

# 情報科学演習 / データベース入門 A 資料 1

## UNIX のコマンドとファイル

平成 29 年 4 月 10 日

### 目次

1	演習用コンピュータの利用開始と終了	1
2	コマンドの実行	1
2.1	コマンドの実行	1
3	困ったときには	2
4	ファイル	3
4.1	ファイルを作ってみよう — ls, cat, >, >>	3
4.2	ファイル名の付け方	4
4.3	ファイル名の変更とファイルの複写・削除 — mv, cp, rm	5
4.4	コマンドのオプション	6
5	ディレクトリとその操作	6
5.1	ディレクトリの作成 — mkdir	7
5.2	ディレクトリ名の変更とディレクトリ削除 — mv, rmdir	7
6	ホームディレクトリとカレントディレクトリ	7
6.1	ホームディレクトリ	7
6.2	カレントディレクトリ	8
6.3	カレントディレクトリの変更 — cd	8
7	練習問題	8
7.1	その 1	8
7.2	その 2	9
8	コマンド一覧	10

# 1 演習用コンピュータの利用開始と終了

この授業では、Windows が動作しているコンピュータから、UNIX<sup>1</sup> が動作しているコンピュータに遠隔ログイン (remote login) して演習を行います。

UNIX では、コンピュータの利用開始の手続きを、ログイン (login) と呼びます。遠隔ログイン (remote login) とは、目の前 (local) のコンピュータから、ネットワークで接続されている遠隔 (remote) のコンピュータにログインすることです。

本校の教育用コンピュータから UNIX 機に遠隔ログインする方法については、別の資料を参照してください。ログアウトするときには exit コマンドを実行してください。

## 2 コマンドの実行

UNIX が搭載されているコンピュータの操作の基本は、キーボードからコマンドを入力して実行することです。本章では、コマンド実行に関わる基本事項を学びます。

### 2.1 コマンドの実行

- コマンド (command) はコンピュータへの命令です。
- UNIX にログインして表示される

```
[user@host ~]$
```

をプロンプト (prompt) と呼びます<sup>2</sup>。プロンプトが存在する行をコマンド行 (command line) と呼びます。

- プロンプトが現れれば、キーボードからコマンドを入力できます。入力の誤りは <BS> (Backspace キー) で修正できます。続いて <ENTER> (エンター・キーまたはリターン・キー) を押せば、入力したコマンドが実行されます。

例) date コマンドで現在の日付と時刻を表示する

```
date<ENTER>
```

- UNIX のコマンドにおける大文字と小文字は、異なるものとして区別されます<sup>3</sup>。
- ほとんどのコマンドには、引数 (ひきすう; argument) を使って、コマンドの動作に関わる指示を与えることができます。引数が必須のコマンドもあります。

引数は、コマンド行に入力したコマンドの名前に続いて、空白で区切って記述します。

引数として記述すべき事柄はコマンド毎に異なります。

例) 引数 +%D を付けて date コマンドを実行する

```
date +%D<ENTER>
```

<sup>1</sup>実際は Linux という OS (オペレーティングシステム) を使います。これは UNIX 系 OS と呼ばれるものの一つです。

<sup>2</sup>プロンプトの表示形式は設定や状況によって変わります。利用者が設定を変更することも可能です。

<sup>3</sup>Windows のコマンドには、大文字・小文字の区別がありません。

### 2.1.1 練習

1. `cal` コマンドを実行してください。今月のカレンダーが表示されましたか。
2. 大文字で `CAL` と入力して `<ENTER>` を押してください。Shift キーを押しながら英字を入力すると大文字になります。

「コマンドが見つかりません (Command not found.)」としかられますね。UNIX に `cal` コマンドはありますが、`CAL` コマンドはありません。

3. `cal 10 1966` を実行してください。10 や 1966 は、`cal` コマンドの引数です。

`cal` コマンドは、引数が無ければ今月の、引数があれば指定の月あるいは年のカレンダーを表示します。

4. 途中に空白を入れずに `cal101966` を実行してください。

`cal101966` というコマンドはありません。コマンドと引数、および複数の引数を区切るには空白が必要です。

5. `man date` を実行してください。man はコマンド等のマニュアルを表示するコマンドです。

表示が一画面におさまりきらないため、最終行に何やら表示されて止まっています。 `<SPACE>` (スペース・キー) を押すと次の画面に進みます。q を打つと終了します。

6. `man` を実行してください。

`man` コマンドには、調べたいコマンドの名前 (コマンド名) 等を、引数として与える必要があります。引数が必要かどうかは、コマンドによって異なります。

7. `clear` コマンドを実行してみましょう。

きれいになりましたか。

8. `echo` コマンドを実行してみましょう。

何も表示されませんね。

9. `echo I love you.` を実行してみましょう。

`echo` は与えられた引数を画面に出力 (表示) します。

## 3 困ったときには

- キーボードから入力できない。

`CTRL-q` を押してみてください。CTRL-q は Ctrl キーを押しながら q キーを押すことを意味します。

- プロンプトが出ない。

CTRL-d を押してみてください。それでもだめなら CTRL-c を押してみてください。CTRL-c は現在実行しているコマンドを強制終了します。

- 英字のキーを押すと大文字で入力される。Shift キーと一緒に押せば小文字になる。

キーボードの Caps Lock ランプが点灯していれば、大文字入力になります。Caps Lock キー、または Shift キーと同時に Caps Lock キーを押して、ランプを消灯させてください。

- exit コマンドでログアウトしようとするとき「中断した job が残っています」と表示されてログアウトできない。

fg とタイプしてエンターを押すと、中断していた job (プログラム) が画面に現れ動作し始めます。それを正しい方法で終了させてからログアウトを試してください。

## 4 ファイル

コンピュータで作業したことを残しておきたいときにはファイル (file) を利用します。ファイルを作るには、その内容に加えて、ファイルの名前 (ファイル名) が必要です。ファイルはファイル名で区別され、ファイル名を使ってファイルの内容を見たり、内容を変更できます。

ファイルの作成や利用の方法はいろいろありますが、この章では、コマンド行の操作だけでファイルを扱います。

### 4.1 ファイルを作ってみよう — ls, cat, >, >>

- ls コマンドは所持するファイルの名前 (ファイル名) を表示するコマンドです。
- 「cat ファイル名」を実行すれば、ファイルの内容が表示されます<sup>4</sup>。
- コマンド [と引数] の後ろに「> ファイル名」を指定すると、コマンドの実行結果は画面に表示されません。その代わりに、指定したファイル名の新しいファイルができ上がり、そこに結果が書き込まれます。
- コマンド [と引数] の後ろに「>> 既存ファイルの名前」を指定すると、コマンドの実行結果は、指定したファイルに追加されます。

では、実際に試してみましょう。

1. ls コマンドを実行してみましょう。

ファイルはありますか？

2. date コマンドを実行してみましょう。

現在の時刻が表示されましたね。

---

<sup>4</sup>cat コマンドで内容を表示できるのは、テキストファイルと呼ばれるファイルのみです。

3. `date > now` を実行しましょう。

`date` コマンドの実行結果は画面に表示されません。その代わりに、`now` という名前のファイルに書き込まれました。

4. 再度、`ls` を実行してみましょう。

`now` という名前のファイルがありますね。

5. `cat now` を実行して、`now` の内容を見てみましょう。

#### 4.1.1 練習: ファイルへの内容追加など

1. `date >> now` を実行してください。続いて `ls` と `cat now` を実行しましょう。

`date` コマンドの結果が追加されて、`now` の内容が 2 行になりました。

2. 再度 `date > now` を実行してから、`ls` と `cat now` を実行しましょう。

元の `now` の内容は無くなってしまいましたね。>>> はコマンドの実行結果をファイルに追加するのに対し、> はファイルの内容を新しい結果で置き換えます。

3. `now cat` を試してみましょう。

しかられましたね。一般に、プロンプトに続いては、ファイル名 (`now`) ではなく、コマンド名 (`cat`) を打つ必要があります<sup>5</sup>。

4. `cat dog` を実行してみましょう。

`cat` コマンドの引数には、既存のファイルの名前を与えなければいけません。

5. 引数を与えずに `cat` を実行してみましょう。

プロンプトが現れませんね。第 3 章を読んで、正しい対処をしてください。

## 4.2 ファイル名の付け方

- UNIX では / をファイル名に含めることはできません。
- 小文字と大文字は区別されます<sup>6</sup>。特に理由がなければ、小文字のファイル名を使うのが慣例です。
- UNIX では、.(ドット;ピリオド)で始まる名前のファイルは、多くのコマンドで特別に扱われます。

当面、英数字と.(ドット),\_(アンダーバー)のみを組み合わせたファイル名の利用を推奨します。ただし、ドットで始まるファイル名は避けるのが無難です。

ファイルはファイル名で識別して操作するものですから、ファイル名はファイルの内容を連想できるものにしましょう。ドットやアンダーバーはファイル名を構成する単語を区切るために使われます。

<sup>5</sup>例外はありますが、細かなことは無視しましょう。

<sup>6</sup>Windows のファイルシステムでは、ファイル名の大文字・小文字は区別されません。そのため、大文字・小文字のみが異なる名前のファイルを、同じフォルダに作ることはできません。

### 4.2.1 練習

1. `date > a/b` を実行してみましょう。

エラーメッセージが表示されましたね。エラーの意味は、後でディレクトリの学習をすればわかります。

2. `ls` を実行しましょう。続いて `date > Now` を実行して、`ls` を実行しましょう。

`now` と `Now` は区別されていますか。

3. `now` と `Now` の内容を見比べてください。

4. `cal` を実行してください。次に `cal > this_month` を実行してから、ファイルができたか、確認してください。

## 4.3 ファイル名の変更とファイルの複写・削除 — `mv`, `cp`, `rm`

コマンドを使って、作成したファイルの名前を変更したり、ファイルの内容を別のファイルに複写したり、ファイルを削除することができます。そのためのコマンドは `mv`, `cp`, `rm` です。ファイル名を引数として次の形式で使います。*file1* や *file2* などの *italic* (斜体) の表記は、コマンド実行の際に適当な語 (ここではファイル名) に置き換える必要があることを意味します。

`mv file1 file2` *file1* の名前 (ファイル名) を *file2* に変更

`cp file1 file2` *file1* の内容を *file2* に複写 (*file1* と同じ内容を持つファイル *file2* を作る)

`rm file` *file* で指定したファイルを削除

### 4.3.1 注意

- 削除したファイルは二度と戻りません。`rm` コマンドを実行するときには心を落ち着かせてから `<ENTER>` を押すことを習慣づけましょう<sup>7</sup>。
- `cp` や `mv` の最後の引数 (上で *file2* と表記したもの) に既存のファイルの名前を指定すると、元のファイルは上書きされて無くなります<sup>8</sup>。このことは「コマンド > ファイル名」でファイルにコマンドの実行結果を書き込むときも同様です<sup>9</sup>。誤って必要なファイルを消さないように、既存のファイルをよく確認してから実行しましょう。

### 4.3.2 練習

1. 再び `ls` を実行してから `cat now` を実行しましょう。

2. ファイル `now` の名前を `old` に変更します。`mv now old` を実行してください。名前が変わったことを `ls` コマンドで確認してください。`cat` コマンドでファイルの内容も見てください。

<sup>7</sup>`rm` 実行時、本当にファイルを削除してよいか、確認のメッセージを出すように設定しているシステムもあります。

<sup>8</sup>`cp` や `mv` の実行時に、既存のファイルを上書きしてよいかどうか、確認のメッセージを出すように設定されているシステムもあります。

<sup>9</sup>設定によっては上書きされないこともあります。

3. old と同じ内容のファイル old.copy を作ります。cp old old.copy を実行してください。ファイルができたか ls で確認してから、old と old.copy の内容を見比べてみましょう。
4. old.copy を削除します。rm old.copy を実行してください。うまくいったか確認してください。
5. 続いて old と this\_month を削除してください。結果を確認してください。

## 4.4 コマンドのオプション

多くのコマンドには、オプション (option) と呼ばれる、コマンド実行に必須ではない引数を指定できます。オプションは、コマンド動作の詳細を指定するもので、通常、- か -- に続いて指定します。各コマンドのオプションは man コマンドを使って調べることができます。

例) ls -a

ls のオプション -a は . (ドット) で始まるものを含めて、すべてのファイルの名前を表示するためのものです。

注意) ここで表示される . (ドット) で始まる名前のファイルは、コンピュータを正常に使うために必要なファイルです。削除したり、名前を変えたりしないでください。

### 4.4.1 練習

1. cal > month を実行してから、cat month を実行してください。

ファイル month の内容が表示されましたね。

2. cat -n month を実行してみましょう。

cat は -n というオプションを受け付けます。-n は行番号を付して表示するために使います。

3. ls を実行してから、ls -l (エル) を実行してみましょう。

ls コマンドのオプション -l は、ファイルのいろいろな情報を表示する指定です。ここでは、ファイルの最終変更日時がわかることにだけ、注目することにしましょう。

4. ls -a -l と ls -al の実行結果を比較してください。

## 5 ディレクトリとその操作

Windows におけるフォルダに相当するものを、UNIX ではディレクトリ (directory) またはディレクトリファイル (directory file) と呼びます。ディレクトリはファイルを格納する器です。ディレクトリを利用することで、様々なファイルを、その用途別に分類して整理することができます。

ディレクトリファイルという名称から予想できるとおり、ディレクトリもファイルの一種です。これまでに扱ってきたファイルを「通常のファイル」と呼んで、ディレクトリではないことを明示することがあります。

まず、ディレクトリの作成や削除、ディレクトリ名変更の方法を学びましょう。ファイルと同じくディレクトリにも名前（ディレクトリ名）が必要です。ディレクトリ名の付け方はファイル名と同様です。

## 5.1 ディレクトリの作成 — mkdir

ディレクトリを作成するには `mkdir` (make directories) コマンドを使います。引数には作成するディレクトリの名前を指定します。

1. `tmpdir` という名前のディレクトリを作成してください。

```
mkdir tmpdir
```

2. ディレクトリ `tmpdir` ができたかどうか確認します。

ディレクトリもファイルの一種なので、`ls` コマンドで名前を表示できます。`ls` を実行してください。

3. 通常のファイルとディレクトリが同じように表示されるのは不便ですね<sup>10</sup>。両者を区別したいときには、`ls` にオプション `-F` を指定します。

```
ls -F
```

ディレクトリの場合にはディレクトリ名の後に `/` が付加されます。ただし、この `/` はディレクトリであることを表す印であって、ディレクトリ名の一部ではありません。

4. ディレクトリに対して `cat` コマンドを実行したらどうなるでしょう。`cat tmpdir` を試してみましょう。`cat` コマンドは、通常のファイルの内容を表示するコマンドでしたね。

## 5.2 ディレクトリ名の変更とディレクトリ削除 — mv, rmdir

ディレクトリ名の変更には、ファイル名変更と同様に `mv` を使います。ディレクトリの削除は `rmdir` (remove directories) です。

1. `mv tmpdir temp` を実行して、`tmpdir` の名前を `temp` に変更してください。`ls` コマンドで結果を確認してください。
2. `temp` を削除します。`rmdir temp` を実行してください。結果を確認してください。

# 6 ホームディレクトリとカレントディレクトリ

## 6.1 ホームディレクトリ

UNIX で利用できるファイルやディレクトリは、必ずいずれかのディレクトリ内にあります<sup>11</sup>。これまでに幾つかのファイルを作成しましたが、それらはユーザー毎に予め用意されたディレクトリ内に作成していたのです。そのディレクトリをユーザーのホームディレクトリ (home directory) と呼びます。ホームディレクトリを記号 `~` で表すことがあります。

<sup>10</sup>ファイルの種類を色で区別するように設定されていることもあります

<sup>11</sup>ルートディレクトリという特別なディレクトリを除く。



## 6.2 カレントディレクトリ

コマンド操作でファイルやディレクトリを扱うときに必要な概念として、カレントディレクトリ (current directory; current は「現在の」、「現行の」という意味)<sup>12</sup> があります。カレントディレクトリは文字通り、現在使っているディレクトリを意味する用語です。

例えば、`ls` を引数無しで実行して表示されるのは、カレントディレクトリに存在するファイルの名前です。また、`cp` や `mkdir` 等のコマンドにファイル名やディレクトリ名を与えると、コマンドが処理するのはカレントディレクトリ内のファイルやディレクトリとなります<sup>13</sup>。

ユーザーがログインしたときのカレントディレクトリはホームディレクトリです。これまで、ファイルやディレクトリがホームディレクトリに作成されたのはそのためです。

## 6.3 カレントディレクトリの変更 — `cd`

ホームディレクトリ (`~`) 以外のディレクトリに存在するファイルやディレクトリを、これまでと同様にファイル名やディレクトリ名で操作するには、カレントディレクトリの変更の操作を行います。

カレントディレクトリを変更するコマンドは `cd` (change directory) です。`cd` の引数には変更先のディレクトリを指定します。引数を指定しなければ、ホームディレクトリがカレントディレクトリになります (ホームディレクトリに戻れます)。

`cd` の引数にディレクトリ名を指定する際にも、そのディレクトリはカレントディレクトリに存在していなければなりません。他の場所にあるディレクトリを `cd` の引数に与える方法は後の資料で扱います。

### 6.3.1 例

1. カレントディレクトリにディレクトリ `tmp` を作成 — `mkdir tmp`
2. 確認 — `ls -F`
3. カレントディレクトリを `tmp` に変更 — `cd tmp`

何も表示されなければ成功です。プロンプトにカレントディレクトリ名が表示される設定がなされていれば、プロンプトが変わります。

4. カレントディレクトリをホームディレクトリに変更 — `cd`

## 7 練習問題

### 7.1 その1

1. どのような名前のファイルがあるかを確認してください。
2. ファイル `Now` を削除しましょう。うまくいったか確かめましょう。

<sup>12</sup>ワーキングディレクトリ (working directory) や作業ディレクトリと呼ぶこともあります。

<sup>13</sup>カレントディレクトリ以外のファイル操作については後で学びます。

3. 現在の日付と時刻が入っているファイル `Now` を再び作りましょう。
4. ファイル `Now` の内容をファイル `NOW` に複写しましょう。 `NOW` の内容を確認しましょう。
5. ファイル `month` の内容を確認してから、`month` を `NOW` に複写してみましょう。 `NOW` の内容を確認しましょう。
6. ファイル `month` の名前を `Month` に変更しましょう。うまくいったかどうか、確認しましょう。
7. 今月のカレンダーが入ったファイル `.month` を作成してください。ファイル名の先頭にドットをつけ忘れないように。 `ls` コマンドを使って、作成したファイル名を表示してください。ファイルの内容も確認してください。
8. ファイル `NOW` と `.month` を削除してください。うまくいったか確認してください。
9. `-n` という名前のファイルがあったとして、このファイルの内容を `cat` で表示するにはどうしたらいいでしょう？<sup>14</sup>  
少なくともこの問いの正解がわかるまでは<sup>15</sup>、`-` で始まる名前のファイルを作らないでください<sup>16</sup>。

## 7.2 その2

1. ホームディレクトリに `eis17` という名前のディレクトリを作りましょう。ちゃんと作成できたことを確認しましょう。
2. 今作成したディレクトリ `eis17` をカレントディレクトリにしましょう。
3. カレントディレクトリにはまだファイルがないことを確認しましょう。
4. `echo` コマンドを使って、自分のユーザー名 (ログイン名) を画面に表示させてみましょう。
5. ディレクトリ `eis17` に、自分のユーザー名 (ログイン名) が書かれたファイル `loginname` を `echo` コマンドを使って作りましょう。
6. ファイル `loginname` の内容が正しいことを確認しましょう。間違っていたら、ファイルを作りなおしましょう。
7. `loginname` にローマ字表記の自分の氏名を追加しましょう。
8. `loginname` の内容を確認しましょう。
9. `loginname` のファイル名を `myUserName` に変更しましょう。
10. ホームディレクトリをカレントディレクトリにしましょう。

---

<sup>14</sup>通常のコマンドは `-` で始まる引数をオプションとして解釈します。したがって、`cat -n` だと、`-n` はファイルではなくオプションとみなされてしまいます。

<sup>15</sup>今は分からなくても構いません。

<sup>16</sup>`-n` という名前のファイルを作ってしまった人は `mv -- -n newname` でファイル名を変更してください。

## 8 コマンド一覧

*file ...* は複数のファイルを指定できることを, `[]` は中の要素が省略可能であることを表します。ここには次回紹介するコマンドも含んでいます。

分類	コマンド	機能
オンラインマニュアル	<code>man command</code>	<i>command</i> のマニュアルを表示
ファイル	<code>ls [file ...]</code>	カレントディレクトリに存在するファイルの名前を表示する。 <i>file ...</i> が指定された場合, その名前を表示する。 <i>file ...</i> がディレクトリである場合, その中のファイル名を表示する。
	<code>ls -F [file ...]</code>	ファイルの種類を表す記号を付加して表示
	<code>ls -a [file ...]</code>	ドットで始まるファイルを含めて表示
	<code>ls -l [file ...]</code>	詳細情報を表示
	<code>cp file1 file2</code>	<i>file1</i> を <i>file2</i> に複写
	<code>cp file ... directory</code>	<i>file ...</i> を <i>directory</i> に複写
	<code>mv file1 file2</code>	<i>file1</i> の名前を <i>file2</i> に変更
	<code>mv file ... directory</code>	<i>file ...</i> を <i>directory</i> に移動
	<code>rm file ...</code>	<i>file ...</i> を削除
	<code>cat [file ...]</code>	<i>file ...</i> の内容を (ファイルが複数の場合は連結して) 表示
ディレクトリ操作	<code>cd [directory]</code>	カレントディレクトリを <i>directory</i> に変更 ( <i>directory</i> を省略した場合はホームディレクトリに変更)
	<code>pwd</code>	カレントディレクトリ名を絶対パス名で表示
	<code>mkdir directory ...</code>	<i>directory ...</i> の作成
	<code>rmdir directory</code>	内容が空の <i>directory</i> を削除
	<code>mv directory1 directory2</code>	<i>directory1</i> の名前を <i>directory2</i> に変更
	<code>rm -r directory ...</code>	<i>directory ...</i> をディレクトリ階層を含めてすべて削除
その他	<code>cal</code>	カレンダーを表示する
	<code>clear</code>	端末スクリーンをクリアする
	<code>date</code>	システムの日付と時刻を表示・設定する
	<code>echo string ...</code>	<i>string ...</i> を画面に表示
	<code>exit</code>	終了