

情報科学演習 資料 3

パス名とコマンドの実行

令和 7 年 4 月 28 日

目 次

1	パス名	1
1.1	ルートディレクトリとディレクトリの階層構造	1
1.2	パス名	1
1.2.1	絶対パス名	2
1.2.2	相対パス名	2
2	パス名を用いたファイルとディレクトリの操作	3
2.1	単一のコマンド引数におけるパス名の利用	4
2.1.1	練習 — カレントディレクトリの変更	4
2.1.2	練習 — ファイル名の一覧表示やファイルの内容閲覧	4
2.2	複数のコマンド引数におけるパス名の利用	5
2.2.1	練習 — ファイルの複写や移動	5
2.3	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	5
3	コマンドの実体とパス名を用いたコマンド実行	6
3.1	練習	6
4	問題	8

1 パス名

1.1 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 1.2.1 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。

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¹、記号 `/` (スラッシュ) で表します²。

図 1 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 `OOOO` と `XXXX` のホームディレクトリを、各々、`OOOO` と `XXXX` で描いています。

1.2 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³。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。

¹Windows 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。

²Windows では、ルートディレクトリを `¥` で表します

³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。

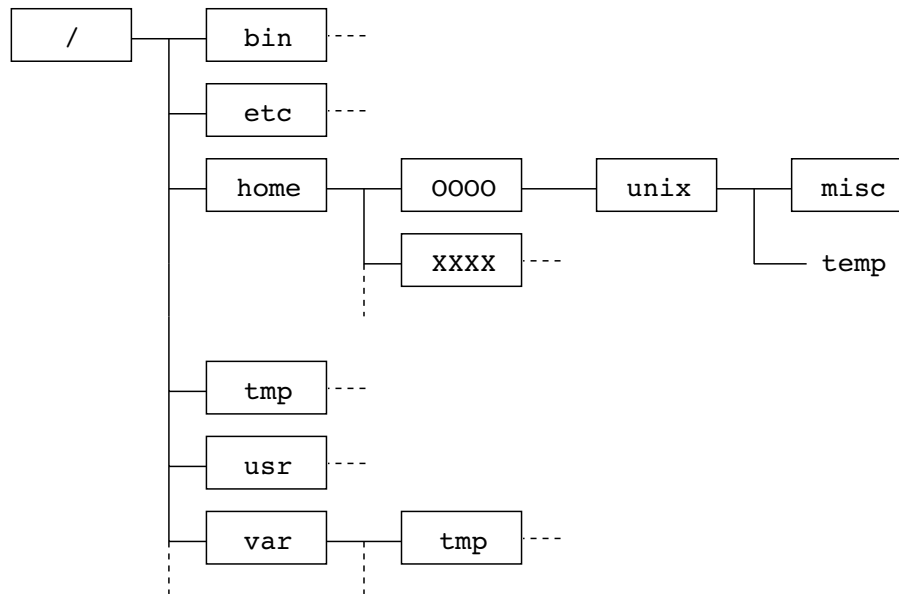


図 1: UNIX システムにおけるディレクトリ／ファイルの階層構造例

1.2.1 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 1 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⁴。

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唯一つです。第 1.1 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。

1.2.2 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。

⁴Windows 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 onsei 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	

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）の場合、祖先（左側）の場合、および、それら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祖先の子孫) 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 1 の OOOO であるとしします。このとき、図 1 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。

2 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。

以降の練習を行うには、図 1 の OOOO の右に描かれているように、ディレクトリ `unix` が自分のホームディレクトリに存在し、さらに `misc`, `temp` が `unix` に存在している必要があります。

2.1 単一のコマンド引数におけるパス名の利用

2.1.1 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 ../../
```

2.1.2 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⁵

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

⁵/etc/redhat-release は特定の OS (Redhat 社の Linux やその派生 OS) にのみ存在するファイルです。

2.2 複数のコマンド引数におけるパス名の利用

2.2.1 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(資料2「ディレクトリの階層構造」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)
```

2.3 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

3 コマンドの実体とパス名を用いたコマンド実行

1. ディレクトリ `/bin` にどんなファイルがあるか調べましょう。`ls /bin` を実行してください⁶。
`/bin` には `cat`, `cp`, `ls`, `pwd` などの名前を持つファイルがあります。
2. コマンド行に `/bin/pwd` とタイプして `<ENTER>` を押してみてください。
`pwd` の代わりに `/bin/pwd` でも `pwd` コマンドを実行できましたね。

このことから予想されるとおり、`pwd` コマンドの実体は、ディレクトリ `/bin` に存在する通常のファイル `pwd` です。UNIX コマンドの多くは、コマンドと同名のファイルとして存在していて、`/bin` に存在するファイルはすべて実行可能なコマンドです^{7,8}。

コマンドの引数にパス名が使えるのと同様に、コマンド自身を指定する際にもパス名が使えます。先ほど `/bin/pwd` で `pwd` コマンドを実行できたのは、そのためです。

より正確にいうと、ファイルとして存在するコマンドを実行するには、本来、コマンドのパス名を使う必要があります。しかし、それでは不便ですので、`/bin` のような特定のディレクトリに存在するコマンドを、コマンド名のみで実行できるようにする仕組みが設けられています⁹。そのおかげで、`/bin/pwd` や `/bin/ls` 等を `pwd` や `ls` とだけタイプして実行できるのです。

3.1 練習

1. カレントディレクトリを、ホームディレクトリに存在する `eis25` に変更した後で、`hostname` コマンドを実行してください。

`hostname` はコンピュータのホスト名（または FQDN (Full Qualified Domain Name)）を出力するコマンドです。
2. `ls /bin/h*` を実行すると、ディレクトリ `/bin` に存在する `h` で始まるファイルの名前が全て出力されますので¹⁰、`/bin` に `hostname` コマンドの実体であるファイル `hostname` が存在することを確認してください。

⁶`/bin` が `/usr/bin` の別名（シンボリックリンクというもの）になっているシステムもあります。

⁷`/bin` に存在するファイルの大部分は実行可能なバイナリ (binary) ファイルです。これらはテキストファイルではないので、`cat` 等で中を読むことはできません。

⁸ファイルとしては存在しないコマンドとして、シェルが持っているコマンド（シェルの組み込みコマンド）があります。例えば `exit` はシェル組み込みコマンドです。シェル組み込みコマンドの使い方はシェルのマニュアルに記載されています。

⁹どのディレクトリに存在するコマンドが、コマンド名のみで実行できるのかは、設定に依存します。具体的には、`echo $PATH` で出力される、シェル環境変数 `PATH` に設定されたディレクトリに存在するコマンドはファイル名（コマンド名）のみで実行できます。なお、`/bin` が `/usr/bin` のシンボリックリンクになっているシステムでは、コマンドの実体は `/usr/bin` にあります。そのため、環境変数 `PATH` に `/bin` が入っていない場合があります。

¹⁰*は任意の文字列にマッチするシェルのメタキャラクタです。使い方は別の資料で学びます。

3. 絶対パス名を使って `hostname` コマンドを実行しましょう。

`/bin/hostname` とタイプして `<ENTER>` ですね。

4. `/bin/hostname` をカレントディレクトリにコピーしてください。ただし、コピー先のファイル名を `myhostname` とします。

`cp /bin/hostname myhostname` でいいですね。実行したら、正しくコピーされたかを `ls` コマンドで確認してください。

5. コピーした `myhostname` は `hostname` と同じ内容のファイルですから、コマンドとして実行できます¹¹。

- (a) まず `myhostname` とタイプして `<ENTER>` を押して、コマンド名による実行を試みてください。

特定のディレクトリに存在するコマンドを除き、コマンドはパス名で実行する必要がありますのでしたね。`eis25` は、その特定のディレクトリではありませんので、ここに存在するコマンドをコマンド名 `myhostname` だけで実行することはできません¹²。

- (b) 相対パス名を使って `myhostname` コマンドを実行しましょう。

カレントディレクトリ (`.`) に存在するファイルであることを明示して `./myhostname` とタイプして `<ENTER>` を押してください。

`hostname` コマンドと同じ出力が得られたのは、`myhostname` が `hostname` と同じ内容のファイル（プログラム）だからです。仮にカレントディレクトリに `a.out` という実行可能なファイル（プログラム）があるとします。これを実行するには `./a.out <ENTER>` とします。

UNIX では、カレントディレクトリに存在する実行可能なファイル（コマンド） *command* を相対パス名で実行するには、カレントディレクトリ (`.`) を明示して `./command <ENTER>` とする必要があります¹³。

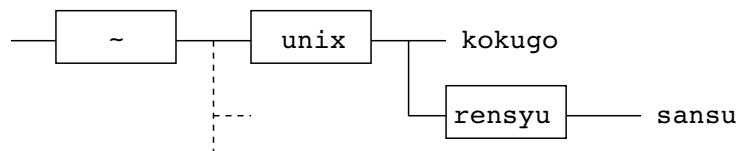
¹¹ コマンドとして実行できるかどうかは、ファイルの内容だけでなく、ファイルのアクセス権（実行可能なファイルかどうか）にも依ります。アクセス権については別の資料で学びます。

¹² Windows のコマンドプロンプトでは、カレントディレクトリに存在するコマンド（プログラム）はコマンド名（プログラムのファイル名）のみで実行できます。

¹³ Windows のコマンドプロンプト等では、カレントディレクトリに存在する実行可能なファイルは、ファイル名だけで実行できます。

4 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、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。

