう。これで、ファイルの差と交わりは取れるはずだね。
yasuoka : ええ。どうもありがとうございました。
```

(間)

```
root : あ、リンクの話じゃなかったっけ。
yasuoka : そうです。
root : あぶないところだったな。じゃ、今のファイルを例に話そうか。
yasuoka : はい。
```

```
~/tmp% ls -l (ぼこ)
total 4
-rw-r--r-- 1 yasuoka      56 Mar 22 13:11 bkup
-rw-r--r-- 1 yasuoka      56 Mar 22 15:20 bkup.sort
-rw-r--r-- 1 yasuoka      35 Mar 22 13:11 org
-rw-r--r-- 1 yasuoka      35 Mar 22 15:20 org.sort
~/tmp% █
```

この 1 1 1 1 1 っていうのが、各ファイルのリンク数を表してるんですよ。

```
root : そう。
yasuoka : リンク数って何ですか？
root : ディレクトリの場合には、そのディレクトリの中にディレクトリがいくつあるか、だよ。ただし、と.. も、数に入れる。試しに mkdir temp で、ディレクトリを作っごらん。
yasuoka : mkdir temp ですね。
```

```
~/tmp% mkdir temp (ぼこ)
~/tmp% ls -l (ぼこ)
```

```
total 5
-rw-r--r-- 1 yasuoka      56 Mar 22 13:11 bkup
-rw-r--r-- 1 yasuoka      56 Mar 22 15:20 bkup.sort
-rw-r--r-- 1 yasuoka      35 Mar 22 13:11 org
-rw-r--r-- 1 yasuoka      35 Mar 22 15:20 org.sort
drwxr-xr-x 2 yasuoka     512 Mar 22 15:25 temp
~/tmp% █
```

temp のリンク数は 2 になってますよ。

```
root : temp の中は空で、. と.. だけしかないからね。さらに mkdir temp/temp
      してごらん。
yasuoka : はい。
```

```
~/tmp% mkdir temp/temp (ぼこ)
~/tmp% ls -l (ぼこ)
total 5
-rw-r--r-- 1 yasuoka      56 Mar 22 13:11 bkup
-rw-r--r-- 1 yasuoka      56 Mar 22 15:20 bkup.sort
-rw-r--r-- 1 yasuoka      35 Mar 22 13:11 org
-rw-r--r-- 1 yasuoka      35 Mar 22 15:20 org.sort
drwxr-xr-x 3 yasuoka     512 Mar 22 15:26 temp
~/tmp% █
```

```
root : ほらね。
yasuoka : ふうん。じゃあ、ディレクトリ以外は？
root : 普通のファイルの場合には、1 つのファイルが何種類の名前を持つてるか、だよ。
yasuoka : 1 つのファイルが、複数の名前を持つことができるんですか？
root : ln を使えばできる。
```

<pre>ln ファイル名 ファイル名 前のファイルへのリンクを、後のファイルとして作成する。 ln -s 文字列 ファイル名 BSD のみ。文字列を内容とするシンボリックリンクを、指定されたファイル名で作る。</pre>

```
root : 試しに ln org.sort new とごらん。
yasuoka : ln org.sort new ですね。
```

```
~/tmp% ln org.sort new (ぼこ)
~/tmp% ls -l (ぼこ)
total 6
-rw-r--r--  1 yasuoka      56 Mar 22 13:11 bkup
-rw-r--r--  1 yasuoka      56 Mar 22 15:20 bkup.sort
-rw-r--r--  2 yasuoka      35 Mar 22 15:20 new
-rw-r--r--  1 yasuoka      35 Mar 22 13:11 org
-rw-r--r--  2 yasuoka      35 Mar 22 15:20 org.sort
drwxr-xr-x  3 yasuoka     512 Mar 22 15:26 temp
~/tmp% █
```

root : org.sort のリンク数が変わったろ？
yasuoka : はい。
root : じゃ、new を org.sort と較べてごらん。
yasuoka : えっと

```
~/tmp% cat org.sort (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
~/tmp% cat new (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
~/tmp% █
```

あ、おんなじだ。でもこれ、cp org.sort new するのとどう違うんですか？
root : org.sort の末尾に、適当に何か付け加えてごらん。
yasuoka : 付け加える？

```
~/tmp% echo tekito >> org.sort (ぼこ)
~/tmp% cat org.sort (ぼこ)
./Makefile
./main.c
./op.c
```

```
./sub.c
tekito
~/tmp% █
```

これでいいですか？
root : で、cat new してごらん。
yasuoka : はい。

```
~/tmp% cat new (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
tekito
~/tmp% █
```

あれ？ new には何もしてないのに。
root : それがリンクだ。つまり、org.sort と new は中身は同じ 1 つのファイルなんだよ。というよりは、その

```
./Makefile
./main.c
./op.c
./sub.c
tekito
```

というファイルに対して、org.sort っていう名前と new っていう名前の 2 つが付いてるんだ。この名前数が、普通のファイルのリンク数。
yasuoka : すると、1 つのファイルに色んな名前が付けられるんですね。
root : そう。
yasuoka : 同じディレクトリじゃないと、いけないんですか？
root : いや、別のディレクトリにあったってかまわない。
yasuoka : と、すると

```
~/tmp% cd temp (ぼこ)
~/tmp/temp% ln ../new newer (ぼこ)
~/tmp/temp% ls -l (ぼこ)
total 2
-rw-r--r--  3 yasuoka      42 Mar 22 15:30 newer
```

```
drwxr-xr-x 2 yasuoka      512 Mar 22 15:26 temp
~/tmp/temp% cat newer (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
tekito
~/tmp/temp% █
```

うーん、すごい。ここで、この newer に何か書き込んだら、外の new も org.sort も変わりますよね。

```
root : そういこと。
yasuoka : でも、要らなくなったリンクは、どうしたらいいんですか？
root : rm すればいい。
yasuoka : rm したら全部削除されちゃうんじゃ？
root : じゃあ、rm ../new してごらん。
yasuoka : はい。
```

```
~/tmp/temp% rm ../new (ぼこ)
rm: remove ../new? y (ぼこ)
~/tmp/temp% ls -l (ぼこ)
total 2
-rw-r--r-- 2 yasuoka      42 Mar 22 15:30 newer
drwxr-xr-x 2 yasuoka      512 Mar 22 15:26 temp
~/tmp/temp% cat newer (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
tekito
~/tmp/temp% █
```

newer は別に变化ありませんね。

```
root : リンク数が 1 つ減ってる。
yasuoka : あ、ほんとだ。

~/tmp/temp% ls -l .. (ぼこ)
total 5
```

```
-rw-r--r-- 1 yasuoka      56 Mar 22 13:11 bkup
-rw-r--r-- 1 yasuoka      56 Mar 22 15:20 bkup.sort
-rw-r--r-- 1 yasuoka      35 Mar 22 13:11 org
-rw-r--r-- 2 yasuoka      42 Mar 22 15:30 org.sort
drwxr-xr-x 3 yasuoka      512 Mar 22 15:26 temp
~/tmp/temp% █
```

うーん、こういうことか。

```
root : そういこと。rm でリンク数が 0 になったファイルは本当に削除されてしま
yasuoka : だけど、リンクが残ってる限り、ファイル自体は残り続ける。
yasuoka : だいたいわかりました。
root : それから、リンクとよく似たものに、シンボリックリンクっていうのがあ
yasuoka : る。ln -s newer sym してみてよ。
yasuoka : -s が付くんですね。
```

```
~/tmp/temp% ln -s newer sym (ぼこ)
~/tmp/temp% ls (ぼこ)
newer sym temp
~/tmp/temp% cat sym (ぼこ)
./Makefile
./main.c
./op.c
./sub.c
tekito
~/tmp/temp% █
```

リンクとシンボリックリンクは、どう違うんですか？

```
root : まずは ls -l。
yasuoka : はい。

~/tmp/temp% ls -l (ぼこ)
total 3
-rw-r--r-- 2 yasuoka      42 Mar 22 15:30 newer
lrwxrwxrwx 1 yasuoka      5 Mar 22 15:40 sym -> newer
drwxr-xr-x 2 yasuoka      512 Mar 22 15:26 temp
~/tmp/temp% █
```

あれ、モードも何だか変だし、後ろに変な矢印も入ってる。

```
root : newer を rm してから、cat sym してごらん。
```

yasuoka : rm newer で、cat sym ですか。

```
~/tmp/temp% rm newer (ぼこ)
rm: remove newer? y (ぼこ)
~/tmp/temp% cat sym (ぼこ)
sym: No such file or directory
~/tmp/temp% █
```

あれ？

```
~/tmp/temp% ls (ぼこ)
sym      temp
~/tmp/temp% ls -l (ぼこ)
total 2
lrwxrwxrwx 1 yasuoka      5 Mar 22 15:40 sym -> newer
drwxr-xr-x 2 yasuoka     512 Mar 22 15:26 temp
~/tmp/temp% █
```

どうなってるんですか？

root : そこでさらに echo tekito > newer とでもして、cat sym してごらん。
yasuoka : はい。

```
~/tmp/temp% echo tekito > newer (ぼこ)
~/tmp/temp% ls (ぼこ)
newer  sym      temp
~/tmp/temp% cat sym (ぼこ)
tekito
~/tmp/temp% █
```

う、さっぱりわかんない。

root : ようするに、sym は newer っていうファイルを見にいってるだけなんだよ。つまり、ln newer sym とした時には、newer の内容自体を newer と sym とで共有するんだけど、ln -s newer sym とした時には、sym が一方的に newer っていう名前のファイルの内容を見にいってるだけで、親亀の newer がなくなったら sym だけではどうにもならない。これがシンボリックリンクだ。

yasuoka : じゃ、普通のリンクの方がいいんじゃないですか？

root : いや、シンボリックリンクはファイルだけでなく、ディレクトリにも使えるんだよ。例えば ln -s .. oya としてごらん。

yasuoka : はい。

```
~/tmp/temp% ln -s .. oya (ぼこ)
~/tmp/temp% ls -l (ぼこ)
total 4
-rw-r--r-- 1 yasuoka      7 Mar 22 15:52 newer
lrwxrwxrwx 1 yasuoka      2 Mar 22 15:58 oya -> ..
lrwxrwxrwx 1 yasuoka      5 Mar 22 15:40 sym -> newer
drwxr-xr-x 2 yasuoka     512 Mar 22 15:26 temp
~/tmp/temp% █
```

root : で、cd oya。

yasuoka : cd oya ですな。

```
~/tmp/temp% cd oya (ぼこ)
~/tmp% █
```

おや？

```
~/tmp% cd temp (ぼこ)
~/tmp/temp% cd oya (ぼこ)
~/tmp% cd temp (ぼこ)
~/tmp/temp% █
```

うーん、こういうことか。

root : そういうことだ。ただ、シンボリックリンクは BSD でしか使えない。最近では System V でも使えるのがあるみたいだけどね。

yasuoka : わかりました。あ、シンボリックリンクを削除するにはどうしたらいいんですか？

root : rm を使う。ディレクトリへのシンボリックリンクでも、rm で削除する。ファイルへのシンボリックリンクの時には、間違っ親亀の方を消さないようにね。

yasuoka : はい。

root : さて、もうこんな時間だ。大体こんなもんで、リンクについてはわかったかい？

yasuoka : はい、すごくためになりました。どうもありがとうございました。