

#### 4.14.2. リダイレクション -> , >> -

リダイレクションは、前節で取り上げた3つのストリーム（ファイルディスクリプタ番号がわかれば任意のストリームを操作可能です）を本来の流れから逸らしてしまう機能です。実際に逸らす先はファイルになります。まずは出力部分でリダイレクトを行ってみましょう。通常使うリダイレクトは“>”と“<”です。リダイレクト先としてファイルも指定することになります。

```
$ ls -l ←普通にls
合計 4
-rwxrwx--- 1 root    root    0  1月 23 16:31 hoge.pl
-rw-rw-r-- 1 SL6 SL6 28  1月 23 16:13 redhat.hard
lrwxrwxrwx 1 SL6 SL6  6  1月 23 16:01 redhat.sym -> redhat
$ ls -l > log ← “>” を用いて出力先を “log” というファイルに変更
                画面上には ls コマンドの結果は表示されない

$ ls
hoge.pl log redhat.hard redhat.sym
↑新規に “log” というファイルが生成されている

$ cat log
合計 4
-rwxrwx--- 1 root    root    0  1月 23 16:31 hoge.pl
-rw-r--r-- 1 root    root    0  1月 24 10:32 log
-rw-rw-r-- 1 SL6 SL6 28  1月 23 16:13 redhat.hard
lrwxrwxrwx 1 SL6 SL6  6  1月 23 16:01 redhat.sym -> redhat
↑
“ls -l” の結果が入っている
```

このように、画面に出されるところをファイルへすり替えることで、コマンドの出力をファイルに落とすことができます。ただし“>”は標準出力（stdout）しかリダイレクトしません。

そのため、

```
$ ls /root/ > log
ls: /root/: 許可がありません
$ cat log
$
↑ファイルにデータがないため、すぐにプロンプトが返っている。
```

という具合に、リダイレクトされない場合があります。これは、出力されたものが**標準エラー出力 (stderr)** を経由しているからです。これはもちろん、エラーをエラーとしてユーザに伝えるという目的は理にかなったものですが、エラーそのものもリダイレクトしたいという場合もあります。そこで、もうひとつのリダイレクトが登場します。

```
$ ls /root/ 2> log
$ cat log
ls: /root/: 許可がありません
↑エラー出力がファイルに入っている。
```

“2>”を用いることで、stderr をファイルに落とすことができます。これだけでもかなり使い方が増えてくるのですが、ここまでのリダイレクトではリダイレクト先のファイルの中身は消されてしまいます。先ほどの続きで確認してみましょう。

```
cat log
ls: /root/: 許可がありません ←中身がある
$ ls /home/ > log
$ cat log
SL6
↑ “許可がありません” が消えて ls の結果になっている。
```

例えば「出力が毎回異なるアプリケーション」の結果を記録したいときに、繰り返し実行しても最後に実行した結果しか見られないのでは非常に不便です。そこで**追記を行うリダイレクト**が存在します。記号は“>>”です。

```
$ ls /root/ 2> log
$ ls /home/ >> log
$ cat log
ls: /root/: 許可がありません ←最初に実行した ls コマンドのエラー
SL6 ←2番目に実行した ls コマンドの結果
```

標準エラー出力を含めて追記する場合は、以下のように実行します。

```
$ ls /root/ >> log 2>&1 ←2 の出力先を、1 の出力先と同じものに設定する
$ cat log
ls: /root/: 許可がありません
example
ls: /root/: 許可がありません
```

ここまでは出力のリダイレクトでしたが、入力 (stdin) もリダイレクトできます。あらかじめファイルにデータが入っていれば、それを入力として渡します。そのための記号は“<”です。コマンドは標準入力を用いているものであれば大抵のものが利用可能なため、ここでは sort コマンドを使ってみましょう。sort コマンドは、引数として指定したファイルの中身を行単位でソート (並べ替え) して出力します。引数としてファイルを指定しなかった時は標準入力からデータを得ようとしています。

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...
$ sort < /etc/passwd
adm:x:3:4:adm:/var/adm:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
avahi-autoipd:x:100:156:avahi-autoipd:/var/lib/avahi-autoipd:/sbi
n/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
...
```

という具合に、先頭文字のアルファベット順に並べ替えられて出力されていることがわかります。なお、sort コマンドは大文字小文字の区別をしてしまうため、区別しないで並べ替える、数字を数字の大小で並べ替える等の処理には別途オプションが必要になります。詳細はオンラインマニュアルで参照して下さい。