

3.11 rmdir

rmdir コマンドはディレクトリの削除する。

削除するディレクトリを空にしておく必要がある。削除するディレクトリ (のパス) 名を *directory* とすると

```
$ rmdir directory
```

ReMove DIRectory

空でないディレクトリを削除する時には `rm -rf directory` とする。

4 ファイルシステム

4.1 階層構造

ディレクトリの親子関係を線で結ぶと木構造⁵ (ルートディレクトリを「根」、ディレクトリの親子関係を「幹」や「枝」、ディレクトリを「枝の分岐点」、ファイルを「葉」と考えると「木」と見なせる) と呼ばれる階層構造を作っている。

表 1: ターミナルエミュレータとファインダとの対応関係

ターミナル	Finder	説明
Application	アプリケーション	すべてのユーザーが利用できるアプリケーションがインストールされている。ターミナルなどのアプリケーションはユーティリティサブフォルダ、東女独自のアプリケーションは TWCUC サブフォルダにインストールされている。
Documents	書類	Word ファイルや PDF ファイルなどを格納する。
Desktop	デスクトップ	ここに格納すると aqua のデスクトップにアイコンが表示される。
Movie	ムービー	動画ファイルを格納する。
Pictures	ピクチャ	画像ファイルを格納する。

4.2 パス

階層的ファイルシステムでは、異なるディレクトリに同じ名前のファイルやディレクトリが存在することができるので、ファイルやディレクトリを一意に特定するため、パス名を用いる。

ルートディレクトリからのファイル名 (システム全体で一意的になる) を絶対パス名といい、カレントディレクトリからの相対位置は相対パス名という。絶対パス名の先頭は/(スラッシュ)である。先頭が/(スラッシュ)でも^(チルダ)でもないパス名は相対パス名である。

絶対パス名と相対パス名を総称しパス名という。

⁵頂点、頂点と頂点を結ぶ辺からなる集合をグラフという。木とは、グラフに属す任意の 2 つの頂点を結ぶ経路がただ 1 つのグラフをいう。

ユーザ asakawa のホームディレクトリの絶対パス名は/Users/asakawa であるが、これを ~asakawa とも表す。

4.3 ファイル名

4.3.1 ファイルには名前が付けられている

UNIX では、大文字と小文字は区別される。なお、“.”(ピリオド)、“_”(アンダースコア)、“-”(ハイフン)以外の記号はシェルが特別な意味に解釈することがあるので用いない方がよい。ファイル名にスペースや日本語があるとトラブルの原因になる。

4.3.2 ファイル名の.(ピリオド)の後ろの文字列は拡張子という

ファイル名の.(ピリオド)の後ろの文字列は拡張子といい、ファイルの性質を表す。適切な拡張子をつけると以下の効果がある。

- ファイル名からファイル形式を知ることができる。
- ネットワークを介してファイルの受け渡しを行うとき、対応するアプリケーションを事前に知らせることができる。
- コンパイラをはじめさまざまなプログラムが処理するファイルが適切な型のファイルか否かを判断する。

表 2: テキストファイルの主な拡張子

拡張子	ファイル
txt	プレインテキスト
html	HTML ファイル
tex	L ^A T _E X 2 _ε のソースファイル
c	プログラミング言語 C のソースファイル
java	java のソースファイル
cpp	C++ のソースファイル
ps	ポストスクリプトファイル
pdf	PDF ファイル

4.3.3 テキストファイル

ファイルには、(文字列として保存されエディタなどで)人間が読めるテキストファイル(text file)と(コンピュータが理解できる)バイナリファイル(binary file)がある。バイナリファイルは、機械語プログラム、デジタル画像情報、デジタル音声情報などである。

このうち、(表示形式の指定が入らない)地の文だけからなるファイルは、プレインテキスト(plain text)という。

表 3: バイナリファイルの主な拡張子

拡張子	ファイル
docx	Microsoft Word ファイル
xlsx	Microsoft Excel ファイル
pptx	Microsoft Powerpoint ファイル
jpeg, jpg	jpeg ファイル (画像ファイル)
gif	gif ファイル (画像ファイル)
png	png ファイル (画像ファイル)
mpeg, mpg	mpeg ファイル (動画ファイル)
exe	(Windows の) 実行可能ファイル

演習 4.1 1. ファイルの操作は Finder でも行える (Graphical User Interface:GUI)。Finder で行ったファイルの作成、消去、フォルダの作成、消去がターミナルエミュレータからコマンドを使って操作 (Character User Interface:CUI) ができることを確認せよ。

2. カレントディレクトリ名を表示するコマンドは何か？
3. カレントディレクトリを変更するコマンドは何か？
4. ファイルの一覧を表示するコマンドは何か？

4.3.4 ドットファイル

```
$ ls -alt
```

All Long Time

とするとカレントディレクトリにあるすべてのファイルが、許可属性、ファイルサイズや最終更新日時を時間順にソートされて表示される。

```
$ ls -alF
```

とするとカレントディレクトリにあるファイルの情報が表示される。

ここで `.DS_Store` は Mac OS X 独自のファイルで、Finder で表示した際のアイコンの大きさや位置の情報が記録されている。削除しても新しい `.DS_Store` が作られるので、支障はない。

`.`(ドット) で始まるファイル (`.DS_Store` など) はリソースフォークや各種設定ファイルで Finder では表示されない。

演習 4.2 ターミナルから以下の操作を行い、実行記録をメールにコピーアンドペーストして送信せよ。

1. ホームディレクトリの直下に `comp2d` というディレクトリを作れ。
2. カレントディレクトリを `comp2d` に変更せよ。
3. ディレクトリ `comp2d` の下に `exerdir` というディレクトリをつくれ。
4. カレントディレクトリを `exerdir` に変更せよ。

5. カレントディレクトリが `exerdir` であることを確認せよ。
6. ディレクトリ `exerdir` にあるファイルは”.”と”..”のみであることを確認せよ。
7. ”This is an exercise.”を内容とするテキストファイル `exer.txt` を `exerdir` に作れ。
8. `exer.txt` を `exer1.txt` という名前でコピーせよ。
9. `exer1.txt` が `exer.txt` のコピーであることを確認せよ。
10. `exerdir` にあるすべてのファイルを表示せよ。
11. `exer1.txt` をデスクトップ `~/Desktop` に移動せよ。
12. `exer1.txt` を削除せよ。
13. カレントディレクトリをホームディレクトリに移動せよ。

4.4 パターンによるファイル名の指定

4.4.1 メタキャラクタ

コマンドラインの文字には `<` や `|` のようにシェルが解釈する文字と、コマンド名や引数のようにコマンドに渡される文字がある。シェルが解釈する文字をメタキャラクタ (meta character)⁶ という。シェルのメタキャラクタはシェルが使ってしまいコマンドには渡されない。

4.4.2 ファイル名置換

ある条件に合致するファイル名およびディレクトリ名をパターンを用いてまとめて指定することができる。ファイル名のパターンを与え、シェルが条件に合致するすべてのファイル名と置き換える機能をファイル名置換という。シェルはパターンとマッチしたファイル名に置き換えてコマンドを実行する。パターンはメタキャラクタを使って与える⁷。

表記	意味
*	任意の文字列
?	任意の 1 文字
[string]	string に含まれるどれか 1 文字。[aA] とすると a か A がマッチする。
[文字 1-文字 2]	文字 1 から文字 2 の間の文字 (順序は ASCII コードによる)。 [0-9] とすると 0 から 9 までの数字 1 文字。[a-z] は a から z までの任意の 1 文字。任意の英小文字

ABC.txt ab.txt ab1.txt ab2.txt abc.txt のうちパターン `*.txt` `ab*.txt` `ab?.txt` `ab[0-9].txt` `[Aa]` `[Bb]` `[Cc].txt` にマッチするものを示す。

⁶メタキャラクタと良く似たものに正規表現がある。メタキャラクタはファイル名やディレクトリ名を指定するのに用いるが、正規表現はシェルで広く用いられる。メタキャラクタでは任意の 0 個以上の文字を表すのに `*` (アスタリスク) が使われるが、正規表現では `(ドット)` が使われる。正規表現では `*` を 0 個以上の繰り返しに用いる。

⁷メタキャラクタの `*` と `?` は Windows でも使える。Windows ではワイルドカード (wild card) とよばれている。

パターン	マッチするファイル
*.txt	ABC.txt ab.txt ab1.txt ab2.txt abc.txt (すべてのファイルとマッチした。)
ab*.txt	ab.txt ab1.txt ab2.txt abc.txt
ab?.txt	ab1.txt ab2.txt abc.txt
ab[0-9].txt	ab1.txt ab2.txt
[Aa][Bb][Cc].txt	ABC.txt abc.txt

ユーザが、

```
$ ls ab*.txt
```

とすると、シェルはパターンを展開しカーネルに

```
$ ls ab.txt ab1.txt ab2.txt abc.txt
```

のコマンドを処理するように伝える。ファイル名は ASCII コードによる辞書式順序で並べられる。

演習 4.3 上記を~/comp2d の下にファイルを作って確かめよ。

```
$ touch abc.txt
$ touch ab.txt
$ touch ab1.txt
$ touch ab2.txt
$ touch ABC.txt
$ ls *txt
ABC.txt      ab.txt      ab1.txt     ab2.txt     abc.txt
$ ls ab*.txt
ab.txt      ab1.txt     ab2.txt     abc.txt
$ ls ab?.txt
ab1.txt     ab2.txt     abc.txt
$ ls ab[0-9].txt
ab1.txt     ab2.txt
$ ls [Aa][Bb][Cc].txt
ABC.txt     abc.txt
```

4.5 ファイルのパーミッション

UNIX は複数のユーザが同時に使う OS であるので、ファイルには「所有者」がいて、他人のファイルを勝手に消したり出来ないようになっている。また、自分(所有者)は読めるけれど他人は読めないファイルや、自分では書き込めるけれど他人には書き込めないファイルなどを作ることができる。また、何人かでグループを作りファイルを特定のグループの人だけが読んだり書いたりすることもできる。あるファイルを誰が読むことができ、だれが書

くことができるか、だれが実行することができるかの設定情報を、許可属性 (パーミッション permission) という。

Mac OS X では所有者とグループを所有権、パーミッションをアクセス権とよぶ。Finder で所有権とアクセス権を見ることができる。Finder で他人のディレクトリを表示したとき、中に入ることでできないディレクトリは書類のアイコンの右下に進入禁止の印 (赤い円に白地の -) がついている。

UNIX では複数のユーザーが属すグループというものが設定されている。すべてのユーザーは、少なくとも一つのグループに属している。自分の属しているグループは、groups または id -p コマンドで調べることができる。

```
$ groups user-name
```

`user-name` を省略するとコマンドを実行したユーザーの属するグループが表示される。

ファイルやディレクトリには一人の所有者と一つのグループが設定されている。UNIX ではあるユーザが新しくファイル (ディレクトリも含む) を作る (メールを受け取ったときも作ったことになる) と、そのファイルの所有者はユーザ自身、グループはそのユーザの属すグループに自動的に設定される。

ファイルにはアクセスの可否が、所有者、グループ、その他、に対し決められている。

表 4: パーミッションの意味

	ファイル	ディレクトリ
<code>read</code>	読める (cp や cat ができる)	中を見ることができる (<code>ls</code> できる)
<code>write</code>	書き換えられる	中のファイルを削除すること、新しいファイルを作ることができる
<code>execute</code>	実行できる	中に入るることができる (<code>cd</code> できる)

`ls -l` コマンドにより、ファイルの型、パーミッション、リンク数、所有者、グループ、大きさ (単位はバイト)、最終更新日時 (最終更新年月日)、ファイル名またはディレクトリ名、が表示される。

パーミッションは 2 文字目から 10 文字目までで、2-4 文字が所有者、5-7 文字がグループ、8-10 文字がその他のアクセス権を表わしている。

例 4.1 ~/comp2d の一部

```
-rw-r--r--  1 asakawa  cis      45 Oct 17 14:39 file1.txt
drwxr-xr-x  2 asakawa  cis      80 Oct 17 14:39 exerdir
```

を取り説明する。

1. `file1.txt` は通常のファイルで、所有者は `asakawa`、所属グループは `cis` である。`asakawa` は読むことも書くこともできるが、所有者以外は読むことだけができる。

r	w	-	r	-	-	r	-	-
可	可	不可	可	不可	不可	可	不可	不可
所有者			同じグループ			その他		

2. `exerdir` はディレクトリで、所有者 `asakawa` は中を見る (`ls` する) ことも、新しいファイルを作ることも、中のファイルを削除することも、ディレクトリの中に入る (`cd` する) ことも出来る。所属グループとその他は、中を見る (`ls` する) こと、中に入る (`cd` する) ことのみ許可されている。

```
$ cd ~asakawa/comp2d
$ ls -l
```

してみよ。

注 4.1 あるファイルを消去できるか否かは、そのファイルの許可属性とは無関係に、そのファイルが置かれているディレクトリの許可属性により決まる。

4.5.1 グループの変更

ファイルの所有者は、そのファイルの所属グループを `chgrp` コマンドによりユーザーの所属する別のグループに変更することができる。

```
$ chgrp group-name file-name
```

CHange GRouP

ここで、`group-name` は新しい所属グループ、`file-name` は、絶対パスあるいは相対パスで表したファイルまたはディレクトリ名である。

4.5.2 パーMISSIONの変更

許可属性の変更は `chmod` コマンドによる。

```
$ chmod mode file-name
```

CHange MODe

`mode` には、記号を用いる方法 (シンボリックモード) と 8 進数を用いる方法 (絶対モード) がある。

記号を用いる方法は以下のとおり

ユーザ		パーMISSION	
u	所有者 (user)	=	パーMISSIONを設定
g	グループ (group)	+	パーMISSIONを追加
o	その他 (other)	-	パーMISSIONを削除
a	全員 (all)		

例えば、`myfile` の許可属性を `rw-----` としたいとき、

```
$ chmod u=rw myfile
$ chmod go= myfile
```

とする。myfile の許可属性が `rw-r--r--` のとき、`rw-----` に変更したい場合は、

```
$ chmod go-r myfile
```

8 進数を用いる方法は以下のとおり。

許可属性は、“r”、“w”、“x” にそれぞれ 4、2、1 を対応させて加えた 0 から 7 までの数字を 3 つ並べ表す。

---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
rw-	6
rwX	7

例えば、myfile の許可属性が `rw-r--r--` のとき、`rw-----` に変更したい場合は、

```
$ chmod 600 myfile
```

演習 4.4 1. `~/comp2d` に中身のないファイル `testfile` を作れ。

2. `~/comp2d/testfile` の許可属性を調べよ。
3. `~/comp2d/testfile` の許可属性を全員が読めるだけで書込できないように変更せよ。
4. `~/comp2d/testfile` を削除してみよ。どのようなメッセージが表示されるか。