

Unix でのファイルのコピーと移動

—ディレクトリを理解しよう—

小 堀 聡
Satoshi KOBORI

理工学部電子情報学科 教授
Professor, Department of Electronics and Informatics



1. はじめに

この記事では、Unix 環境において、コマンド入力によりファイル进行操作する方法について解説する。

ファイルのコピーや移動は、Windows 環境のような GUI (Graphical User Interface) においてならば簡単にできるが、コマンドを使ってファイルやディレクトリの操作を行うのは難しいと感じる学生諸君が多いようだ。それはディレクトリというものを理解したうえで、コマンドを適切に使用しなければならぬからであると思われる。

そこで、ここでは、順序立ててファイルの操作方法について説明していくことで、ディレクトリについての理解を深められるようにしたいと思う。

以下の例については、必ず Unix 環境にログインして実際に実行しながら、内容を確認していくことにしよう。

2. 作業用ディレクトリの準備

本記事の説明では、カレントディレクトリ (現在のディレクトリ) の場所を特に指定していない。ホームディレクトリ (ログインした直後のディレクトリ) で操作することは可能だが、ここではホームデ

ィレクトリの直下に `work` という名前のディレクトリを作成し、そこをカレントディレクトリにしよう。そうすれば、他のファイルやディレクトリが存在しないので、操作の結果を確認しやすい。

具体的な方法は以下のとおりである。(注: これ以降、`▼` はエンターキーを押すことを表すが、`▼` は表示されない。)

```
$ cd▼(ホームディレクトリに戻る)
$ mkdir work▼(work ディレクトリを作成)
$ cd work▼(work ディレクトリに移動)
```

3. コピーおよび移動の際のファイル名の指定

〈準備と確認〉

カレントディレクトリに `d1`, `d2`, `d3`, `d4` というディレクトリが存在し、`d1` には `aaa` というファイルが存在するという状態にする。もしこれらのファイルやディレクトリがないなら、自分で作成する。逆に、すでに同じ名前のファイルやディレクトリがあるならば、削除するか、名前を変更する (他のファイルやディレクトリが存在しても構わないが、結果の確認の際には注意が必要である)。

〈ファイル名を省略してのコピー〉

コピーの際にファイル名を変更しないならば、コピー先にファイル名は指定（記述）しなくてもよく、ディレクトリ名でだけでよい。次のようにしてみよう。

```
$ cp d1/aaa d2▼
```

これにより d1 にある aaa が d2 にコピーされる。ls -l で確認してみよう。

```
$ ls -l d2▼
```

〈ファイル名を変更してのコピー〉

コピーの際にはファイル名を変更することができる。たとえば、d1 の aaa を d2 へ bbb という名前前でコピーすることができる。次のようにしてみよう。

```
$ cp d1/aaa d2/bbb▼
```

d2 に bbb があることを確認してみよう。

```
$ ls -l d2▼
```

d1 の aaa と d2 の bbb は、ファイル名は異なるが両者は同じものである（cat で同じかどうか確認してみよう）。

〈カレントディレクトリへのコピー〉

カレントディレクトリは「.」で表される。コピー先がカレントディレクトリの場合は、たとえば、以下のようにすればよい。次のようにしてみよう。

```
$ cp d1/aaa .▼
```

カレントディレクトリに aaa があることを確認してみよう。

```
$ ls -l▼
```

〈親ディレクトリへのコピー〉

同様に、コピー先が親ディレクトリ（1つ上のディレクトリ）の場合は、親ディレクトリを示す「..」を用いればよい。このことを試すために、まず、d2 ディレクトリに移動させよう。

```
$ cd d2▼
```

そして、以下のようにすれば、d2 にある bbb が d2 の親ディレクトリにコピーされる。次のようにしてみよう。

```
$ cp bbb ..▼
```

d2 の親ディレクトリに移動して bbb があることを確認しよう。

```
$ cd ..▼  
$ ls -l▼
```

なお、この場合、親ディレクトリに移動せずに

```
ls -l ..
```

としても同じことが確認できる（その方が手数は少ない）が、ここでは親ディレクトリに移動した方が分かりやすいので移動させている。

〈同じレベルの他のディレクトリへのコピー〉

d1 と d2 は同じ親ディレクトリの下にあり、このようなディレクトリは同じレベル（階層）にあるという。もし今 d2 ディレクトリにいる状態で、d2 の bbb を d1 にコピーするにはどうしたらいいだろうか。まず、d2 に移動してみよう。

```
$ cd d2▼
```

ここで、d1 は親ディレクトリの直下にあることから、「../d1」と現すことができる。この表現を使えば、d2 の bbb を d1 にコピーできる。次のようにして d1 に bbb があることを確認してみよう。

```
$ cp bbb ../d1▼  
$ cd ../▼  
$ ls -l d1▼
```

〈ファイル名を省略してのファイルの移動〉

移動に関してもまったく同様である。以下の例のとおりにして、d1のaaaをd3に移動させ、aaaが移動したことを確認しよう（d1にはないことも確認する）。

```
$ mv d1/aaa d3▼  
$ ls -l d3▼  
$ ls -l d1▼
```

〈ファイル名を変更しての移動〉

以下の例のとおりにして、d2のaaaをcccという名前にしてd3に移動させ、そのことを確認してみよう（d2にはないことも確認する）。

```
$ mv d2/aaa d3/ccc▼  
$ ls -l d3▼  
$ ls -l d2▼
```

〈カレントディレクトリへのファイルの移動〉

以下の例のとおりにして、d3のcccをカレントディレクトリに移動させ、cccがあることを確認しよう。

```
$ mv d3/ccc .▼  
$ ls -l▼
```

〈親ディレクトリへのファイルの移動〉

以下の例のとおりにして、d4のdddをd4の親ディレクトリに移動させ、dddがあることを確認しよう。

```
$ cp d3/aaa d4/ddd▼(ファイルの準備)  
$ cd d4▼  
$ mv ddd ../▼  
$ cd ../▼  
$ ls -l▼
```

〈同じレベルの他のディレクトリへの移動〉

以下の例で、d3にあるaaaを同じ階層にあるd4に移動させ、aaaがあることを確かめてみよう。

```
$ cd d3▼  
$ mv aaa ../d4▼  
$ cd ../▼  
$ ls -l d4▼
```

〈解説：mvでファイル名を変更できる理由〉

mvはmoveという語の省略形であることから分かるように、本来は「移動」のためのコマンドである。しかし、mvはファイル名を変更する際にも用いる。

このことは〈ファイル名を変更しての移動〉の特別な場合と考えることができる。

まず、ファイル名を変更しての移動の通常の場合

```
mv d1/aaa d2/bbb
```

というようになるが、ここで移動先が同じディレクトリならば、

```
mv d1/aaa d1/bbb
```

となり、d1のaaaが消えてd1に同じ内容のbbbが現れる。結果的にaaaという名前がbbbに変更されたことになる。

一方、コピー先が同じディレクトリの場合は、

```
cp d1/aaa d1/bbb
```

となり、d1にはaaaが残ったままd1に同じ内容のbbbが現れる。これは単なるコピーである。

このようなcpとmvの役割の違いをしっかりと

理解しておこう。

4. コピー先や移動先に 指定したディレクトリがない場合

〈準備と確認〉

カレントディレクトリに `aaa`, `bbb`, `ccc`, `ddd` というファイルが存在する状態にする。もしこれらのファイルがないなら、自分で作成する。また、`d5`, `d6` というディレクトリが存在しないことを確認する。もしすでにこの名前のディレクトリが存在するならば、削除するか、名前を変更する（他のファイルやディレクトリが存在しても構わないが、結果の確認の際には注意が必要である）。

〈指定したディレクトリがない場合のコピー〉

ファイル名を変更しないならば、コピー先のディレクトリ名だけを記述すればよいが、もしその名前のディレクトリが存在しなかったらどうなるだろうか。次のようにしてみよう。

```
$ cp aaa d5▼
```

こうやってもエラーメッセージは出ない。何が起きたのだろうか。 `ls -l` で確認してみると、`d5` というファイル（ディレクトリではなく）が作られていることが分かる。つまり、`aaa` が `d5` という名前のファイルにコピーされた。コンピュータ（シェル）は、`d5` というのはファイル名だと解釈してコピーしたのである。

また、もし `d5` という名前のファイルが先に存在していたとしても、シェルのやることは同じである。たとえば、`cp bbb d5` とすれば、`bbb` というファイルが `d5` に上書きされるだけである（実際に確認してみよう。本学のシステムでは上書きの際に確認のメッセージが出るが、これには「`y`」を入力する）。

ここで、`d5` は後の操作にじゃまなので、削除しておこう（確認のメッセージには「`y`」を入力する）。

```
$ rm d5▼
```

では、`d5` がディレクトリであることを知らせるならば、どうなるだろうか。ディレクトリ名の後ろに「`/`」を付けることで、それがディレクトリであることを示すことができる。次のようにしてみよう。

```
$ cp aaa d5/▼
```

こうすると、`d5` というディレクトリはないため、エラーメッセージが出て、ファイルのコピーは行われない。

では、ちゃんとコピーされるように、以下のようにして、ファイルがコピーされることを確認しよう。

```
$ mkdir d5▼  
$ cp aaa d5/▼  
$ ls -l d5▼
```

以上のことから分かるように、ディレクトリにコピーする場合は、ディレクトリ名をよく確認することが必要である。しかし、勘違いやタイプミスの可能性はあるので、ディレクトリ名の後ろに「`/`」に付けておけば、間違ったコピーがされてしまうことは防げる。

〈指定したディレクトリがない場合の移動〉

移動に関してもまったく同様である。以下の例のとおりにして、`ccc` が `d6` というファイル名に変更されることを確認しよう。

```
$ mv ccc d6▼  
$ ls -l▼
```

ここで、`d6` は後の操作にじゃまなので、削除しておこう（確認のメッセージには「`y`」を入力する）。

```
$ rm d6▼
```

次に、以下の例のとおりにして、エラーメッセー

ジが出ることを確認しよう。

```
$ mv ddd d6/▼
```

そして、以下の例のとおりにして、ddd がディレクトリ d6 に正しく移動することを確認しよう。

```
$ mkdir d6▼  
$ mv ddd d6/▼  
$ ls -l d6▼
```

〈ディレクトリごとのコピーや移動〉

一方、ディレクトリごとコピーしたり移動させたりする場合は、コピー先や移動先にはディレクトリが存在している必要はない。教科書や参考書などの例においても存在しないことを前提に記されている。

5. コピーや移動の際のファイルの上書き

〈準備と確認〉

カレントディレクトリに aaa というファイル、d7 と d8 というディレクトリが存在し、d7 と d8 にはどちらにも aaa というファイルが存在するという状態にする。もしこれらのファイルやディレクトリがないなら、自分で作成する。逆に、すでに同じ名前のファイルやディレクトリがあるならば、削除するか、名前を変更する（他のファイルやディレクトリが存在しても構わないが、結果の確認の際には注意が必要である）。

〈コピー先にコピー元のファイルと同じ名前のファイルがある場合〉

カレントディレクトリとディレクトリ d7 にはどちらも同じ名前の aaa というファイルがある。カレントディレクトリの aaa を d7 にコピーしたらどうなるか、確認してみよう。

```
$ cp aaa d7/aaa▼
```

「上書きしてよいか」のメッセージのメッセージ

が出てくるので、「y」を入力する。

同じファイル名なので分かりにくい为上書きされている（cat で内容を確認してみよう）。つまり、コピー先に元々あった aaa は消えてしまったことになる。

このことは cp aaa d7 のように、コピー先のファイル名を省略した場合でも同じで、d7 に aaa というファイルがもしあれば上書きされてしまう。一般的には cp コマンドでコピーする際には上書きは警告されないの注意が必要である（本学のシステムでは確認のメッセージが出るように設定されているが、他のシステムではそうとは限らない）。

〈移動先に移動元のファイルと同じ名前のファイルがある場合〉

移動に関してもまったく同様である。カレントディレクトリとディレクトリ d8 にはどちらも同じ名前の aaa というファイルがある。カレントディレクトリの aaa を d8 に移動したらどうなるか、確認してみよう。

```
$ mv aaa d8/aaa▼
```

コピーの場合と同様、「上書きしてよいか」のメッセージのメッセージが出てくるので、「y」を入力する。本当に移動したかどうか確認してみよう。

```
$ ls -l aaa▼
```

そのようなファイルはないというメッセージが出る。では、d8 の aaa を確認してみよう。

```
$ ls -l d8/aaa▼
```

aaa は移動したことが分かる。つまり、移動先に元々あった aaa は消えてしまったことになる。このことは mv aaa d8 のように、移動先のファイル名を省略した場合でも同じで、d8 に aaa というファイルがもしあれば上書きされてしまう。通常 mv コマンドで移動する際には上書きは警告されないの注意が必要である（本学のシステムでは確認

のメッセージが出るように設定されているが、他のシステムではそうとは限らない。

6. おわりに

この記事では、Unix 環境において、コマンドを使ってファイルやディレクトリを操作する方法を説明し、同時にディレクトリについての理解を深めることを目指した。

ディレクトリは、Windows などの GUI 環境ではフォルダと呼ばれるものに相当し、文字通り、ファイルを入れて整理・管理するための「入れ物」である。

確かに、フォルダとフォルダの間では、マウスを使ってファイルのコピーや移動が簡単にできる。しかしながら、GUI 環境を使用する場合でも、いくつかの理由でディレクトリについて理解しておくべきである。

一つには、GUI 環境では複雑なことを一気にすることはできず、単純な操作をひたすら繰り返すしか方法がないからである。

そうした場合、Windows 環境であれば、スタートメニューのアクセサリからコマンドプロンプトの画面（端末のウィンドウ）を起動し、そこからコマンド（Unix でのコマンドと似ているものが多い）を使って操作したり、実行するコマンド群をあらかじめ記述したバッチファイルを利用したりすることができる。ただ、そうした場合には、ディレクトリ

を正しく指定する必要がある。

もう一つは、Windows などのシステムの詳細な設定やネットワーク関係の設定においても、やはりディレクトリについての知識が不可欠となるからである。

要するに、GUI 環境だけでは、操作に時間がかかったり、できることが限られてしまったりするので、Windows であっても、コンピュータを本格的に使いこなそうと思うならば、ディレクトリについての理解を深めておくことが必要なのである。

Unix 環境では、本稿では紹介しなかったが、ワイルドカードや正規表現などを用いることで、複雑なファイルやディレクトリの操作（たとえば、条件に合ったファイルだけを特定のディレクトリにコピーすることなど）が可能となる。さらには、シェルスクリプトを作成して、そうした操作を自動化することもできる。本記事で基本的なことを学んだあとは、そうした応用も学習してほしい。

参考文献

- 中井 獏：ゼロからわかる UNIX 基礎講座，技術評論社（2009）
- 舟本 奨：実用 UNIX ハンドブック第3版，ナツメ社（2006）
- 中島康彦：UNIX で「やりたい」ことを「できる」にかえる基本の12章，メディア・テック出版（2000）
- 山下哲典：UNIX シェルスクリプトコマンドブック第2版，ソフトバンククリエイティブ（2010）