

# プログラミング 第1回レポート

202111609 仲村和士

2022年5月24日

## 1 はじめに

課題内容を始める前に重要なことを何点かこの節で述べる。

### 1.1 BibTeX のコンパイルについて (重要)

はじめに本レポートをコンパイルする際の注意点について述べたい。本レポートの文献情報には日本語書籍が含まれている。したがって通常の BibTeX ではなく pBibTeX を利用することを推奨する。

```
$ pbibtex -kanji=utf-8 file_name
```

### 1.2 L<sup>A</sup>T<sub>E</sub>X 利用の方針

随時インターネットで調べた結果を活用するが、日本語 L<sup>A</sup>T<sub>E</sub>X の代表的書籍である、奥村、黒木 [1] に記述があるものはこれに従う。講義資料と齟齬がある点についてはその都度記述する。

ただし、例外的に jarticle を利用する点のみ前述の書籍の記述から外れるものとする。L<sup>A</sup>T<sub>E</sub>X の作法としては非推奨であるが、今回のレポートのレギュレーションである以上それに従うものとする。

## 2 Linux のコマンド利用

### 2.1 課題内容と方針

本課題は講義ノートで扱ったコマンドを L<sup>A</sup>T<sub>E</sub>X の表形式でまとめる課題である。講義ノートでは Linux のシェル利用の基礎知識<sup>1</sup>なども多く書かれているが、本レ

---

<sup>1</sup>レポートを書く前に整理したところ、環境構築と導入、シェルの補完関係の機能、絶対パスと相対パス、テキストエディタ、パイプとリダイレクトなどが含まれる。これらも非常に大切であるが、レポートの設問意図からは外れると思われるため、あえて記述はしない。

ポートでは設問の意図より、それらは省略してファイル操作コマンドを中心にまとめることとする。

講義ノートで出てきているコマンドはあまり多くないとはいえ、ただ順番に列挙しただけではあまり見やすくないし、表にする意味が薄れてしまう。そこで、コマンドを4つの系統に分けて整理することにした。以下の4つである。

1. ファイルおよびディレクトリの情報確認とディレクトリの移動
2. ファイルおよびディレクトリへの操作
3. プログラムの中断と再開
4. その他

次の小節からはこれに沿って本文を構成する。また、表の出力には table 環境および tabular 環境を利用する。表の構成はコマンド名、主要なオプション、機能、実行例とする。これに関しては私が昨年書いた情報科学特別演習の最終レポート内で類似のコマンドリストを作っているため、こちらも一部参考にした。

## 2.2 情報確認とディレクトリの移動

ここからは、実際に UNIX コマンドを用いてファイルおよびディレクトリの情報を確認する方法、およびディレクトリの移動方法を解説する。

まず、ファイルおよびディレクトリの基本的な情報を調べるときには ls コマンドを用いる。ls コマンドは引数にパスを与えるとそのディレクトリに存在するファイル一覧を表示する。引数がないときにはカレントディレクトリの情報を表示する。与えられたパスがファイルのときはそのファイル自身の情報を表示する。よく使うオプションは -l と -a であり、-l オプションではより詳しい情報を表示する。-a ではデフォルトでは非表示の dotfiles を表示する。

ファイルの中身を確認するときには cat コマンドを使う。cat コマンドは引数に取ったファイルの内容を標準出力へ出力する。引数がない時には、標準入力を受け取る。しかし、ファイルの内容が多く、一画面に収まらないときもある。そのようなときは less コマンドを使う。less コマンドは与えられたファイルの中身を一画面ずつ出力する。移動には j,k で上下移動、q で終了といった vim ライクなキーバインドが用いられる。

ファイルの行数、単語数、バイト数を確認するときには wc コマンドを利用する。wc コマンドは引数にとったファイルの行数、単語数、バイト数を表示する。

カレントディレクトリを表示するときには pwd コマンドを利用する。余談であるが、私は初学の頃にパスワードを設定する passwd コマンドと混乱したので注意されたい。

カレントディレクトリを移動するときには `cd` コマンドを使う。`cd` コマンドを使うと引数に与えたパスに移動する。パスの指定方法は絶対パスでも相対パスでもよい。引数なしでの実行は普通 `cd ~` と等価であり、ホームディレクトリに移動する。また余談であるが、非常に便利な `cd -` というコマンドを紹介する。これを実行するとひとつ前にいたディレクトリに移動することができる。ひとつ前のディレクトリのみにはしか移動できないので完全に代用することはできないが、意図的にスタックを操作する必要がないので類似の機能を提供する `pushd/popd` より手軽に利用できる。

表 1 にここまでのコマンドをまとめた。

表 1: コマンドまとめ: 情報確認とディレクトリ移動

| コマンド名             | 主要オプション            | 機能                  | 実行例                         |
|-------------------|--------------------|---------------------|-----------------------------|
| <code>ls</code>   | <code>-l -a</code> | ディレクトリにあるファイル情報を表示  | <code>ls -la</code>         |
| <code>cat</code>  |                    | ファイルの中身を表示          | <code>cat scheme.sql</code> |
| <code>less</code> |                    | ファイルの中身を 1 画面ずつ表示   | <code>ps aux   less</code>  |
| <code>wc</code>   |                    | ファイルの行数、単語数、バイト数を表示 | <code>ls -l   wc</code>     |
| <code>pwd</code>  |                    | カレントディレクトリを表示       | <code>pwd</code>            |
| <code>cd</code>   |                    | カレントディレクトリを移動       | <code>cd ..</code>          |

## 2.3 ファイルおよびディレクトリへの操作

ファイルおよびディレクトリに対する操作を行う方法について説明する。

まず、空のファイルを作成するときには `touch` コマンドを用いる。`touch` コマンドは本来、引数にとったファイルのタイムスタンプを変更するためのコマンドであるが、引数にとったファイルが存在しない場合は新たに作成する。`-c` オプションを付けると、新規ファイル作成を行わない。

ファイルを削除するときには `rm` コマンドを利用する。`rm` コマンドは引数にとったファイルを削除する。一度削除すると復元できないので注意が必要である。`-r` オプションを付けるとディレクトリ内を再帰的に削除することができる。

ファイルをコピーするときは `cp` コマンドを利用する。`cp` コピー元 コピー先 のように記述する。`-r` オプションをつけるとディレクトリ内を再帰的にコピーする。`-i` オプションを付けると上書きに対する警告をしてくれる。

ファイルを移動するときには `mv` コマンドを利用する。書式は `cp` コマンドと同様である。`-i` オプションの挙動も同様である。`mv` コマンドは `cp` コマンドと異なり、削除の権限がないと利用できないのでシステム管理の際は注意を要する。

ディレクトリを作成するときには `mkdir` コマンドを利用する。`mkdir` ディレクトリ名 という形式で利用する。

ディレクトリを削除するときには `rmdir` または `rm -r` を利用する。`rmdir` は空のディレクトリのみ削除できるのに対して、`rm -r` は中にファイルが入っていても再帰的に削除できる。

表 2 にここまでのコマンドをまとめた。

表 2: コマンドまとめ: ファイルおよびディレクトリへの操作

| コマンド名              | 主要オプション            | 機能                | 実行例                       |
|--------------------|--------------------|-------------------|---------------------------|
| <code>touch</code> | <code>-c</code>    | タイムスタンプの変更、ファイル作成 | <code>touch a b c</code>  |
| <code>rm</code>    | <code>-r -f</code> | ファイルの削除           | <code>rm a</code>         |
| <code>cp</code>    | <code>-i -r</code> | ファイルのコピー          | <code>cp a ./sub/a</code> |
| <code>mv</code>    | <code>-i</code>    | ファイルの移動、リネーム      | <code>mv ./sub/a x</code> |
| <code>mkdir</code> |                    | ディレクトリの作成         | <code>mkdir /mnt</code>   |
| <code>rmdir</code> |                    | ディレクトリの削除         | <code>rmdir ./sub</code>  |

## 2.4 プログラムの中止と再開

前面にあるプログラムの中止と再開に関するコマンドについて説明する。

前面で実行中のジョブは、`C-z` により中断できる。中断されているジョブを表示するためには `jobs` コマンドを使う。`jobs` コマンドを入力すると、実行状態、ジョブ番号およびカレントジョブがわかる。

停止中のジョブについてフォアグラウンド/バックグラウンドで動かすためにはそれぞれ `fg/bg` コマンドを用いる。これらのコマンドはジョブ番号を引数にとってその番号のジョブを対象に動作する。`%` から始まるジョブ番号のみを入力することでもフォアグラウンドに復帰できる。

表 3 にここまでのコマンドをまとめた。

表 3: コマンドまとめ: プログラムの中止と再開

| コマンド名                | 主要オプション | 機能             | 実行例                     |
|----------------------|---------|----------------|-------------------------|
| <code>jobs</code>    |         | 中断している job を表示 | <code>jobs</code>       |
| <code>fg</code>      |         | 処理をフォアグラウンドに戻す | <code>fg</code>         |
| <code>bg</code>      |         | 処理をバックグラウンドで行う | <code>bg</code>         |
| <code>kill</code>    |         | プロセスを止める       | <code>kill 11870</code> |
| <code>%job 番号</code> |         | job 番号の処理に復帰   | <code>%2</code>         |

## 2.5 その他のコマンド

主に環境構築と導入で使用されたコマンドについて説明する。

Open SSH クライアントで ssh 接続する際には ssh コマンドを用いる。ssh user@hostname の形で使用するのが最も基本的であるが、ssh では公開鍵認証を利用する場合が大半であり、その際には -i オプションを付ける。ssh コマンドには多くのオプションがある。たとえば、-p オプションを付けるとデフォルトの 22 番以外のポートを指定できる。

カレンダーを表示する際には cal コマンドを利用する。引数で年・月を指定できる (最大西暦 9999 年まで)。引数を指定しない場合は現在の年のカレンダーを表示できる。

現在の時間を表示するには date コマンドを利用する。-u オプションを付けると UTC で表示する。そのほか、オプションを付けると何種類かのフォーマットで表示できる。

コマンドの使い方を調べるときには man コマンドを利用する。man コマンドは引数にとったコマンドのマニュアルを表示する。操作は less コマンドとほぼ同じである。

過去に実行したコマンドを再実行するときには history コマンドが便利である。history コマンドを実行すると過去に実行したコマンドが番号とともに一覧表示され、!番号 で再実行できる。

表 4 にここまでのコマンドをまとめた。

表 4: コマンドまとめ: その他

| コマンド名   | 主要オプション  | 機能                  | 実行例           |
|---------|----------|---------------------|---------------|
| ssh     | -i -p など | Open SSH で ssh 接続する | ssh user@host |
| cal     |          | カレンダー表示             | cal 9999      |
| date    | -u       | 日時の表示               | date -u       |
| man     |          | マニュアルの表示            | man man       |
| history |          | コマンド実行履歴の表示         | history       |

## 2.6 確認

本設問での指示内容は以下の通りである。

- 講義で扱ったコマンドを表としてまとめて本文で参照する
- 表の構成を説明する

ここまででこれらの条件はすべて満たしている。よって設問の題意を満たしていることが確認できた。

## 3 ソースコードの挿入

### 3.1 課題内容の方針

本課題は与えられた C 言語のソースコードを図として掲載するという内容である。もっとも簡単な方法は verbatim 環境を利用することであるが、ここでは listing パッケージを利用してよりきれいなリストの作成を試みる。listings パッケージの用法については情報科学類 [2, p.183] による。また、オプションのつけ方に関しては [3] も参照した。listing パッケージの利用方法を簡単に述べておくと、まず、プレアンプルで `\lstdefinestyle` を用いてスタイルを定義する。その際、複数のスタイルを定義できるが、共通の設定に関しては `\lstset` に記述するとコンパクトである。ここまでで listings パッケージを利用する準備が整った。listings の利用方法は次の 3 種類である。

1. L<sup>A</sup>T<sub>E</sub>X ソースに直接埋め込む
2. ファイルから読み込む
3. インラインで挿入する

今回はファイルが与えられているので 2 の方法を採用する。この方法では `\lstinputlisting` コマンドを利用する。

### 3.2 実際に挿入する

3.1 節で方針を説明したので、その通りにソースコードを挿入する。Listing 1 は C 言語によるソートの実装例である。

Listing 1: バブルソートの実装例

```
1 #include <stdio.h>
2
3 void sort(int n, int num[]){
4     int i, j;
5
6     for(i=0; i<n-1; i++){
7         for(j=i+1; j<n; j++){
8             if (num[i] > num[j]){
9                 int tmp;
10                tmp = num[i];
11                num[i] = num[j];
12                num[j] = tmp;
13            }
14        }
15    }
```

```

16 }
17
18 int main(int ac, char* av[]){
19     int num[] = { 3, 1, 4, 5, 9, 2, 6, 8, 7 };
20     int i;
21
22     for(i = 0; i < 9; i++){
23         printf("%3d", num[i]);
24     }
25     printf("\n");
26
27     sort(9, num);
28
29     for(i = 0; i < 9; i++){
30         printf("%3d", num[i]);
31     }
32     printf("\n");
33 }

```

### 3.3 確認

Listing 1 は行番号がつけられており、画像ではない図として挿入されている。よって題意を満たしている。

## 4 画像の挿入

### 4.1 課題内容と方針

本課題は全学計算機を操作しているコマンドラインをキャプチャして、画像を図として挿入するという内容である。私の環境は Windows であるから、画像のキャプチャには [Win]+[Shift]+s で開始される「切り取り&スケッチ」の機能を利用する。画像を挿入するには graphicx パッケージの \includegraphics コマンドを利用する。図として挿入するためには figure 環境を用いる。本レポートでは H オプションを利用するために here パッケージも読み込んでいる。

ここで、graphicx の注意点について述べておく。講義資料では PNG を埋め込むためには .xbb ファイルが必要と記されているが、それは旧情報である。現在ではこのようなファイルは不要であり、むしろ誤動作の原因となるので消しておくのが安全である。[1, p.126]

## 4.2 実際に挿入する

図1は全学計算機を使っている様子である。この図の中で行っていることを順番に説明する。

最初の`l`と書かれた部分ではエイリアスが利用できることを確認している。`.bashrc`にエイリアスが書いてあり、`ls`の代わりに`l`で動作するようになっている。次の`echo`では`alias`が便利であるということを標準出力を利用して主張している。次の`ls | wc`ではホームディレクトリを`ls`で調べた結果をパイプで`wc`に渡して行数、単語数、文字数を調べている。深い意味はない。次の`man man | cat | wc`はパイプを連鎖的に用いたワンライナーであり、`man`コマンドのマニュアルを`cat`でまとめて標準出力に流し、`wc`で大体のボリュームを確認している。次のコマンド`lua2latex`では、`lua2latex`が利用できることを確認している。この実行結果により、副次的に`TeX Live`のバージョンが2021であり、ほぼ最新の環境が利用できるということが分かった。次の`echo`ではその旨を主張しているが、リダイレクトでファイルに保存している。ファイル名は`Arch is the best`プロジェクト<sup>2</sup>のパロディである。最後の`echo`では、この画像が何をしているところなのかを示している。

```
s2111609@mastlinux1:~$ l
20170407.flg*  Contacts/  Documents/  Favorites/  Local-Repository/  Pictures/  Saved Games/
AppData/      Desktop/    Downloads/  Links/      Music/      Programming/  Searches/
s2111609@mastlinux1:~$ echo "alias is useful."
alias is useful.
s2111609@mastlinux1:~$ ls | wc
  20    22   187
s2111609@mastlinux1:~$ man man | cat | wc
<standard input>:232: warning [p 3, 1.2i]: cannot adjust line
<standard input>:232: warning [p 3, 1.3i]: cannot adjust line
<standard input>:237: warning [p 3, 2.5i]: cannot adjust line
<standard input>:239: warning [p 3, 2.8i]: cannot adjust line
<standard input>:691: warning [p 8, 3.3i]: cannot adjust line
  530   2210  41954
s2111609@mastlinux1:~$ man cal | cat | wc
  101    758   5232
s2111609@mastlinux1:~$ lua2latex
This is LuaHBTeX, Version 1.13.2 (TeX Live 2021)
restricted system commands enabled.
**^C
s2111609@mastlinux1:~$ echo "TeX Live 2021が使えるほぼ最新の環境です" > LuaLaTeX-is-the-best
s2111609@mastlinux1:~$ echo "全学計算機システムのLinuxを動かしているところ"
全学計算機システムのLinuxを動かしているところ
s2111609@mastlinux1:~$ |
```

図 1: 全学計算機を使っているところ

## 4.3 確認

本節の指示内容は次の通りであった。

<sup>2</sup>Arch Linux の優位性を示すための、少々思想が強いプロジェクト。ArchWiki に多くのプログラミング言語による実行方法が記載されている。



1. キャプチャ方法を含めて、課題内容を達成した方法を記述する
2. 全学計算機システムを利用しているところをキャプチャして図として挿入する
3. 図の内部で何をしているのかを説明する

1 は 4.1 節で記述した。2 と 3 は 4.2 節で記述した。よって題意を満たしている。

## 5 本の紹介

### 5.1 概略

本節では私の好きな本および、私が世話になった本を紹介したい。私の本棚の本は専門書/技術書が大半を占めているが、それでは面白くないため後半では意図的に技術書ではない本を 2 冊紹介することにした。前半の 2 冊が世話になった本で後半 2 冊が好きな本 (非技術書) である。

- L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 美文書作成入門 改訂第 8 版 [1]
- ホログラフィ入門: コンピュータを利用した 3 次元映像・3 次元計測 [4]
- 三日間の幸福 [5]
- 恋する小惑星 [6]

### 5.2 本の内容紹介

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 美文書作成入門 改訂第 8 版 [1] は言わずと知れた日本語 L<sup>A</sup>T<sub>E</sub>X の大ベストセラー書籍の最新版である。L<sup>A</sup>T<sub>E</sub>X を活用する上で必要なほぼすべてのトピックが網羅されており、日常的に L<sup>A</sup>T<sub>E</sub>X を利用する人は通読したうえで信頼性の高いリファレンスとして手元に置くべき一冊である。

ホログラフィ入門: コンピュータを利用した 3 次元映像・3 次元計測 [4] は私が 1 年次の ARE のときに大変世話になった本であり、ホログラフィの概念、および波動光学を利用した回折計算法などが実装とともに解説されている実用的な一冊である。

三日間の幸福 [5] は Web 発の小説であり、金欠学生の主人公が寿命を売れるという噂を聞き、査定を依頼するが、最低買取価格である 1 万円/年を言い渡され、生きる気力を失った主人公が残り 3 カ月を残してすべて売するというシーンから始まる。彼がどのような 3 カ月を過ごすのか ― ぜひ手にとってみてほしい。

恋する小惑星 [6] はまんがタイムきららキャラットに連載された漫画である。「小惑星を見つけたい」という夢を持った高校生の主人公を中心に地質、地図、気象

などさまざまな地学の分野に興味をもつ地学系女子たちによる物語である。つくばは本作品の聖地であるから、筑波大学の学生は必見である。

### 5.3 確認

本節での指示内容は以下の通りである。

- 課題フォームに記載された構成を満たす
- 本を itemize 環境で列挙し、文献を参照する
- 文献を参照しつつ、100 字程度でそれぞれの本の説明を書く

本節ではこれらをすべて満たしている。よって設問の題意を満たしていることが確認できた。

## 6 感想

### 6.1 Linux の基礎について

私は Linux の利用経験が 1 年以上あるため、ここまでの講義内容で新たに学んだことや得られたこと、苦労したことはほとんどなかったように思われる。しかし、オプションの類は忘れがちなのでレポートを書く際に改めてそれぞれのコマンドについて調べたことによりいくつか記憶に定着させることができた。また、表としてまとめる際にコマンドのオプションの共通点などが整理できたことは有益だった。(例: `-r` は再帰的という意味、`-i` は上書きを防止する、など)

### 6.2 L<sup>A</sup>T<sub>E</sub>X について

私は L<sup>A</sup>T<sub>E</sub>X については昨年少し使ったくらいであり、ローカルで環境構築をして本格的に使い始めたのは今年になってからである。その際にいろいろなことを調べたので講義資料の記述に引っかかりを覚える点が多くあった。正しい知識を得るために [1] を読み込んだ経験により L<sup>A</sup>T<sub>E</sub>X の知識が大いに向上したことは非常に有意義であった。特に `amsmath` パッケージ周りの理解が向上したように感じられる。

## 6.3 レポートについて

前節までで記述すべき内容はすべて終了している。ここからはおまけである。

レポートを書くうえで難しかったのは課題内容に対する取り組み方針の記述である。解答を得るための取り組みの手順は書こうと思えばいくらでも細かく書くことができる。今回の出題では求められる高いクオリティの割にテンプレート、例が示されていないのでどれくらいの粒度で書くのが良いとされるのかは全く不明である点が難しかった。

レポートの採点基準についても気になった。課題指示に具体例が示されていないので非常に曖昧な部分が多いように思われる。

上記の注意事項に抵触する場合は、成績評価の範囲外となる。

(4) その他の不備は減点の対象となることがある。

という記述からは教員の一存でどんなレポートでも 0 点にすることができる権限があると読み取れ、恣意的な運用が可能である。

私個人としては、他人の不備を指摘して減点すると公言するためには、まず自身の資料が完璧であるべきだと考える。時代に即していない内容が堂々と初学者に教えられる現在の状況は大変遺憾である。

## 参考文献

- [1] 奥村晴彦, 黒木祐介. L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 美文書作成入門 改訂第 8 版. 技術評論社, 2020.
- [2] 筑波大学 情報学群情報科学類. 教育用計算機システム使用の手引き 2022 年度版, 2022. <https://www.coins.tsukuba.ac.jp/tebiki/2022/tebiki2022.pdf>.
- [3] TomKid. Latex でコードを載せるときとか目次の作り方とか. 図を figure や表を table に変更する方法も, 2018. <https://qiita.com/TomKid/items/5da182037c2253dfb28d>.
- [4] 伊藤智義, 下馬場朋禄. ホログラフィ入門: コンピュータを利用した 3 次元映像・3 次元計測. 講談社, 2017.
- [5] 三秋縊. 三日間の幸福. メディアワークス文庫, 2013.
- [6] Quro. 恋する小惑星 (アステロイド)(1). 株式会社芳文社, 2018.