

計算機概論 I 第 2 回

Editor(Gedit) 入門, Gnome 端末からコマンドを使う

Apple Silicon(Apple M1 など) でこの授業の内容を実行する方法:

<http://www.cc.u-ryukyu.ac.jp/~b977046/gairon/2022/mac.pdf>

この資料: <http://www.cc.u-ryukyu.ac.jp/~b977046/gairon/2022/02.pdf>

目次

1	本日の目標	1
2	用語の説明	1
3	実習 1: Gedit を使う	3
3.1	Gedit を起動する	3
3.2	ファイルの保存	4
4	実習 2: Gnome 端末を使った簡単なコマンドの実行	5

1 本日の目標

- エディタという概念を知り, gedit が使えるようになる.
- コマンドを使ってコンピュータを使う

2 用語の説明

テキストエディタ (text editor) 文字情報のみのファイル (テキストファイル) を作成, 編集, 保存するためのソフトウェア. 文字の入力, 削除, コピー, 貼り付け, 検索, 置換, 整形等の機能を備えている. CentOS では, 今回使う Gedit 以外に vi(vim), Emacs 等がある.

例えば, 小説家がコンピュータに小説の内容を入力するとします. そのとき必要なのは, 小説の文章を効率良く入力するための機能で, それを見栄え良く配置する必要はありません. 文章の配置を決めるのは出版社や印刷屋の仕事です. エディタは, この「文を効率良く入力する機能」に特化したソフトウェアです.

数理学科の皆さんのエディタの使い方は, 「ソースコードエディタ」としてです. コンピュータ内のデータは全て 2 進法の数です. コンピュータに対する命令も 2 進法の数です. しかし, 人間には 2 進法の数は不便な

ので、コンピュータに対する指示を人間が理解可能な文書で記述し、それをコンピュータが翻訳して実行するという仕組みが開発されました。実際皆さんが利用するプログラム (OS, アプリケーション) は、そうして作られています。コンピュータに対する指示を記述した人間が理解可能な文書を、ソースコード (source code, 略してソース) と言います。今後、大学のコンピュータ関連の授業では、ソースコードを入力する事が多くあります。

コンピュータは機械ですから、曖昧な指示は通用しません。ソースコードには、コンピュータが理解できる (人間も理解できる) 曖昧性の無い人工的な言語が書かれます。ソースコードエディタでは、この人工的な言語の入力に対する補助機能 (キーワードのハイライトや括弧の対応関係の表示) が備わっています。

下に挙げる、gedit, vi(vim), Emacs はソースコードエディタとして十分な機能を持っています。これらは、MacOS, Windows, Linux 全てで動作します。情報処理センターでは、Mac には vi が、Linux には全てが導入されており、Windows には全てが入っていません。それぞれの特徴は、次のようになります。

vi(vim) Bill Joy(Sun Microsystems 社の創立者。Sun Microsystems の開発した OS, Solaris は、長年情報処理センターのサーバの OS であった。) が、California 大学バークレー (Berkeley) 校で学生時代、1976 年ころ開発したエディタ。vi は Visual editor の最初に 2 文字。vim は Vi improved の略。Unix 系の OS では必ず導入されている。モードの概念があるため初心者にはとりつきづらいが、作業の全てがキーボードでできる (というかそもそもマウスが使えない)、なれると非常に高速な作業ができる、最小限のシステムでも使えるなどの理由で、現在も利用されつづけている。ちなみに、<http://www.cc.u-ryukyu.ac.jp/~b977046/gairon/2021/00.pdf> で紹介した macOS, WSL, Linux(Mathlibre) の全ての環境で vi が導入されています。私自身、例えば情報科学演習の教材の Web ページの編集 (微修正) には、Mac で vi を利用しています。Emacs は、現時点では、改めて導入したいと利用できないエディタになっています。

Emacs Richard Stallman が 1975 年ころから開発を始めたエディタ。Editor Macros を略して Emacs。多くの Unix 系の OS に導入されている。名前のおり、編集機能の集合体で、vi よりはかなり大きなプログラムである。高度な機能を備えるように、拡張が容易なプログラムとなっている。vi と同様作業の全てがキーボードだけでできる。きちんとした設定ファイルや拡張機能があれば、gedit より使いやすい。残念ながら、現在の情報処理センターには、その拡張機能が導入されていない (今回のシステムの変更で、少しはマシになった)。以前のシステムには入っていたので、それまでは Emacs を利用していた。また、macOS のキーボード機能の中には、emacs 由来のものが多くあるので、emacs の操作体系を知っていると、Mac が便利に使えるようになる。

gedit 多くの Linux ディストリビューションに導入されている。Windows のメモ帳に似ているが、ソースコードエディタとしての機能が備わっている (Windows のメモ帳にはその機能はない)。残念な点は、日本語入力で全角の記号や英数が入力できる事で、これはソースコードの間違いを探す上で大変になる (特に全角の空白は目に見えないが、ソースコードとしてエラーになる) ところ。

時間の都合で、vi や Emacs の使い方の講義はしませんが、勉強をして損はしませんので、興味がある人は、前回の講義で紹介した、

はじめての Linux これだけは知っておきたい、小林真也監修、宇戸寿幸他著、森北出版、

などで勉強して下さい。これら以外にもテキストエディタはたくさん開発されています。単純にテキストを入力するだけでなく、文字列の検索や置換の方法を知る事は、コンピュータを使いこなす上で重要な技能です。

3 実習 1: Gedit を使う

3.1 Gedit を起動する

この講義では、エディタとして Gedit を使います。メニューバーの「アクティビティ」をクリックして、一番下のアプリケーションを表示するの中を探せば、「テキストエディター」というのが入っていると思いますが、それが Gedit です。それを選んでください。画面の一番大きな白色の部分に入力した文字列 (テキスト) が表示されます。その部分の左上端に縦棒が点滅していますが、それを「カーソル」といいます。

Gedit のメニューは、ウィンドウ上部の保存アイコンの右側にある四角いアイコンです。

かな漢字変換について

日本語入力は、キーボードの全角/半角キーで切り替えます。切り替わらない場合は、メニューバー右上の方に A という文字があると思いますので、それをクリックします。その中に、かな漢字という項目があると思いますので、それを選択します。

日本語環境でソースコードを入力する際、最も厄介な問題が全角のアルファベット、数字、記号です。

情報科学演習で HTML を利用しましたが、HTML のタグは全て半角文字でした。プログラミング言語や L^AT_EX 等でも、処理系に対する指示は全て半角文字で、全角文字は使われません。特に厄介なのが全角の空白で、目に見えないため、プログラムソースの中のこの文字によるエラーの発見は、難しくなります。

これ以外にも、日本語の半角カナの問題が起こることもあります。半角カナという、昔に定義された文字集合がありますが、現在では全く意味を持ちませんし、エラーの原因になることもあります。日本語の文字を入力する場合は、常に全角文字を入力する様にしてください。

文字の入力の際には、どの様な環境 (スマートフォンなども含む) であっても、ここに述べたことを実行する様に心がけてください。

Gedit の練習も兼ねて、次のページの英文を入力してください。バックslash \ は、英語入力モード (半角モード) で、円記号 ¥ のキーを押します。(何故そうなるかは、情報科学演習の講義で述べましたし、後の講義でも一度解説します。気になる人は調べてください。)

```

\documentclass{article}
\begin{document}

A ring  $A$  is a set with two binary operations
(addition and multiplication) such that

\begin{enumerate}
\item  $A$  is an abelian group with respect to addition.
\item Multiplication is associative
and distributive over addition.
\item There exists an identity  $1$  such that  $x1 = 1x = x$  for all  $x \in A$ .
\end{enumerate}

\end{document}

```

3.2 ファイルの保存

今の状態では、入力した内容はコンピュータのメモリ (1 次記憶装置) に存在しており、コンピュータの電源を切ればその内容は消えてしまいます。入力した内容が消えないようにするには、ファイルという形で保存して、2 次記憶装置 (ハードディスク) に記録しなければなりません。そのためには、次の操作をします。

1. ウィンドウ上部の「保存」アイコンをクリックする。
2. 保存のダイアログが表示されるので、「名前」の部分で「ring.txt」と入力する。
3. 左の欄は、「ホーム」が選ばれていることを確認する。選ばれていなければ「ホーム」を選ぶ。
4. 保存ボタンを押す (あるいはエンターキーを押す)。

この操作をすると、ホームディレクトリ「xxx のホーム」に ring.txt という名前のアイコンができるので、ファイルブラウザ (アクティビティボタン (左上) を押して出てくるキャビネットアイコン) を起動して見てください。

同じ内容を別のファイル名で保存することも出来ます。上の内容を、「ring」というファイル名で保存するには、

1. 保存ボタンの右の ≡ の記号があるボタンを押す (メニューボタン)。
2. 名前をつけて保存を選ぶ。
3. 保存のダイアログが表示されるので、「名前」の部分で「ring」と入力して「保存」ボタンを押す (あるいはエンターキーを押す)。

実習: 今のファイルの下から 2 行目に次を加えて ring.tex という名前で「別名で保存」して下さい。このとき画面が変化する事も注意して下さい。

```
The identity element is then unique.
```

上の操作では、ring.tex というのが $\text{T}_{\text{E}}\text{X}$ のソースコードを示す拡張子 .tex がついているので、Gedit が $\text{T}_{\text{E}}\text{X}$ モードに変わりました。最初からモードを指定するには、メニューから

表示 → ハイライトモード

で LaTeX を選ぶと、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の強調表示モードが設定できます。

4 実習 2: Gnome 端末を使った簡単なコマンドの実行

この講義では、端末エミュレータを使ってシステムを使う事も重視します。アクティビティを押すと、左側に黒い画面を模したアイコンがあると思うので、それをクリックします。あるいは、アクティビティから左側の一番下のアイコンを選んで、アプリケーション一覧から「端末」「ターミナル」と書かれたアプリケーションを起動します。これは Gnome ターミナルと呼ばれるソフトウェアです。

端末 (ターミナル) 狭義には、TELEX 端末のように、文字をコンピュータ (やシステム) に入力するためのキーボードと、コンピュータ (やシステム) からの出力を印字するための印刷機構からなる装置のことを指します。一般的には、(主に) 文字をコンピュータに入力したり、逆にコンピュータからの出力を表示したりする装置、もしくはそのような物理的、もしくはソフトウェア的に作られた仮想的な機構のこと。

端末エミュレータ 端末の機構を実現するソフトウェア。

端末エミュレータのウィンドウの左上隅に表示されている `Bash-4.4$` の文字列を、コマンドプロンプト (日本語では意識してコマンド入力促進子) といいます。コマンドプロンプトは設定によって変える事もできます。その右にカーソルが点滅していると思います。適当な文字を入力して ENTER キーを押します。多くの場合、

```
Bash-4.4$ asdfg
```

```
asdfg: コマンドが見つかりません.
```

の様にコマンドが見つかりませんと言うメッセージが出ます。

CLI では、ユーザはやりたい事はどのコマンドで実現できるかを知らなければなりません。またどのようなコマンドが実行できるかも知る必要があります (後の授業でやります)。最悪の場合、必要なコマンドは存在しません。このときは、プログラミングをして作ります (計算機言語 I・II で講義する)。

GUI 環境では多くの操作がマウスでできますが、Linux の場合、その操作に対応するコマンドをユーザにわからないように実行している事が多くあります。それを知るのが、この講義の 1 つの目的です。

コマンドプロンプトのある行がコマンド入力行で、コマンド名を入力して最後に ENTER キーを押すことで、コマンドが実行されます。ウィンドウには、コマンド実行の結果とエラーメッセージが表示されます。

「マウスと Window システムを使って操作できる内容を何故 CLI でやるのか?」という根源的な質問に対しては、「時には CLI の方が便利な事もある。」というのが答です。余り経験が無いとは思いますが、ネットワークサービスの設定等は、設定ファイルの作成と編集が主な作業になりますが、この様な作業を多量に行うとなると、GUI より CLI の方が迅速な処理ができるのです。

以下では、`Bash-4.4$` で始まる行はコマンドの入力行で、これに続く文字列は、コマンドの実行の例です。

注意 キーボードをタイプしても何も起こらなければ、端末エミュレータがアクティブかを確認して下さい。

date 現在の日時を知るコマンドです。次のようにタイプしてください。今の日時がわかります。

```
Bash-4.4$ date
```

JST は日本標準時です。現在では、NTP (network time protocol) を利用して、インターネットに繋いだコンピュータは、正確な時刻設定ができる様になっています。

clear 画面をクリアするコマンドです。次のようにタイプしてみてください。

```
Bash-4.4$ clear
```

cal **calendar** を表示します。

```
Bash-4.4$ cal
```

とタイプしてみてください。今月のカレンダーが表示されます。来月のカレンダーを表示するには

```
Bash-4.4$ cal 11 2022
```

とタイプします。11 2022 は 2022 年 11 月の意味です。(cal と 11 と 2022 の間には、空白を入れる。)

```
Bash-4.4$ cal 2023
```

とタイプすると、来年のカレンダーが全部でてきます。

```
Bash-4.4$ cal 9 1752
```

も試して下さい。英国のカレンダーを採用していると思いますが、私は歴史は詳しくありません。

who, w 今あなたが使っているワークステーションを、同時に使っている人全員を表示します。

```
Bash-4.4$ who
```

とタイプしてみてください。

```
Bash-4.4$ w
```

とタイプすると、少し違った表示が出ます。次をタイプとすると、あなたの login 名が出力されます。

```
Bash-4.4$ whoami
```

whoami というコマンドは「頭がおかしいのではないかと感じるとは思いますが、管理者権限を持つ事が出来る人には重要です。管理者になれる人でも、コンピュータを操作する際には、通常は管理者以外のアカウントを使います。管理者には、システム全体を破壊してしまう権限があるからです。システム管理をする人は、それを実行するときだけ一時的に管理者になります。そのときに、自分は一般ユーザなのかそれとも管理者なのかがわからなくなることがあり、それを確かめるための「whoami」コマンドなのです(経験者は語る)。

du **disk usage**. ハードディスクの使用状況をレポートします。センターは多くの学生が利用するため、個人が使えるハードディスク容量に、制限を付けてます。これを越えますと、授業に使用できなくなりますので注意して下さい。センターでは個人用のデータは一括して管理しており、Windows 等で使ったファイルも合計した容量です。画像や音楽を保存しだすと、簡単に制限を越えましますので注意して下さい。

```
Bash-4.4$ du
```

手っ取り早く、ホームディレクトリにあるファイルの大きさ全体を知るには、次の様にタイプします。

```
Bash-4.4$ du -sk .
```

(du スペース -sk スペース ピリオドです。) こうタイプすると、結果はキロバイト単位です。「-sk」の部分は、du というコマンドに対するオプションと呼ばれるもので、コマンドの挙動を変更する際に用います。どのようなオプションがありどのように挙動は変更できるかは、後で述べる man コマンドで知る事ができます。

情報処理センター全体での使用容量の制限が 1.5GB ですが、そのためには前回見た全体のホームの使用量を知る必要があります。全体のホームの使用量を知るコマンドは、

```
Bash-4.4$ du -sk ..
```

(du スペース -sk スペース ピリオド 2つの連続) です。ピリオドの意味は後の講義で解説します。

df マウントしているファイルシステムの状況をレポートします。ファイルシステムについては、後の講義でもう一度触れます。

```
Bash-4.4$ df
```

bc 電卓です。実はプログラマブルですがここではそれについては触れません。

```
Bash-4.4$ bc
```

とすると、計算式の入力を待つモードに入ります。例えば、

```
1234567890+987654321
```

とタイプすると、

```
2222222211
```

と答が返ってきます。終了するには、C-d(Control キーを押しながら d を押す。以下この省略記号を使います。) か quit(をタイプして ENTER) です。

uname 使っているシステムの名前を表示します。

```
Bash-4.4$ uname
```

Unix では、OS に対してそれをインプリメントした企業 (あるいは、非営利団体) がつけた名前がついています。より詳しい情報は次で得られます。

```
Bash-4.4$ uname -a
```

hostname 使っているコンピュータの名前を表示する。Unix では、コンピュータに名前がついています。

```
Bash-4.4$ hostname
```

man オンラインマニュアルです。例えば clear の説明を見なければ、次のように入力します。

```
Bash-4.4$ man clear
```

次で man コマンドの説明がえられます。

```
Bash-4.4$ man man
```

: (左端がコロン) という表示で止まったら、スペースを押せば、次のページが見られます。b キーで 1 ページ戻ります。表示を終了してコマンドプロンプトに戻るには、q キーを押します。ほとんどのコマンドにオンラインマニュアルがありますので、コマンドの内容を詳しく知りたいとき、man コマンドでそれを確かめることが Unix を使う上では重要です。どのようなコマンドが存在しているかを調べる方法は、後の講義で触れます。man コマンドにキーワード検索機能もついています。例えば、unix に関係するコマンドにどのようなものがあるかを知るには、(スペースが入る事に注意)

```
Bash-4.4$ man -k unix
```

とすると、それにマッチするコマンドの一覧を表示してくれます。実際に実行してみてください。

ls ファイルとディレクトリのリストを表示します。ファイルという言葉は、情報科学演習でも習いました。ディレクトリは、フォルダと同じ概念です。これらについては、後の講義もう一度触れます。

```
Bash-4.4$ ls
```

とタイプしてみてください。本来のホームに WWW という名前のディレクトリがあり、これが情報科学演習で作成した Web ページのためのフォルダです。(/WWW の前にピリオド 2 つ)。

私の情報科学演習の単位を取得した人は、次の実行結果をみて下さい。

```
Bash-4.4$ ls ../WWW
```

echo 入力文字列をそのまま繰り返す (こだまさせる) プログラムです。

```
Bash-4.4$ echo Unix
```

としてみてください。これでは面白くありませんが、

```
Bash-4.4$ echo *
```

```
Bash-4.4$ echo ?????
```

とかは、少し違った結果になります。なぜそうなるかは、後の講義で取り上げます。

ps 現在動かしているプロセス (実際に動いているプログラム) を表示するコマンドです。

```
Bash-4.4$ ps
```

を実行してみてください。プロセスについても後の講義でもう一度触れます。

top 動いているプロセスを、仕事量の多い順に表示する。

```
Bash-4.4$ top
```

そこに現れるデータの意味は、あとの講義で触れます。Xorg というプロセスが上位に来ていると思いますが、それが Window System です。終了は q キーです。

file ファイル型を表示するコマンド。先程作ったファイル ring について実行してみてください。

```
Bash-4.4$ file ring.txt
```

他のファイル型を見てください。先程作ったファイルでも、ring.tex だと違う結果になります。

```
Bash-4.4$ file ring.tex
```

もっと別のファイル、例えば/bin/ls というファイルでは、どうでしょうか (/bin/ls の意味は、後の講義で解説します)。

```
Bash-4.4$ file /bin/ls
```

次も試してください (/WWW の前にはピリオドが 2 つ)。

```
Bash-4.4$ file ../WWW/index.html
```

cat 本来は、ファイル連結の (Concatenate) コマンドですが、cat ファイル名でファイルの内容を表示する。

```
Bash-4.4$ cat ring
```

cat の引数を 2 つのファイル名にするとどうなるでしょう。

```
Bash-4.4$ cat ring ring.tex
```

ファイル ring に続いて ring の内容が出力されます。すなわち、2 つのファイルを連結して出力しているのです。3 つ以上のファイルを引数に与えても同様の結果になります。

ファイルの中身が長いと、延々と出力をし続けます。その出力を一時的に止めるには、標準的な環境では、C-s(Control を押しながら s) です。適当なキーを押す事により、止まった出力が再開します。また、cat そのものの動作を中止するには、C-c(Control を押しながら c) です。

ファイル名を指定しないとどうなるでしょう。

```
Bash-4.4$ cat
```

(カーソル)

aiueo と打って ENTER を押してみます。

aiueo ← これを入力

```
aiueo
```

となります。理由は後の講義で解説します。とりあえず C-d と打ちますと、この状態が終了します。

注意 むやみに知らないファイル (というかバイナリファイル) を cat してはいけません。暇なときに

```
Bash-4.4$ cat /usr/bin/vim
```


とでもするとわかります。放っておくと長時間かかるので、C-c で止めます。止まった後、運が良ければ何事もなかったように元に戻りますが、大抵は画面が意味不明な文字の嵐になります。壊れたわけではないので、そのウィンドウは終了して下さい。新しく Gnome 端末を開いて下さい。

```
Bash-4.4$ man cat
```

でわかるはずです。

more, less ファイルの内容を 1 ページ毎に出力する。

```
Bash-4.4$ more ring
```

```
Bash-4.4$ less ring
```

のように使います。終了は q キーです。今のところ、長い内容のファイルが無いので、つまらない結果しかできません。more は標準の Unix のコマンドで、less は more の機能強化版として開発されました。

od ファイルを 8 進表示するコマンドです。次を試して下さい。

```
Bash-4.4$ od ring
```

オプション x をつけると 16 進数表示をします。また、オプション c をつけると保存されているデータのうちの、0 - 127 に対応する数字を対応する文字として表示します。次を試して下さい。

```
Bash-4.4$ od -x ring
```

```
Bash-4.4$ od -c ring
```

wc デフォルトでは、ファイルの中の行数、単語数、ファイルの大きさを順に出力します。(word count)

```
Bash-4.4$ wc ring
```

head ファイルの先頭部分を表示するコマンド。

```
Bash-4.4$ head ring.tex
```

でファイル ring.tex の先頭 10 行が出力されます。出力する行数の指定の仕方は、各自調べて下さい。

tail ファイルの末尾部分を表示するコマンド。

```
Bash-4.4$ tail ring.tex
```

でファイル ring.tex の末尾 10 行が出力されます。出力する行数の指定の仕方は、head と同じです。

diff 2つのファイルの相違部分を表示する。違いが無ければ、何も出力しません。

```
Bash-4.4$ diff ring ring.tex
```

今見ている画面に、今の時刻が表示する事を考えます。システムから時刻を取得する方法は何通りかありますが、上で述べた date コマンドを利用するのが、手っ取り早い方法です。例えば、メニューバー右端に date コマンドを実行した結果を 1 秒ごとに表示するように Window システムを作れば、時刻を表示する Window システムが出来上がります。「それぞれは機能の少ない簡単なプログラムを寄せ集めて、1つのシステムを構成する。」というのがあります。多くのコマンドは、このような「機能は少ないが簡単に作れる小さなプログラム」なのです。多機能な大きなプログラムも、このような小さな部品を集めて作られています。

補足 (この講義のテキスト)

この講義は、概ね昨年度の講義内容になる予定です。下の場所が、昨年の講義の PDF の置き場です (遠隔授業に対する補足は、参考にしないでください。)

<http://www.cc.u-ryukyu.ac.jp/~b977046/gairon/2021/>

今年度のテキストの L^AT_EX ファイルを, 講義終了後間違いを修正して, 順次置く予定です.

<ftp://ftp.math.u-ryukyu.ac.jp/pub/gairon/2022/>

PDF ファイルは次の場所に置く予定です.

<http://www.cc.u-ryukyu.ac.jp/~b977046/gairon/2022>

質問, その他は次が教員のメールアドレスですので, こちらで受け付けます.

suga@math.u-ryukyu.ac.jp

資料 (数学の専門書も含む) の読み方

私が配布する資料は (数学の専門書も), 無駄や冗長性をできるだけ排除してあります. 「キーワードを拾って飛ばし読みする」という行為は, 大学では (少なくとも数理科学科では), 物事の理解が進まず, 時間を無駄にする行為です. おそらく, 受験勉強でこのような勉強法を身につけたのではないかと思いますが, そろそろ, それから脱却してください. 配布資料は, 全てをきちんと読んでください. 人間は, 無意識に飛ばし読みをしたり, 思い込みをしたりするということも, 知っておいてください.

システムをスリープしなくなるように設定する (再掲)

授業中にシステムをスリープさせる必要はないので, 次の設定をして下さい.

アクティビティ → アプリケーションメニュー → 設定

と選んで, 左の蘭の Power を選択し, ブリンクスクリーンを「なし」にする.

次回からは, 授業が始まる前に CentOS に login しておいて下さい.