

UNIX のコマンドとファイル

2011 年 6 月 16 日

目 次

第 8 回 (6/16)

1	UNIX の実習について	1
1.1	どうして UNIX を学ぶの？	1
1.2	UNIX 機の利用開始と終了	1
1.3	困ったときには	1
1.4	参考書籍	2
2	コマンドの実行	2
2.1	コマンドの実行	2
2.1.1	練習	3
3	ファイル	3
3.1	ファイルを作ってみよう — ls, cat, >, >>	4
3.1.1	練習: ファイルへの内容追加など	4
3.2	ファイル名の付け方	5
3.2.1	練習	5
3.3	ファイル名の変更とファイルの複写・削除 — mv, cp, rm	5
3.3.1	注意	6
3.3.2	練習	6
3.4	コマンドのオプション	6
3.4.1	練習	6
4	コマンドのまとめ	7
5	問題	7
5.1	その 1	7
5.2	その 2	8

1 UNIX の実習について

1.1 どうして UNIX を学ぶの？

今回以降、この授業では UNIX (ユニックス) というオペレーティングシステム (OS) を利用して実習を行います¹。OS とは、コンピュータ全般を管理するソフトウェアのことです。OS の役割を簡単に言えば、コンピュータが持つ入出力などの諸機能を、容易に使えるように利用者やアプリケーション・ソフトウェアに提供することですが、当面は「Windows ではないものが動いているコンピュータを使う」とだけ考えていても差し支えありません。

さて、情報科学専攻の専門科目には、受講生が UNIX の基本的な使い方を知っていることを前提にした授業があります。卒業研究で必要となることもあります。この授業で UNIX を学ぶ大きな理由は、それらの準備のためです。加えて、今後の授業の目的には、UNIX の実習を通じて、コンピュータ一般に関わる知識を身に付けることも含まれています。すなわち、今後の授業で扱う内容の多くは、UNIX 以外の OS を用いる場合にも理解しておくべき基礎的事項も多く含まれます。

UNIX は大学等の学術機関や研究所などに普及している OS で、様々なシステムの開発や研究が UNIX を使って行われています。企業等でも、特に Web やデータベース等のサーバ用途に、UNIX が広く採用されています。従って、UNIX の利用法やその基礎的概念を学べば、それらは将来、直接あなたの役に立つかもしれません²。

1.2 UNIX 機の利用開始と終了

この授業では、Windows が動作しているコンピュータから、UNIX 機に遠隔ログイン (remote login) して、UNIX の実習を行います。

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。

本校の教育用端末から遠隔ログインする方法については、「UNIX 実習用コンピュータの利用手順」³ に記述がありますので、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。

1.3 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。

¹実際には UNIX の親戚である Linux を利用します。

²もちろん自分のコンピュータの OS として UNIX を使うことも可能です。

³<http://echoes.hak.hokkyodai.ac.jp/db/view/?id=20429>

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 キーを押してください。

1.4 参考書籍

- たのしい UNIX — UNIX への招待 —。坂本文。アスキー出版局
- 続・たのしい UNIX — シェルへの招待 —。坂本文。アスキー出版局
- UNIX プログラミング環境。B. W. Kernighan & R. Pike 著。アスキー出版局

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⁴。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⁵。
- 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>
```

⁴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。

⁵Windows 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 Hokkaido Univ. of Education` を実行してみましょう。

`echo` は与えられた引数を画面に表示するコマンドです。

3 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、その中身に加えて、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中身を見たり、中身を変更できます。

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。

3.1 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⁶。
- 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。

まだファイルを作っていないので、何も表示されません。

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。

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 の中身を見てみましょう。

3.1.1 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、> はファイルの中身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⁷。

4. cat dog を実行してみましょう。

⁶cat 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。

⁷例外はありますが、細かなことは無視しましょう。

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 1.3 節を読んで、正しい対処をしてください。

3.2 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⁸。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。

3.2.1 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。

3.3 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。

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 と同じ内容のファイル file2 を作る)

rm file — file を削除

⁸Windows 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。

3.3.1 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⁹。
- 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¹⁰。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¹¹。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。

3.3.2 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中身も見てください。
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。

3.4 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。

3.4.1 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。

⁹rm 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。

¹⁰cp 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。

¹¹設定によっては上書きされないこともあります。

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。

4 コマンドのまとめ

<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</code>	引数に与えたテキストを出力する
<code>exit</code>	おしまい
<code>man</code>	オンラインマニュアルページを整形し表示する
<code>ls</code>	ディレクトリの中身 (ファイル) をリスト表示する
<code>cat</code>	ファイルの内容を (連結して) 出力する
<code>mv</code>	ファイルを移動する (ファイル名を変更する)
<code>cp</code>	ファイルやディレクトリを複写 (コピー) する
<code>rm</code>	ファイルやディレクトリを削除する

5 問題

5.1 その1

UNIX のコマンドを使って設問の答えを調べましょう。または設問の操作を行いましょう。

1. 今日は何日でしょう。
2. 今月中に日曜日は何回あるでしょう。
3. 今年の元日は何曜日だったでしょう。
4. 自分の生まれた日は何曜日だったでしょう。
5. 自分の名前を計算機から出力 (表示) させてみましょう。
6. `man man` を実行したときには何が起きますか。結果を予想をしてから実行してみましょう。
7. `man woman` および `woman man` を実行したときの出力を比較して、なぜこのように異なった出力が得られるのかを考えましょう。

一般論として「エラーメッセージ」は注意深く読む習慣をつけましょう。

5.2 その2

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。
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。ファイルの中身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. `echo hakodate` を実行しましょう。結果を確認したら、中身が `hakodate` であるファイルを作ってください。ファイル名はお任せします。
10. `-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？¹²
少なくともこの問いの正解がわかるまでは¹³、`-` で始まる名前のファイルを作らないでください¹⁴。

¹²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。

¹³今は分からなくても構いません。

¹⁴`-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。