

情報科学演習／データベース入門 A 資料 2

ディレクトリの階層構造とパス名

平成 30 年 4 月 16 日

目次

1	復習 — ディレクトリ関係の用語と記号のまとめ	1
2	階層化したディレクトリの作成と利用	1
2.1	階層化したディレクトリ	1
2.1.1	練習	2
2.2	カレントディレクトリと親ディレクトリの操作	3
2.2.1	親ディレクトリと子ディレクトリ	3
2.2.2	カレントディレクトリと親ディレクトリの指定法 — “.” と “..”	3
2.2.3	練習	4
3	ディレクトリ間のファイルの移動と複写	5
3.1	ディレクトリ間のファイルの移動と複写 — mv, cp	5
3.1.1	練習	5
4	問題	6
5	パス名	6
5.1	ルートディレクトリとディレクトリの階層構造	6
5.2	パス名	7
5.2.1	絶対パス名	8
5.2.2	相対パス名	8
6	パス名を用いたファイルとディレクトリの操作	9
6.1	実習準備	9
6.2	単一のコマンド引数におけるパス名の利用	10
6.2.1	練習 — カレントディレクトリの変更	10
6.2.2	練習 — ファイル名の一覧表示やファイルの内容閲覧	10
6.3	複数のコマンド引数におけるパス名の利用	11
6.3.1	練習 — ファイルの複写や移動	11
6.4	まとめ — コマンドに対するファイルとディレクトリの指定方法	11
7	問題	12

1 復習 — ディレクトリ関係の用語と記号のまとめ

ディレクトリ: ファイルを分類して格納するための器。関係コマンド一覧は前回資料末尾にある。

カレントディレクトリ: 現在使っているディレクトリ。

- コマンドの引数に与えたファイル名やディレクトリ名は、カレントディレクトリのファイルやディレクトリを指すことになる。
- カレントディレクトリを別の *directory* に変更するには、`cd directory` を実行する。

ホームディレクトリ: ユーザーが自分のファイルやディレクトリを格納するためのディレクトリ。

- ログイン直後のカレントディレクトリはホームディレクトリである。
- 記号 `~` で表すことがある。
- 引数無しで `cd` を実行すれば、ホームディレクトリがカレントディレクトリとなる。

2 階層化したディレクトリの作成と利用

2.1 階層化したディレクトリ

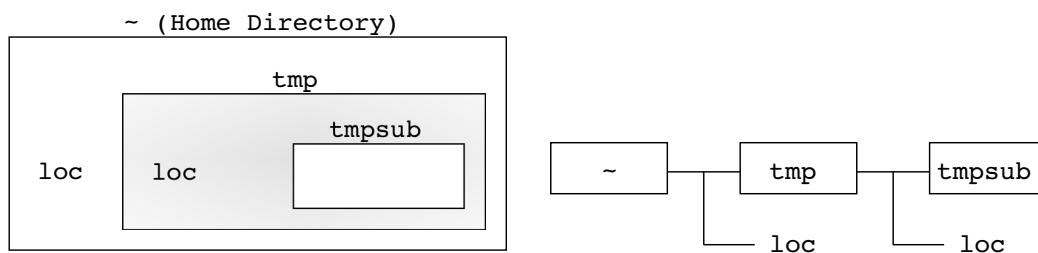


図 1: 階層化ディレクトリの例

- ディレクトリの中には、さらにディレクトリを作ることができます。この仕組みを利用すると、ファイルを大分類・中分類・小分類のように、階層的に分類して整理できます。図 1 の左の図では、ホームディレクトリの中にディレクトリ `tmp` があり、さらにその中にディレクトリ `tmpsub` がある様子を示しています。なお、`loc` は通常のファイルです。
- 右の図は、左の図を別の形式で描いたものであり、ディレクトリの中にあるファイルやディレクトリを右側に実線で結んで描画することにより、ディレクトリの階層的な構造を表しています。今後、ディレクトリの階層構造の表現には、右の図の形式のみを用います。
- 左の図における斜線は、カレントディレクトリが `tmp` であることを表現したものです。この状態でカレントディレクトリに存在するのは、斜線部に書かれたファイル `loc` とディレクトリ `tmpsub` です。これらが右の図ではどれであるか、わかりますか。

コマンド操作で指定するファイル名やディレクトリ名は、カレントディレクトリのファイルやディレクトリである点に留意してください。

- 一つのディレクトリに、同じ名前のファイルやディレクトリを複数作ることはできませんが、ディレクトリが異なれば可能です。従って、図 1 のように、同名のファイル `loc` がホームディレクトリおよびディレクトリ `tmp` に存在することは許されます¹。
- `ls` コマンドの引数に既存のディレクトリ `directory` を与えて

```
ls [options] directory
```

の形式で実行すれば、`directory` 内のファイル一覧が出力されます²。

もしも、カレントディレクトリにディレクトリ `tmpsub` があるならば、

```
ls tmpsub
```

は `tmpsub` に存在するファイルの一覧を出力します。

2.1.1 練習

図 1 に示したディレクトリとファイルをホームディレクトリ (`~`) に作ります。自分が今のディレクトリにいるのか (カレントディレクトリがどこなのか) を常に意識しながら、次の操作をしてください。

1. 通常のファイル `loc` をホームディレクトリに作ります。 `loc` の中身は `Here is my home` とします。

```
echo Here is my home > loc
```

正しくできたか、`ls` コマンドおよび `cat loc` で確認してください。

2. カレントディレクトリをホームディレクトリから `tmp` に変更します。ディレクトリ `tmp` が存在することを確認してから、

```
cd tmp
```

を実行してください。 `tmp` が存在しなければ `mkdir` コマンドで作成してから上記を実行してください。なお、今後、ディレクトリ変更の操作を、単に「`tmp` に移る・行く」のように表現することがあります。

3. `ls` を実行して、カレントディレクトリである `tmp` に存在するファイルを調べてください。 `tmp` には、まだ何も作っていませんから、何も表示されませんね。
4. `echo` コマンドを使って、ディレクトリ `tmp` の中に `loc` という通常のファイルを作ってください。ファイルの中身は `Here is tmp` とします。正しくできたか確認してください。
5. `mkdir` コマンドを使って、ディレクトリ `tmp` の中に `tmpsub` というディレクトリを作ってください。結果を確認してください。

¹ファイル名 `loc` は単語 `location` からとった名前です。後の練習では、このファイルを実際に作成し、カレントディレクトリがどこであるかを確認するために使います。

²`ls` にオプション `-d` を指定すると、ディレクトリの内容ではなく、ディレクトリ自身を表示します。

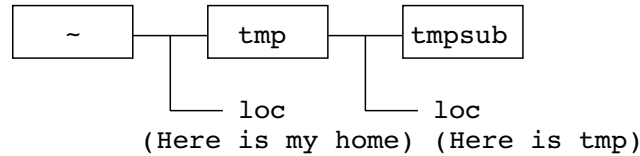


図 2: 親ディレクトリと子ディレクトリ

6. ホームディレクトリに戻るために、引数無しで `cd` を実行してください。
ちゃんとホームに帰れましたか？ファイル `loc` の中身を確認してください。
7. `rmdir tmp` を試してください。
ディレクトリ `tmp` を削除できませんね。 `rmdir` は中身が空のディレクトリを削除するコマンドだからです。 `tmp` は消さないでおきましょう。
8. ホームディレクトリをカレントディレクトリとしたままで、ディレクトリ `tmp` の中身 (`tmp` に存在するファイル) を確認してみましょう。 `ls` の引数にはディレクトリ名を与えることができるのでしたね。 `ls tmp` と `ls -F tmp` を実行してください。

2.2 カレントディレクトリと親ディレクトリの操作

2.2.1 親ディレクトリと子ディレクトリ

図 2 は、前節 2.1.1 で作成した、ホームディレクトリ (`~`) 内のディレクトリ/ファイルの構造を、再度示したものです。

あるディレクトリの一つ上位のディレクトリを **親ディレクトリ** (parent directory) と呼びます。図 2 では、 `tmpsub` の親ディレクトリは `tmp` であり、 `tmp` の親ディレクトリは `~` です。逆に、 `tmpsub` を `tmp` の子ディレクトリ (child directory) またはサブディレクトリ (subdirectory) と呼びます。 `tmp` は `~` の子ディレクトリです。

2.2.2 カレントディレクトリと親ディレクトリの指定法 — “.” と “..”

カレントディレクトリの子ディレクトリは「カレントディレクトリに存在するディレクトリ」です。そのため、子ディレクトリの名前をそのままコマンドの引数として指定すれば、それがコマンドの処理対象となります。

一方、カレントディレクトリ自身や「カレントディレクトリの親ディレクトリ」は、カレントディレクトリに存在するディレクトリではありません。したがって、ディレクトリ名をコマンドの引数に与えても、コマンドの処理対象にはなりません。これらをコマンドの引数にしたければ、次の記号を使います。

- .(ドット) : カレントディレクトリ自身
- ..(ドット二つ) : カレントディレクトリの親ディレクトリ

2.2.3 練習

1. カレントディレクトリを、図 2 の tmpsub に変更してください。

tmpsub はカレントディレクトリに存在するディレクトリではないので、`cd tmpsub` だけではうまくいきません。まず `tmp` に移ってから、tmpsub に移りましょう。

2. tmpsub (カレントディレクトリ) に存在するファイルを `ls` コマンドで調べましょう。
何も表示されませんね。

3. `ls` に `-F` と `-a` の二つのオプションを付けて実行してみましょう。`ls` に複数のオプションを付けるときには、それらを `-aF` のようにまとめて書くことができます。

```
ls -aF
```

`-a` は `.` で始まるものを含めて全てのファイル名を表示する `ls` のオプションでしたね。カレントディレクトリ自身とその親ディレクトリの存在を確認できましたか。

4. カレントディレクトリを、親ディレクトリである `tmp` に変更します。

まず、`cd tmp` を試しましょう。

コマンドの引数には、カレントディレクトリに存在するファイル/ディレクトリ名を指定するのでしたから、これではうまくいきません。

そこで、先ほどカレントディレクトリに存在することを確認した、親ディレクトリの記号 `..` を使います。

```
cd ..
```

エラーメッセージは出ませんね。成功です。

5. カレントディレクトリの内容を `ls -aF` で表示してみてください。loc というファイルが存在するはずですから、ファイルの中を見てください。

今はちゃんと `tmp` にいますね。

6. `ls -aF .` の結果を予想して、試してください。

`ls` の引数に `.` を指定すると、カレントディレクトリに存在するファイルの一覧が得られます。したがって、結果は `ls -aF` と同じです。

7. `ls -aF` や `ls -aF .` の表示では `..` も現われましたから、親ディレクトリ (ホームディレクトリ) の指定には、再度 `..` を使うことができます。

`ls` コマンドの引数に `..` を与えて、親ディレクトリに存在するファイルの一覧を表示してみましょう。

8. `..` を使ってホームディレクトリに戻りましょう。

3 ディレクトリ間のファイルの移動と複写

3.1 ディレクトリ間のファイルの移動と複写 — mv, cp

前の資料では、mv と cp の引数にファイル名を与えてファイル名の変更とファイルの複写を行いましたが、mv や cp の最後の引数に既存のディレクトリ名 *directory* を指定すると、mv と cp は次のとおり動作します。

mv *file directory* — *file* を *directory* に移動する

cp *file directory* — *file* を *directory* に、同じ名前のファイルとして複写する

directory として . や .. を指定することも可能です。なお、mv の場合、*file* としてディレクトリ名を与えれば、ディレクトリを別のディレクトリに移動することもできます³。

3.1.1 練習

1. 準備

- (a) カレントディレクトリをホームディレクトリにするために、cd コマンドを引数無しで実行してください。
- (b) ホームディレクトリに通常のファイル `now` がありますか？中身はどうなっていますか？ファイル `now` が無ければ `date > now` で作成してください。
- (c) ディレクトリ `tmp` がホームディレクトリに存在することを `ls -F` で確認してください。

2. ファイルの移動

- (a) ファイル `now` をディレクトリ `tmp` に移動します。 `mv now tmp` を実行してください。
- (b) `ls` を実行して、ホームディレクトリから `now` が無くなったことを確認しましょう。
- (c) `ls tmp` を実行して、ディレクトリ `tmp` に `now` が存在することを確認しましょう。
- (d) カレントディレクトリを `tmp` に変更してから、ファイル `now` の内容を見てみましょう。`now` はディレクトリ `tmp` に移動できていますね。

3. ファイルの複写

- (a) ディレクトリ `tmp` のファイル `now` を親ディレクトリ（ホームディレクトリ）に複写します。 `cp now ..` を実行してください。
- (b) `ls` を実行して、カレントディレクトリ (`tmp`) に `now` が残っていることを確認しましょう。
- (c) カレントディレクトリ (`tmp`) のファイル `now` の内容を見てみましょう。
- (d) カレントディレクトリを親ディレクトリ（ホームディレクトリ）に変更して、複写した `now` の内容を見てみましょう。

³cp でディレクトリを複写するにはオプション `-r` が必要です。

4 問題

1. ホームディレクトリに存在する通常のファイルやディレクトリの名前一覧を表示しましょう。
2. その中でディレクトリはどれでしょう？ ファイルの種類を表す記号 (ディレクトリの場合は /) がつくように、ファイルやディレクトリ名の一覧を表示しましょう。
3. ホームディレクトリに存在するファイル `loc` の名前を `loc.home` に変更してください。
4. ホームディレクトリの子ディレクトリである `tmp` に存在するファイル `loc` の名前を `loc.tmp` に変更してください。
5. ディレクトリ `tmp` に存在するファイル `now` を、ディレクトリ `tmpsub` に移動してください。
6. ディレクトリ `tmp` をカレントディレクトリにした状態で、ディレクトリ `tmpsub` にファイル `now` があることを確認してください。
7. ディレクトリ `tmpsub` をカレントディレクトリにした状態で、ディレクトリ `tmp` にファイル `now` が無いことを確認してください。
8. ホームディレクトリに存在するディレクトリ `tmp` の名前を `junk` に変更してください。
9. ディレクトリ `junk` と、その中のすべてのファイルやディレクトリを削除してください。ディレクトリの削除には `rmdir` コマンドを使ってください。

5 パス名

5.1 ルートディレクトリとディレクトリの階層構造

以下を順に実行してみましょう。

1. ホームディレクトリがカレントディレクトリになっていることを確実にするために、引数無しで `cd` コマンドを実行してください。
2. カレントディレクトリ名を表示する `pwd` というコマンドを実行してみましょう。
/ で区切られた文字列の一番右側に、自分のユーザー名 (ログイン名) が表示されていれば、カレントディレクトリはホームディレクトリです (ホームディレクトリにいます)。`pwd` の出力の正確な意味は第 5.2.1 節で説明します。
3. `ls -aF` を実行して、ホームディレクトリに存在する全てのファイルとディレクトリを、ファイルの種類を表す印つきで表示しましょう。表示が画面に納まらない場合、おまじないとして

```
ls -aF | head
```

を実行してください。.. (親ディレクトリ) がありますね。みなさんのホームディレクトリには、さらに上位のディレクトリが存在するのです。

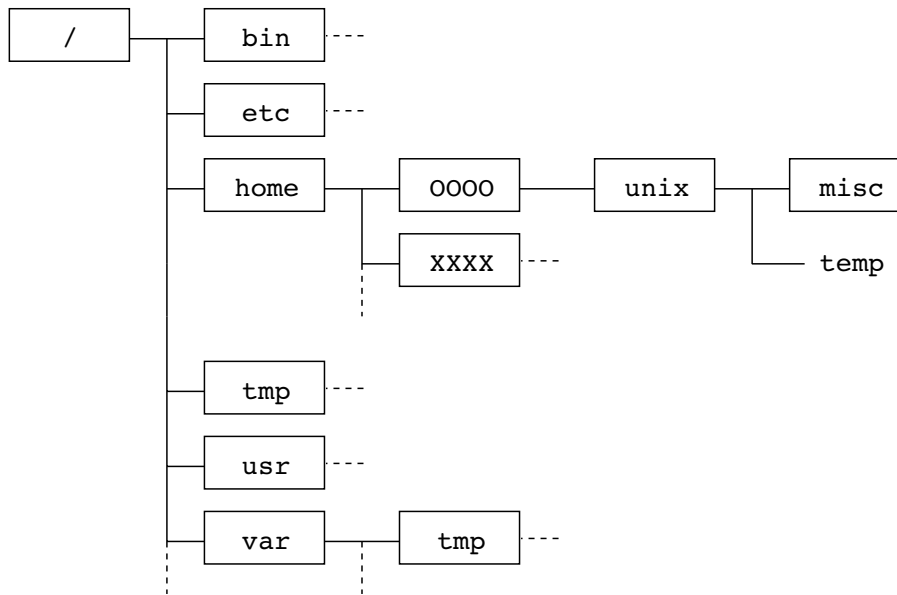


図 3: UNIX システムにおけるディレクトリ／ファイルの階層構造例

4. `cd ..` を実行してカレントディレクトリを上位のディレクトリに変更し、`pwd` や `ls -aF` の結果を観察しましょう。この操作を繰り返してください。

`pwd` の出力が `/` になったら、`cd ..` を実行しても、`ls` や `pwd` の出力は変わりません。行き止まりです。

UNIX では、ディレクトリやファイルから成る階層構造（木構造）が、システムにただ一つ存在します。その階層構造に唯一存在する最上位（根; root）のディレクトリをルートディレクトリ（root directory）といい⁴、記号 `/`（スラッシュ）で表します⁵。

図 3 は UNIX システムにおけるディレクトリやファイルの階層構造の一例です。ユーザーのホームディレクトリも、すべてこの階層構造の中にあります。この図では、ユーザー `0000` と `XXXX` のホームディレクトリを、各々、`0000` と `XXXX` で描いています。

5.2 パス名

パス名 (pathname) とは、ファイルやディレクトリの名前を、ディレクトリの階層構造における場所の記述を含めて表現するものです⁶。パス名には絶対パス名 (absolute pathname) と相対パス名 (relative pathname) の二種類があります。

⁴Windows ではドライブ毎にディレクトリ（フォルダ）の階層構造を作ることができ、ルートディレクトリやカレントディレクトリもドライブ毎に存在します。一方、UNIX にはドライブの概念はありませんので、ルートディレクトリはただ一つです。

⁵Windows のコマンドプロンプトでは、ルートディレクトリを `¥` で表します

⁶ディレクトリ階層内の場所の情報を含む名前に対しても、ファイル名やディレクトリ名という用語を使うことがありますが、この授業ではパス名として区別して扱います。

5.2.1 絶対パス名

絶対パス名では、ディレクトリやファイルの位置を記述する際の基点を、ルートディレクトリ (/) に固定します。これにより、絶対パス名は必ず / で始まります。絶対パス名を使ったファイルやディレクトリの表記法は次のとおりです。

ルートディレクトリ: /

ルートディレクトリ以外: ルートディレクトリを意味する / に続き、ルートディレクトリから当該ディレクトリやファイルに至るまでの道筋（順路）を、 / で区切りながら順に記す。

例えば、図 3 におけるディレクトリ OOOO の絶対パス名は /home/OOOO であり、ファイル temp の絶対パス名は /home/OOOO/unix/temp です⁷。

システム内のどんなファイルやディレクトリであっても、それを表す絶対パス名は一つです。第 5.1 節で使った pwd コマンドは、カレントディレクトリの名前を絶対パス名で出力するコマンドです。

5.2.2 相対パス名

相対パス名は、階層構造内のあるディレクトリを基点として、ファイルやディレクトリの場所を表現するものです。ただし、コマンド操作における相対パス名の利用では、その基点はカレントディレクトリになりますので、この資料では相対パス名を「カレントディレクトリを基点としてファイルやディレクトリの場所を表す名前」の意味で使います。絶対パス名との大きな違いは、パス名の先頭に / を付けないことです。

相対パス名の書き方は次のとおりです。ここでは、相対パス名で表現したいディレクトリやファイルが、カレントディレクトリの子孫（ディレクトリの階層構造図では、右側が子孫）、祖先（左側）、およびそれ以外の場合に分けて説明します。

カレントディレクトリの子孫の場合: カレントディレクトリからファイルやディレクトリまでの道筋（順路）を / で区切りながら順に記す。ただし、カレントディレクトリ名は記さない。カレントディレクトリを意味する . (ドット) を明示して、 ./ に続いて順路を記述してもよい。

カレントディレクトリの祖先の場合: カレントディレクトリからディレクトリまでの順路を、親ディレクトリを意味する .. (ドット二つ) を / で区切りながら書き並べる。

上記以外 (カレントディレクトリの子孫の子孫) の場合: 祖先の相対パス名に、その子孫であるファイルやディレクトリまでの相対パス名を追加する。

カレントディレクトリが図 3 の OOOO であるとしします。このとき、図 3 におけるディレクトリやファイルのいくつかを、絶対パス名と相対パス名で表したのが表 1 です。カレントディレクトリを変更すれば相対パス名も変わることに注意しましょう。

⁷Windows におけるパス名では、 / の代わりに ¥ を用います。例えば、H ドライブのルートディレクトリの下の子孫の絶対パス名は H:\¥onsei です。

表 1: 絶対パス名と相対パス名の対応

絶対パス名	相対パス名	備考
/home/OOOO	.	カレントディレクトリ
/home/OOOO/unix	unix または ./unix	
/home/OOOO/unix/temp	unix/temp または ./unix/temp	
/home/OOOO/unix/misc	unix/misc または ./unix/misc	
/home	..	カレントディレクトリの親
/	../..	カレントディレクトリの親の親
/home/XXXX	../XXXX	カレントディレクトリの親の子
/var	.././var	
/var/tmp	.././var/tmp	

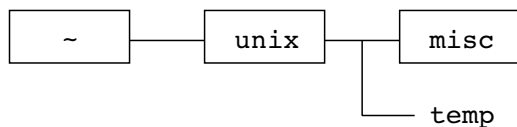
6 パス名を用いたファイルとディレクトリの操作

ファイル名やディレクトリ名を引数とするコマンドには、ファイル名やディレクトリ名に代え、パス名を指定することができます。パス名には、絶対パス名と相対パス名のどちらでも使えます。

ファイル名やディレクトリ名だけでは、カレントディレクトリに存在するファイルやディレクトリしか操作できませんでしたが、パス名を利用すれば、カレントディレクトリがどこであるかわからず、任意のファイルやディレクトリの操作が可能です。

6.1 実習準備

ホームディレクトリにディレクトリ `unix` を作成し、そこにディレクトリ `misc` を作成してください。また、ホームディレクトリに存在するファイル `now` を `unix` に移動し、その名前を `temp` に変更してください。



作成したら、ホームディレクトリで `ls -R unix` を実行し、次の出力が得られることを確認してください。

```

unix:
misc temp

unix/misc:

```

6.2 単一のコマンド引数におけるパス名の利用

6.2.1 練習 — カレントディレクトリの変更

以下の操作を順に行ってください。cd コマンドを実行する度に、pwd コマンドでカレントディレクトリがどこであるかを確認してください。

1. まず、pwd コマンドでカレントディレクトリの絶対パス名を確認しましょう。
2. 絶対パス名を使ってルートディレクトリに移る

```
cd /
```
3. 絶対パス名を使ってディレクトリ /var/tmp に移る

```
cd /var/tmp
```
4. ホームディレクトリに移る (ここではパス名を使わずに、引数無しの cd で済ませます)

```
cd
```
5. 相対パス名を使って、ディレクトリ unix の子ディレクトリである misc に移る

```
cd unix/misc
```
6. 相対パス名を使ってホームディレクトリに移る

```
cd ../../..
```

6.2.2 練習 — ファイル名の一覧表示やファイルの内容閲覧

以下では絶対パス名のみを使いますので、カレントディレクトリがどこであっても、実行結果は同じです。

1. ルートディレクトリに存在するファイルの一覧

```
ls /
```
2. ルートディレクトリの子ディレクトリである etc に存在するファイルの一覧

```
ls /etc
```
3. ルートディレクトリの子ディレクトリである etc に存在するファイル redhat-release の内容閲覧⁸

```
cat /etc/redhat-release
```

⁸/etc/redhat-release は特定の OS (Redhat 社の Linux やその派生 OS) にもみ存在するファイルです。

6.3 複数のコマンド引数におけるパス名の利用

6.3.1 練習 — ファイルの複写や移動

以下は、すべてホームディレクトリをカレントディレクトリとした状態で実行してください。cp や mv は、最後の引数が既存のディレクトリかどうかによって動作が異なりますので、注意してください(第 3 章参照)。これらのコマンドを実行したら、ls コマンドや cat コマンドを適切に実行し、結果を確認してください。

1. まず、カレントディレクトリ (ホームディレクトリ)、およびカレントディレクトリの子ディレクトリである `unix` に、どんなファイルやディレクトリがあるのか、確認しておきましょう。

```
ls -F
```

```
ls -F unix
```

2. ディレクトリ `unix` に存在するファイル `temp` の名前を `Temp` に変更
(`mv file1 file2` の形式で mv を実行)

```
mv unix/temp unix/Temp
```

3. ルートディレクトリの子ディレクトリである `etc` に存在するファイル `redhat-release` を、カレントディレクトリのファイル `dummy` に複写 (ディレクトリ `dummy` は存在しないとする)
(`cp file1 file2` の形式で cp を実行)

```
cp /etc/redhat-release dummy
```

4. ディレクトリ `unix` に存在するファイル `Temp` を、ディレクトリ `misc` に複写
(`cp file directory` の形式で cp を実行)

```
cp unix/Temp unix/misc
```

5. ディレクトリ `unix` に存在するファイル `Temp` をディレクトリ `misc` に、ファイル名 `tempfile` として複写 (`misc` に `tempfile` というディレクトリは存在しないとする)
(`cp file1 file2` の形式で cp を実行)

```
cp unix/Temp unix/misc/tempfile
```

6. `tempfile` をカレントディレクトリに移動
(`mv file directory` の形式で mv を実行)

```
mv unix/misc/tempfile .(ドット)
```

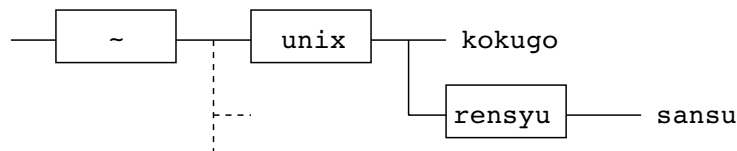
6.4 まとめ — コマンドに対するファイルとディレクトリの指定方法

- カレントディレクトリ内のファイルやディレクトリを扱う場合:
ファイル名やディレクトリ名を、そのままコマンドの引数とすればよい。
- カレントディレクトリ以外のファイルやディレクトリを扱う場合:
ファイルやディレクトリを、パス名でコマンドの引数に与える。
- パス名に使われる記号

- . (ドット): カレントディレクトリ
- .. (ドット二つ): 親ディレクトリ
- / (スラッシュ): ルートディレクトリ, またはディレクトリやファイルの親子関係を意味する区切り文字

7 問題

1. ホームディレクトリに存在するファイルやディレクトリの名前を確認してください。ホームディレクトリの tempfile を削除してください。
2. カレントディレクトリをホームディレクトリとしたままで, ディレクトリ unix の子ディレクトリである misc に存在するファイルの名前を確認し, misc に存在する Temp を削除しましょう。
3. カレントディレクトリをホームディレクトリとしたままで, ルートディレクトリの子ディレクトリである bin に存在するファイル true を, ホームディレクトリの子ディレクトリ unix に複写しましょう。ただし, 複写先でファイル名が mytrue となるようにしてください。この操作は cp コマンドを一回だけ実行して行ってください。
4. カレントディレクトリを unix に変更してから, ホームディレクトリのファイル/ディレクトリの一覧を出力し, さらにホームディレクトリのファイル dummy をディレクトリ misc に移動してください。
5. 次の指示に従って, ファイルやディレクトリの操作を行ってください。カレントディレクトリは自由に変更して構いません。できるだけ少ない手間で行ってください。
 - (a) ホームディレクトリに通常のファイル kokugo と sansu を作成してください。ファイルの作成には echo コマンドと > を使います。ファイル kokugo の内容は aiueo とし, sansu の内容は 1+1=2 としておきましょう。
 - (b) ディレクトリ unix の内容が次の図のとおりになるように, ディレクトリ rensyu を用意して, ホームディレクトリの kokugo と sansu を移動してください。図に示した以外のファイルやディレクトリを unix から削除してください。



- (c) さらに次のようにしてください。

