

第1章 MacOS

1.1 MacOSのファイルシステムとコマンド

1.1.1 ファイル名

ファイル (file) : 情報のかたまり

フォルダ (folder) : ファイルをまとめるための特殊なファイル

MacOSではフォルダ(ディレクトリ)によってファイルを管理できる。FinderなどのGUIを用いる際は「フォルダ」と呼ばれるが、歴史的には「ディレクトリ(directory)」と呼ばれてきた。

ファイル名 : ファイルに付けられた名前

MacOSではファイル名に漢字も利用できるが、ターミナルウィンドウでの利用や他の計算機への移動を考えると、半角の英数字に限定しておくほうが無難

拡張子 : ファイル名の最後の「.」から後の部分

ファイルの種類を反映していることが多い

.txt 文書ファイル

.doc MSWordのファイル

.ppt PowerPointのファイル

.xls Excelのファイル

.java Java言語のプログラムファイル

(多くの場合) 拡張子に対してアプリケーションが対応づけられており、対応づけられたアプリケーションによって開かれる

1.1.2 Finder(ファインダ)

Finder(ファインダ) : MacOS上でファイルシステムを管理するためのGUIベースのアプリケーション

ドック(dock)から起動する。対応するアイコンをクリックするとファイル/フォルダを選択し、ダブルクリックするとファイル/フォルダを開くことができる。

サイドバー : Finderウィンドウの左端の欄

デスクトップ、ホーム、アプリケーションなどが表示されている

パスバー：Finder ウィンドウの下の段

絶対パスが表示される（メニューから「表示 → パスバーを表示」を選択する）

環境設定：応用プログラムの調整機能

「Finder → 環境設定」から起動する

拡張子の表示：環境設定の「詳細」で「…拡張子を表示」にチェックを入れる

ファイル操作：「開く」「情報を見る」「移動」「コピー」「削除」など

ファイル（フォルダ）を選択したうえで、右ボタンをクリックするとメニューが現れる

ホーム (home)：ユーザ固有のフォルダ

ユーザのファイルやフォルダはホームから下に作成される

デスクトップ (Desktop)：デスクトップ環境を反映するフォルダ

ホーム直下のフォルダの1つで、このフォルダ内のファイルやフォルダはバックグラウンドに表示される

1.1.3 ディレクトリ (directory) とパス (path)

CUI ベースのコマンドでファイルを指定する場合には、ディレクトリの表記法に慣れておく必要がある。^{*}

ディレクトリ：ファイルをまとめるための見出し専用のファイル

異なったディレクトリにあると、同じ名前でも違うファイルとなりうる。

例：図 1.1 のディレクトリ '理工学部' や通常ファイル '電気'

サブ (下位) ディレクトリ：ディレクトリのさらに下にあるディレクトリ

上位ディレクトリ：ディレクトリの上にあるディレクトリ (記号：'..')

階層ディレクトリ：上位ディレクトリ，サブディレクトリによって作られる階層構造

ルートディレクトリ (root directory)：階層構造の最上部 (記号：'/')

カレントディレクトリ (current directory)：現在注目しているディレクトリ (記号：'./')

ホームディレクトリ (home directory)：各ユーザに割り当てられたディレクトリ (記号：'~/')

ターミナルウィンドウを起動した時にはまずホームディレクトリに訪れる。個々のユーザのホームディレクトリは '~ユーザ名' となる。

パス, パス名 (path)：個々のファイルを指定するための名前

目的のファイルまでの道筋を示す。ディレクトリ名の区切りには '/' を用いる。

絶対パス (absolute path)：ルートディレクトリを起点としたパス

必ず '/' から始まる。

相対パス (relative path)：カレントディレクトリを起点としたパス

サブディレクトリ名 ないし './' から始まる。

^{*}1.3 節の「ファイル関連のコマンド」で実際に試してみると良い。

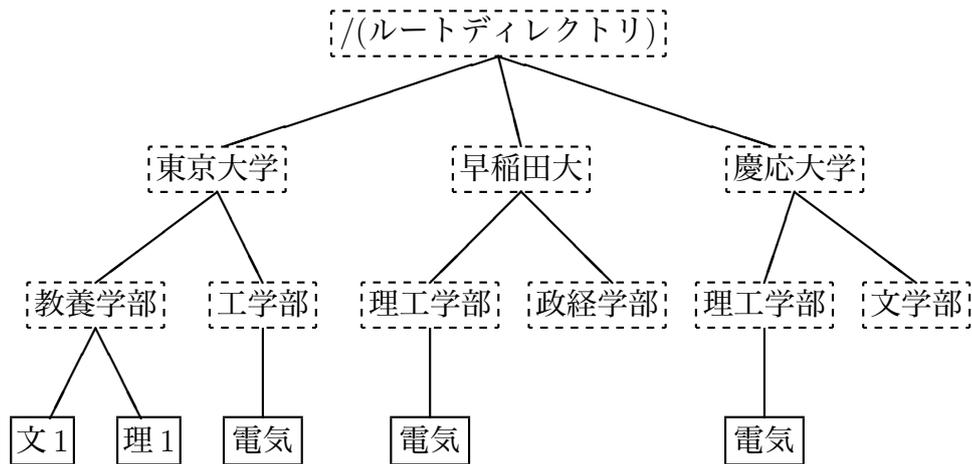


図 1.1: 階層ディレクトリ構造*

絶対パス	相対パス
/	../..
/東京大学	..
/東京大学/工学部	.
/東京大学/工学部/電気	電気
/早稲田大/理工学部/電気	../../早稲田大/理工学部/電気

図 1.2: 絶対パスと相対パス (カレントディレクトリ: /東京大学/工学部)

1.2 MacOS のコマンド

MacOS においては CUI ベースのインタラクションも可能になっている。ターミナルプログラムからコマンドによって各種のプログラムを実行できる。

1.2.1 コマンド入力

コマンド: プログラム起動のための命令

シェル: MacOS のコマンドを入力して実行するプログラム

シェルプログラムが実行されておりコマンド入力ができる状態

プロンプト: 入力促進記号

カーソル: 入力位置の表示記号

コマンド名: コマンドの名称*

オプション: コマンドの機能を微妙に変更する指定 ('-' で始まる)

*MacOS ではコマンド探索パス (シェル変数: path) に存在する実行可能なプログラムファイルのファイル名がコマンド名となる。

パラメータ：コマンド入力に際して与えられる付加的な情報

エラーメッセージ：コマンドなどに誤りがあった際に表示されるメッセージ

コマンドの入力形式は以下のとおり。ただし、オプションの有無やパラメータの個数はコマンドごとに異なる

```
pc123456m:~ 1234567890$ コマンド名 [-オプション] パラメータ 1 パラメータ 2... ¶
```

情報教育棟でのプロンプトは、「計算機名:ディレクトリ名 ユーザ名\$」という形式をしている。最後の¶はリターンキー (<Return>) を意味するものとする。ターミナルウィンドウ上で clear (clear screen, 画面消去) コマンドなどを試してみる。

```
pc123456m:~ 1234567890$ jjj ¶
bash: jjj: command not found                (エラーメッセージ)
pc123456m:~ 1234567890$ cp ¶
usage: cp [-R [-H | -L | -P]] [-fi | -n] [-apvX] source_file target_file
       cp [-R [-H | -L | -P]] [-fi | -n] [-apvX] source_file ... target_directory
pc123456m:~ 1234567890$ clear ¶
pc123456m:~ 1234567890$ date ¶
2015年 5月 1日 金曜日 16時 38分 39秒 JST    (日付と時刻の表示)
pc123456m:~ 1234567890$
```

1.3 ファイル関連のコマンド

ls (list)：ファイルの一覧と情報を表示する

より詳しい情報を見るなら '-l' オプションをつける。パーミッション、リンク数、所有者、グループ、ファイルサイズ、最終変更日時などがわかる。

(ここではもともと AppData, Desktop, Documents, Library, Mail, Movies, Music, Pictures, Public, Sites などのディレクトリ/フォルダがある)

```
pc123456m:~ 1234567890$ ls ¶
AppData          Mail              Public
Desktop          Movies            Sites
Documents        Music             windows
Library          Pictures          お気に入り
```

pwd (print working directory)：カレントディレクトリを表示する

```
pc123456m:~ 1234567890$ pwd ¶
/home/1234567890                (ログイン直後はホームディレクトリが
                                カレントディレクトリとなる)
```

cd (change directory)：カレントディレクトリを変更する

パラメータに変更先のディレクトリのパスを指定する。パラメータを省略するとホームディレクトリに戻る。

```

pc123456m:~ 1234567890$ cd .. ㊦          (1つ上に上がる)
pc123456m:home 1234567890$ pwd ㊦
/home
pc123456m:home 1234567890$ cd .. ㊦      (更に1つ上がる)
pc123456m:/ 1234567890$ pwd ㊦          (ルートディレクトリに行く)
/
pc123456m:/ 1234567890$ cd .. ㊦        (これ以上は上がれない)
pc123456m:/ 1234567890$ pwd ㊦
/
pc123456m:/ 1234567890$ ls ㊦
Applications
Developer
: (途中省略)
usr
var
ユーザーズガイドと情報
pc123456m:/ 1234567890$ cd usr ㊦        (/usrに行く)
pc123456m:usr 1234567890$ pwd ㊦
/usr
pc123456m:usr 1234567890$ ls ㊦
X11          etc          libexec      sbin
X11R6        include      llvm-gcc-4.2 share
bin          lib          local        standalone
pc123456m:usr 1234567890$ cd /home ㊦    (/homeに行く)
pc123456m:home 1234567890$ pwd ㊦
/home
pc123456m:home 1234567890$ cd ㊦        (ホームディレクトリに戻る)
pc123456m:~ 1234567890$ pwd ㊦
/home/1234567890

```

mkdir (make directory) : ディレクトリを作成する

```

pc123456m:~ 1234567890$ pwd ㊦          (カレントディレクトリを確認)
/home/1234567890
pc123456m:~ 1234567890$ cd Desktop ㊦    (Desktopに行く)
pc123456m:Desktop 1234567890$ pwd ㊦    (カレントディレクトリを確認)
/home/1234567890/Desktop
pc123456m:Desktop 1234567890$ ls ㊦      (ファイルの一覧)
pc123456m:Desktop 1234567890$ mkdir mydir1 ㊦ (mydir1を作る)
pc123456m:Desktop 1234567890$ ls -l ㊦   (mydir1ができた)
total 0
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir1

```

```
(ディレクトリ, 所有者は全て (rwx) 可能, グループとその他は全て不可 (---))
(所有者は 1234567890, グループは teacher, ファイルサイズは 68 バイト)
pc123456m:Desktop 1234567890$ cd mydir1 ¶ (mydir1 に行く)
pc123456m:mydir1 1234567890$ pwd ¶ (カレントディレクトリを確認)
/home/1234567890/Desktop/mydir1
pc123456m:mydir1 1234567890$ ls -l ¶ (mydir1 はまだ空)
pc123456m:mydir1 1234567890$ mkdir mydir2 ¶ (mydir2 を作る)
pc123456m:mydir1 1234567890$ ls -l ¶ (mydir2 ができた)
total 0
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir2
pc123456m:mydir1 1234567890$ cd mydir2 ¶ (mydir2 に行く)
pc123456m:mydir2 1234567890$ pwd ¶ (カレントディレクトリを確認)
/home/1234567890/Desktop/mydir1/mydir2
pc123456m:mydir2 1234567890$ ls -l ¶ (mydir2 はまだ空)
pc123456m:mydir2 1234567890$ cd .. ¶ (1つ上のディレクトリに行く)
pc123456m:mydir1 1234567890$ pwd ¶ (mydir1 に戻っている)
/home/1234567890/Desktop/mydir1
```

cat (concatenate) : ファイルの内容を表示する (パラメータにファイル名を与える)
 文書ファイルでないとちゃんと読めない。

```
pc123456m:mydir1 1234567890$ ls -l ¶
total 0
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir2
```

(emacs で ~/Desktop/mydir1/renshu 作製, File->Visit New File, File->Save)
 (たとえば「練習用のファイル」という内容にする)

```
pc123456m:mydir1 1234567890$ ls -l ¶ (通常ファイルとディレクトリ)
total 8
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir2
-rw----- 1 1234567890 teacher 25 4 18 14:29 renshu
```

(ファイル名補完を使ってみよう。たとえば, ren<TAB>と打ってみる。)

```
pc123456m:mydir1 1234567890$ cat renshu ¶ (ファイルの内容を見る)
練習用のファイル
pc123456m:mydir1 1234567890$
```

ls : 指定したディレクトリのファイルの一覧を表示する
 パラメータをつけることで対象となるディレクトリを指定できる。

```
pc123456m:mydir1 1234567890$ cd ~ ¶ (ホームディレクトリに戻る)
```

```
pc123456m:~ 1234567890$ pwd ¶
/home/1234567890
pc123456m:~ 1234567890$ ls -l Desktop/mydir1 ¶ (~/Desktop/mydir1の内容)
total 8
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir2
-rw----- 1 1234567890 teacher 25 4 18 14:29 renshu
pc123456m:~ 1234567890$ ls -l Desktop/mydir1/mydir2 ¶
pc123456m:~ 1234567890$                               (~/Desktop/mydir1/mydir2は空)
```

mv, cp (move, copy) : ファイルを移動・複製する (第2パラメータがディレクトリの場合)
 第1パラメータで複製元のファイルのファイル名 (パス) を指定する
 第2パラメータが移動・複製先のディレクトリ (パス) を指定する

```
pc123456m:~ 1234567890$ cd Desktop ¶                (Desktopに行く)
pc123456m:Desktop 1234567890$ ls -l ¶              (~/Desktopの内容)
total 0
drwx----- 4 1234567890 teacher 136 4 18 14:29 mydir1
pc123456m:Desktop 1234567890$ mv mydir1/renshu . ¶ (renshuをDesktopに移動)
pc123456m:Desktop 1234567890$ ls -l mydir1 ¶      (mydir1にrenshuはない)
total 0
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir2
pc123456m:Desktop 1234567890$ ls -l ¶              (Desktopにrenshuがある)
total 8
drwx----- 3 1234567890 teacher 102 4 18 14:31 mydir1
-rw----- 1 1234567890 teacher 25 4 18 14:29 renshu
pc123456m:Desktop 1234567890$ cp renshu mydir1 ¶   (renshuをmydir1に複製)
pc123456m:Desktop 1234567890$ ls -l ¶              (Desktopにrenshuがある)
total 8
drwx----- 4 1234567890 teacher 136 4 18 14:32 mydir1
-rw----- 1 1234567890 teacher 25 4 18 14:29 renshu
pc123456m:Desktop 1234567890$ ls -l mydir1 ¶      (mydir1にもrenshuがある)
total 8
drwx----- 2 1234567890 teacher 68 4 18 14:28 mydir2
-rw----- 1 1234567890 teacher 25 4 18 14:32 renshu
```

mv, cp : ファイル名を変更する, 通常ファイルを複製する (ディレクトリや特殊ファイルは不可)
 第1パラメータで (複製) 元のファイル名 (パス) を指定する
 第2パラメータで変更/複製後のファイル名 (パス) を指定する

```
pc123456m:Desktop 1234567890$ ls ¶                (ファイル名の確認)
mydir1      renshu
pc123456m:Desktop 1234567890$ mv renshu gomi ¶   (ファイル名を変える)
```

```

pc123456m:Desktop 1234567890$ ls ¶ (ファイル名が変わった)
gomi          mydir1
pc123456m:Desktop 1234567890$ cat gomi ¶ (ファイルの内容は同じ)
練習用のファイル
pc123456m:Desktop 1234567890$ mv mydir1 newname ¶ (ディレクトリ名も変更可能)
pc123456m:Desktop 1234567890$ ls -l ¶ (ディレクトリ名が変わった)
total 8
-rw-----  1 1234567890  teacher   25  4 18 14:29 gomi
drwx-----  4 1234567890  teacher  136  4 18 14:32 newname
pc123456m:Desktop 1234567890$ cp gomi junk ¶ (ファイルを複製する)
pc123456m:Desktop 1234567890$ ls ¶ (ファイルが1個増えた)
gomi          junk          newname
pc123456m:Desktop 1234567890$ cat junk ¶ (ファイルの内容は同じ)
練習用のファイル
pc123456m:Desktop 1234567890$

```

rm (remove) : 通常ファイルを消去する

ディレクトリが指定された場合にはエラーになる

rmdir (remove directory) : ディレクトリを消去する

カラのディレクトリでないと消去できない。

```

pc123456m:Desktop 1234567890$ rm junk ¶ (ファイルを消去する)
pc123456m:Desktop 1234567890$ ls ¶ (ファイルは2個になった)
gomi          newname
pc123456m:Desktop 1234567890$ cd newname ¶ (newname に行く)
pc123456m:newname 1234567890$ ls ¶ (ファイルは2個ある)
mydir2        rensu
pc123456m:newname 1234567890$ rm mydir2 ¶ (ディレクトリは rm で消去不可)
rm: mydir2: is a directory
pc123456m:newname 1234567890$ rmdir mydir2 ¶ (ディレクトリは rmdir で消去)
pc123456m:newname 1234567890$ cd .. ¶ (1つ上の Desktop に行く)
pc123456m:Desktop 1234567890$ rmdir newname ¶ (空でないディレクトリは消去不可)
rmdir: newname: Directory not empty
pc123456m:Desktop 1234567890$ ls newname ¶ (ファイルを確認)
renshu
pc123456m:Desktop 1234567890$ rm newname/renshu ¶ (newname/renshu を消去)
pc123456m:Desktop 1234567890$ rmdir newname ¶ (newname を消去)
pc123456m:Desktop 1234567890$

```