

計算物理学II

第1回

オリエンテーション・計算機システム（Linux）の使い方

秋山 進一郎

2025年10月3日

授業の目的

- ・ 計算機を活用していくための基礎的事項を習得する
 - ・ 計算機の操作 (Linux, Terminal) に慣れる
 - ・ LaTeXによる文書作成
 - ・ 数値計算のためのプログラミング技術の基礎
 - ・ 後続く「計算物理学III, IV」において必須となる技術の習得
 - ・ その他の授業 (座学, 実験を問わない)
 - ・ 卒業研究
 - ・ 大学卒業後
 - ・ ...

使用する計算機について

- ・全学計算機システム（推奨）
 - ・全学計算機についてのwebサイトは[ここ](#)
 - ・この授業を進めるために必要な環境が全て揃っているのでお勧めします
 - ・授業時間外でも利用可能
 - ・サテライトの開室状況は[ここ](#)
 - ・リモートアクセス
- ・自前のパソコン
 - ・自分で環境構築ができるのであれば何を使ってもOK
 - ・クラウドサービスを使う（Overleaf, Google Colab, …）

授業の進め方

- ・ 各授業で最初に簡単な説明をし、その後各自で演習を進める
- ・ 講義, 演習資料はGitHubから入手
 - ・ <https://github.com/akiyama-es>
- ・ 演習中に分からないことがあれば積極的に質問を
 - ・ スタッフ, TA（高橋さん）, 受講生同士での相談, 議論
- ・ 授業時間外の連絡、質問
 - ・ E-mail : akiyama@ccs.tsukuba.ac.jp
 - ・ Teams : チームコードをmanabaで通知

授業日の確認

- ・ 全10回
 - ・ 第1回：10月3日（金）
 - ・ 第2回：10月10日（金）
 - ・ 第3回：10月17日（金）
 - ・ 第4回：10月24日（金）★
 - ・ 第5回：10月31日（金）
 - ・ 第6回：11月14日（金）★
 - ・ 第7回：11月21日（金）
 - ・ 第8回：12月5日（金）★
 - ・ 第9回：12月12日（金）
 - ・ 第10回：12月19日（金）★
- ・ ★の付いた授業にてレポート課題を配布予定

成績評価

- ・ レポート課題（全4回）
 - ・ レポート課題の提出×切（予定）
 - ・ 第1回：11月7日（金）
 - ・ 第2回：11月28日（金）
 - ・ 第3回：12月6日（金）
 - ・ 第4回：1月9日（金）
 - ・ 各レポートをA+, A, B, C, Dの5段階で評価し、各レポート評価を平均して最終的な成績を決定します
 - ・ レポートの提出はmanabaからを基本とします

生成AIの利用について

- ・ 生成AIの長所及び短所, 使用上注意すべき点を踏まえた上で, 生成AIの活用を認める
 - ・ 推奨される使用例 **生成AIの答えを必ず自身で検証する (Critical thinking)**
 - ・ 「〇〇するためのコマンドは何だっけ？」
 - ・ 「書いたコードが動かない！」
 - ・ 「〇〇するためのライブラリは？」
 - ・ 「実行時間をもっと短くできないか？」
 - ・ ダメな例 **生成AIの答えを盲信する**
 - ・ レポート課題を全てAIに解かせ, 何の検証もなくそのまま提出

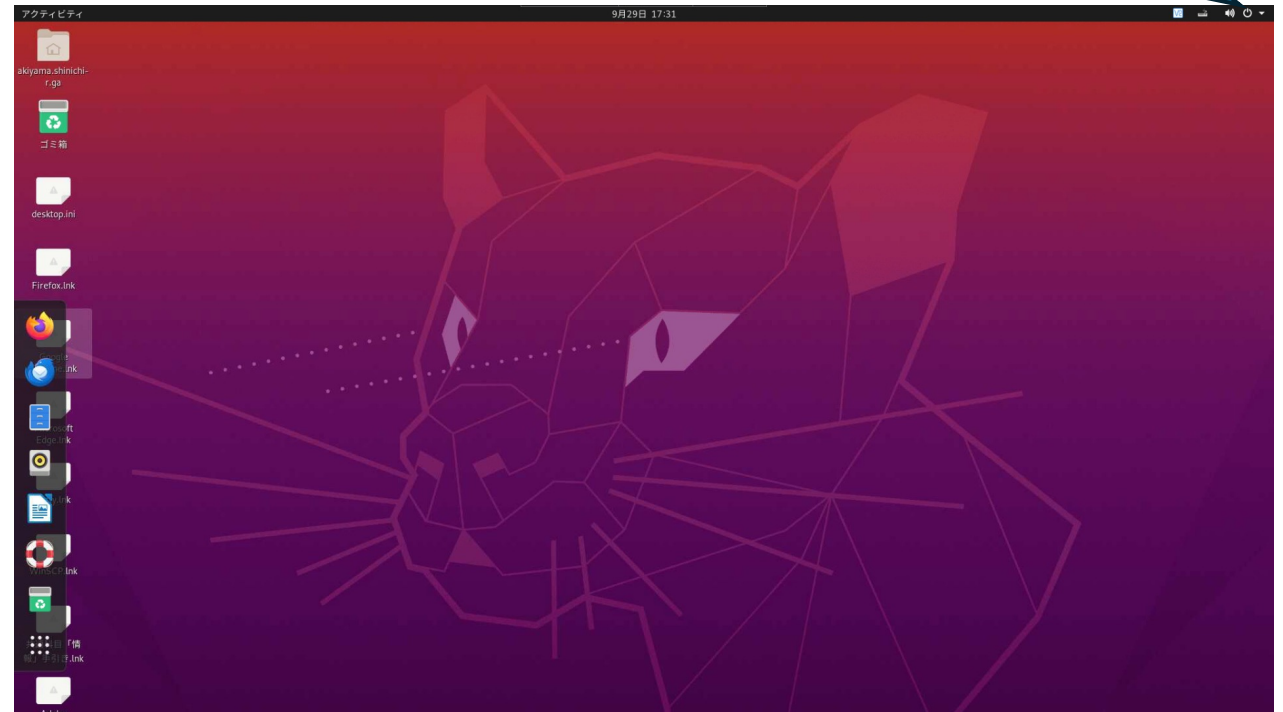
本日の演習内容

- ・ 全学計算機システムを使ってみる
 - ・ Linuxに入る
 - ・ 端末（Terminal）を使ってみる
 - ・ 演習資料の入手
 - ・ いくつかの基本コマンドを使ってみる（ここから各自で演習）
 - ・ いくつかのエディタを使ってみる
- ・ 全学計算機システムへのリモートアクセス（必要に応じて授業後に各自で）

Linuxに入る

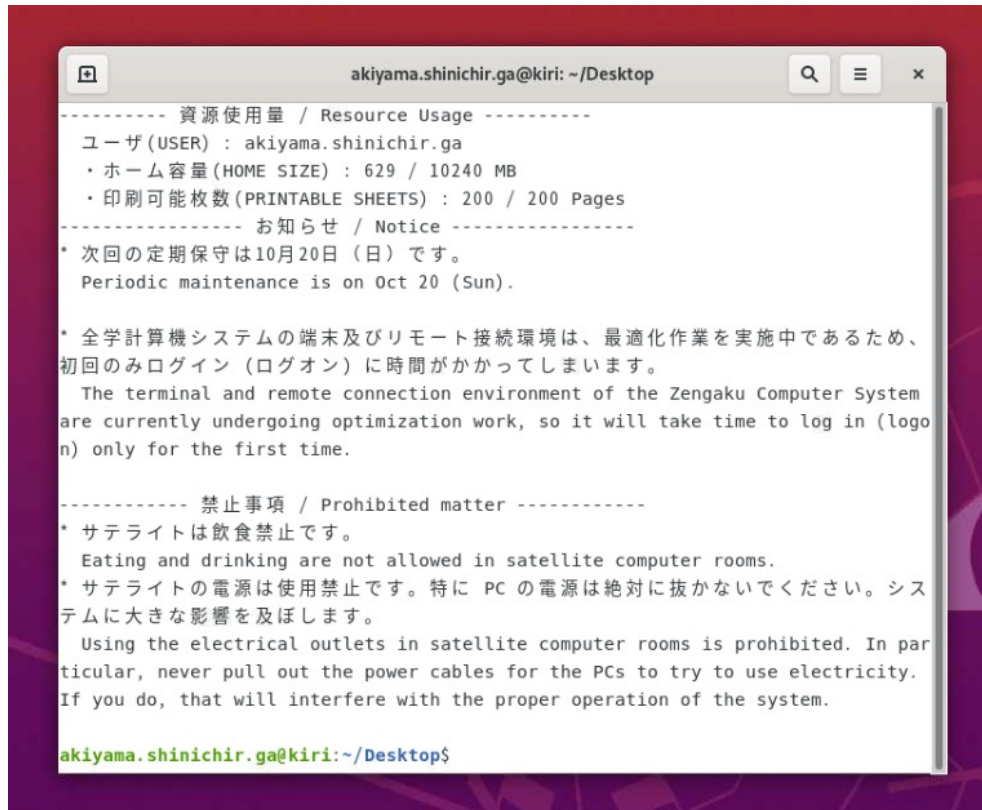
- ・ 電源を入れ, Ubuntu（ウブントゥ）を選ぶ
 - ・ Windowsを選んでしまった人は再起動し, Ubuntuを選ぶ
- ・ ログインする
 - ・ ユーザ名：s+学籍番号の下7桁
 - ・ パスワード：統一認証のパスワード
- ・ 右のような画面になればOK

終了はここから



端末 (Terminal) を起動

- ・ デスクトップで右クリックする
 - ・ 「端末で開く」を選択 (右図参照)
- ・ 下のような画面が立ち上がればOK



```
akiyama.shinichir.ga@kiri: ~/Desktop
----- 資源使用量 / Resource Usage -----
ユーザ (USER) : akiyama.shinichir.ga
・ ホーム容量 (HOME SIZE) : 629 / 10240 MB
・ 印刷可能枚数 (PRINTABLE SHEETS) : 200 / 200 Pages
----- お知らせ / Notice -----
* 次回の定期保守は10月20日 (日) です。
  Periodic maintenance is on Oct 20 (Sun).

* 全学計算機システムの端末及びリモート接続環境は、最適化作業を実施中であるため、
  初回のみログイン (ログオン) に時間がかかってしまいます。
  The terminal and remote connection environment of the Zengaku Computer System
  are currently undergoing optimization work, so it will take time to log in (logo
  n) only for the first time.

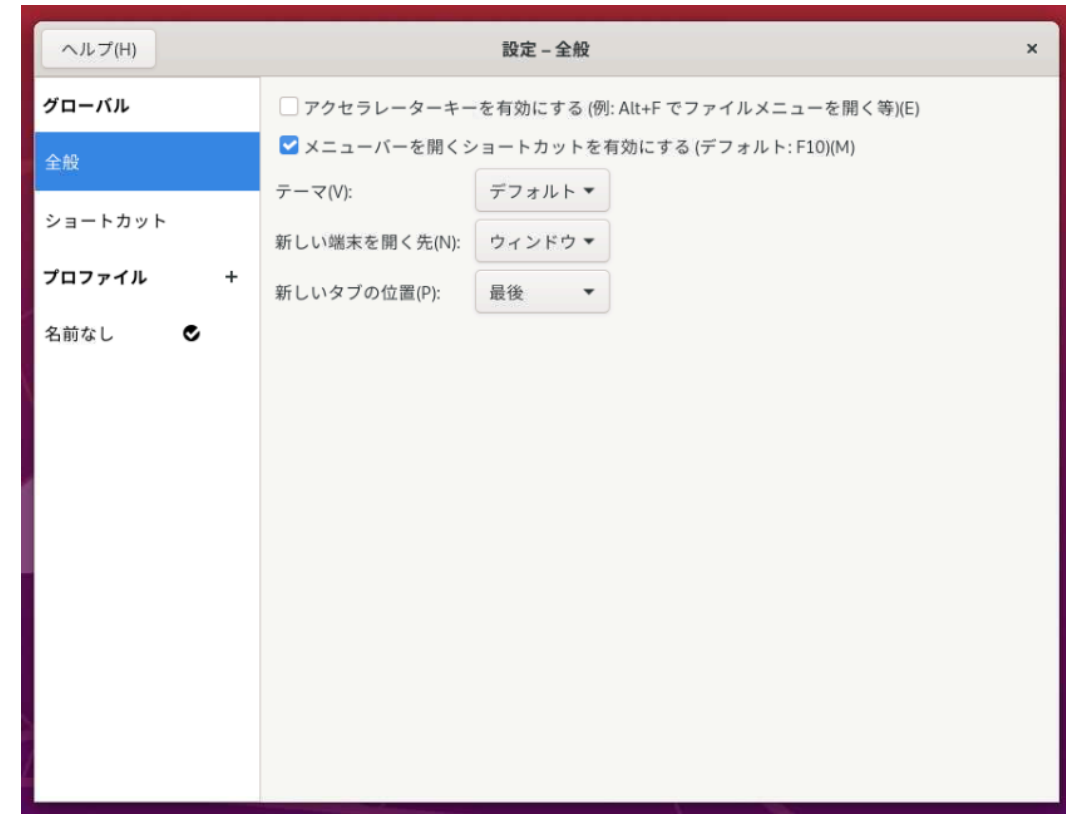
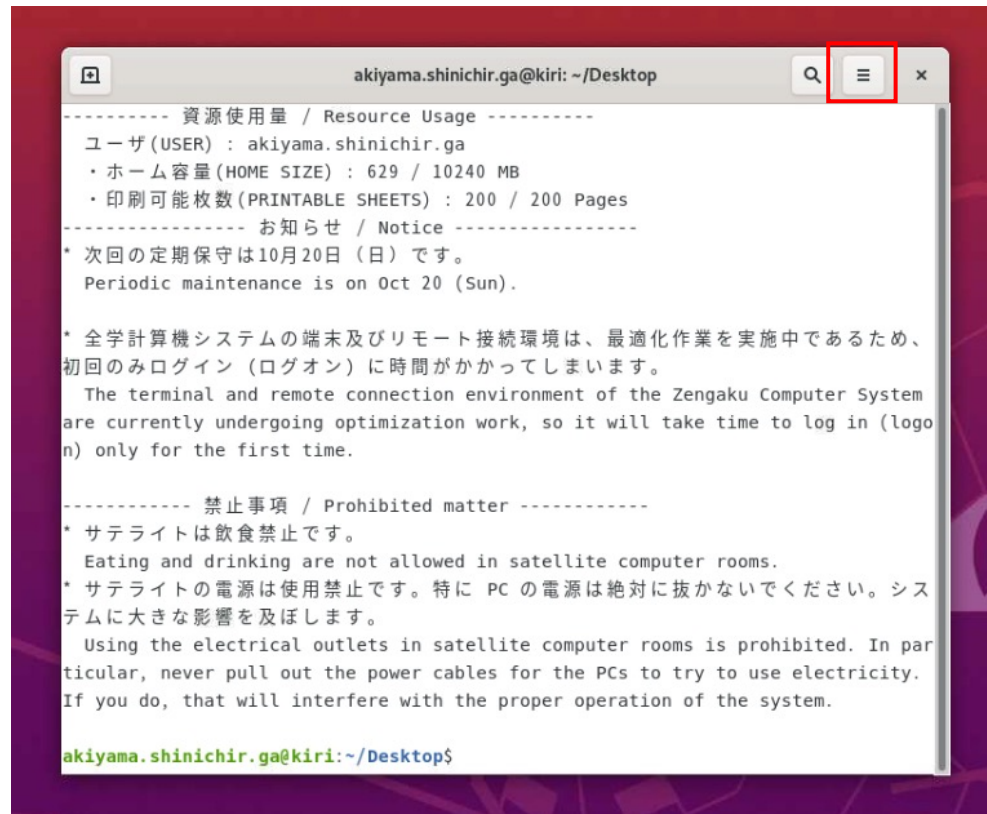
----- 禁止事項 / Prohibited matter -----
* サテライトは飲食禁止です。
  Eating and drinking are not allowed in satellite computer rooms.
* サテライトの電源は使用禁止です。特に PC の電源は絶対に抜かないでください。シス
  テムに大きな影響を及ぼします。
  Using the electrical outlets in satellite computer rooms is prohibited. In par
  ticular, never pull out the power cables for the PCs to try to use electricity.
  If you do, that will interfere with the proper operation of the system.

akiyama.shinichir.ga@kiri:~/Desktop$
```



端末 (Terminal) の設定

- ・ 端末の設定（右上の三本線）をクリック
- ・ 端末のテーマ（色）変更やショートカットコマンドの確認ができる



端末 (Terminal) の表示内容

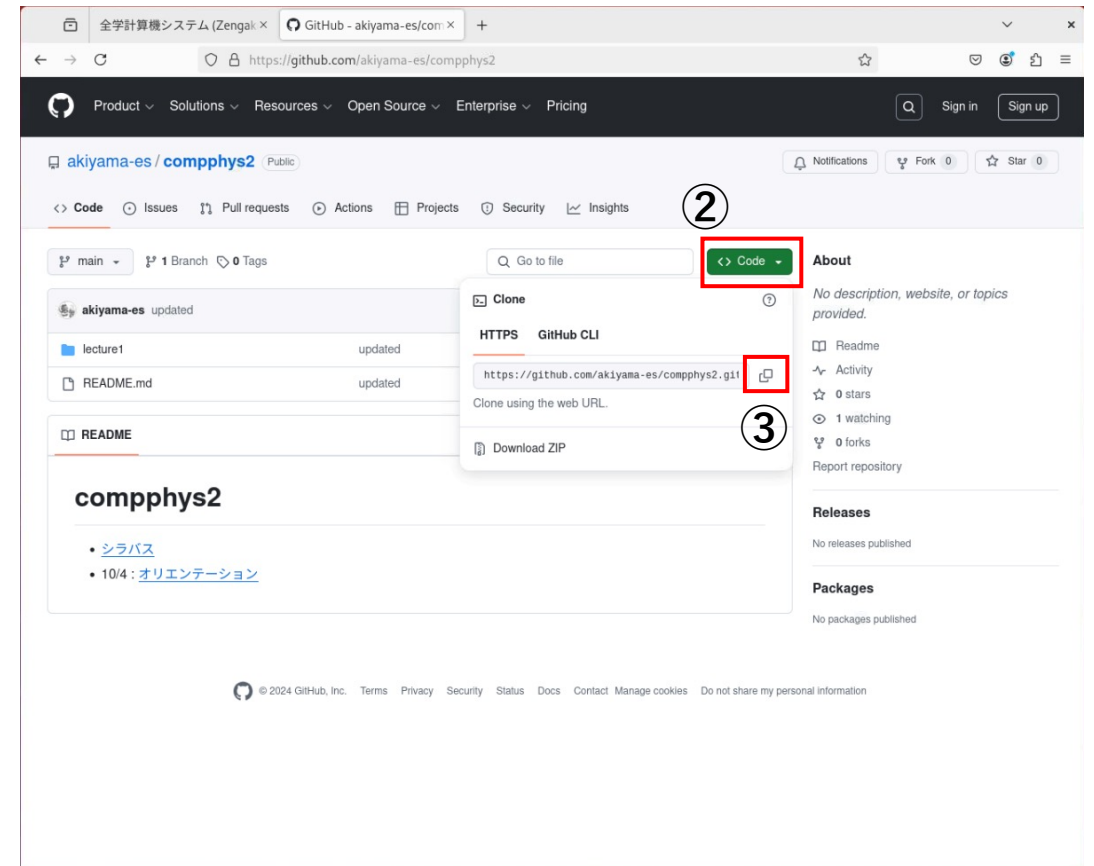
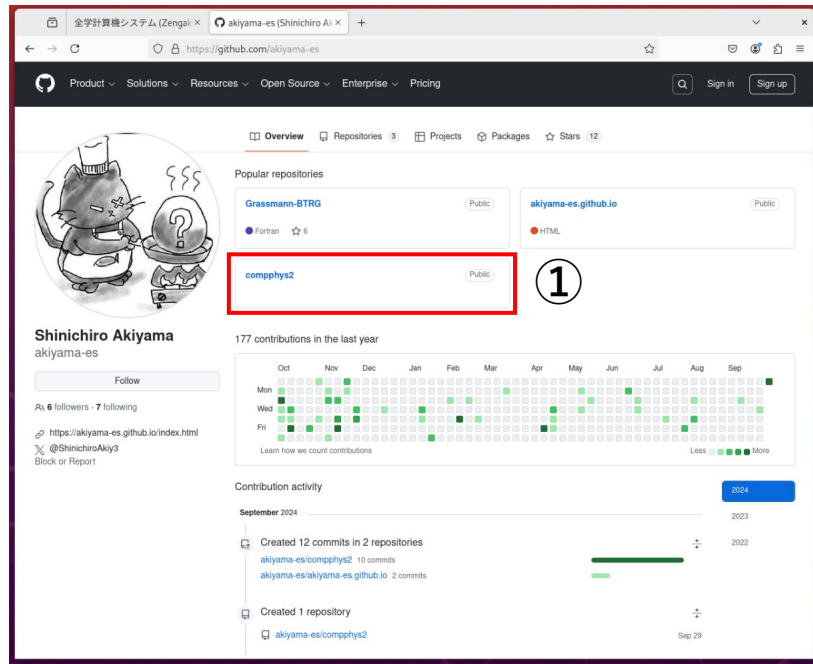
- ・「ユーザ名@コンピュータの名前:今いる場所\$**コマンド入力領域**」

```
akiyama@:~/Desktop$ □
```

- ・ ユーザ名 = akiyama
- ・ コンピュータの名前 = （上では非表示になっています）
- ・ 今いる場所（カレントディレクトリ）= “Desktop” という名前のディレクトリ
- ・ コマンドを入れ, Enterを押すとそのコマンドが今いる場所で実行される
- ・ Terminalを通した計算機の使用に慣れていきましょう
 - ・ 計算機内での場所（ディレクトリ）の移動
 - ・ コマンドの実行を通してやりたい作業を行う（コーディング, プログラムの実行, 等）

講義, 演習資料の入手（この作業は初回のみ）

- ・ ブラウザ（Chrome, Firefox等）から <https://github.com/akiyama-es> にアクセス
- ・ “compphy2”をクリック（①）
- ・ 緑色の“Code”をクリック（②）
- ・ “Copy url to clipboard”をクリック（③）



講義, 演習資料の入手（この作業は初回のみ）

- ・ 端末に戻り, 以下のコマンドを入力し, Enterを押す

```
akiyama@:~/Desktop$ cd ~/
```

- ・ 次に以下のコマンドを入力する（まだEnterは押さない）

```
akiyama@:~$ git clone
```

- ・ “git clone”の後にスペースを入れ, Shift+Ctrl+Vを同時に押す
- ・ 以下のようにってからEnterを押す

```
akiyama@:~$ git clone git@github.com:akiyama-es/compphys2.git
```

講義, 演習資料の入手 (この作業は初回のみ)

- ・ 次のような画面が出ればOK

```
akiyama@:~$ git clone git@github.com:akiyama-es/compphys2.git
Cloning into 'compphys2'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 21 (delta 4), reused 17 (delta 3), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), 191.68 KiB | 631.00 KiB/s, done.
Resolving deltas: 100% (4/4), done.
akiyama@:~$
```

- ・ “compphys2”というディレクトリが出来たはずなので, 以下のlsコマンドを実行して確認

```
akiyama@:~$ ls
compphys2 Desktop dev Documents Downloads Dropbox intel Music Pictures Public slurm-24.11.3 snap Templates Videos work
akiyama@:~$
```

Terminalから演習資料を開く

- ・次のコマンドを実行すると演習資料が開く

```
akiyama@:~$ xdg-open compphys2/lecture1/lecture_material_1.pdf
```

- ・ドキュメントビューアーは立ち上がりましたか？
 - ・立ち上がったらp.17へ移動する
 - ・各自での演習スタート

本日の演習内容

- ・ 全学計算機システムを使ってみる
 - ・ Linuxに入る
 - ・ 端末（Terminal）を使ってみる
 - ・ 演習資料の入手
 - ・ いくつかの基本コマンドを使ってみる（ここから各自で演習）
 - ・ いくつかのエディタを使ってみる
- ・ 全学計算機システムへのリモートアクセス（必要に応じて授業後に各自で）

ディレクトリの移動：cdコマンドを使ってみる

- ・ディレクトリの移動は“cd 移動先のディレクトリ”で行う
 - ・今いる場所（カレントディレクトリ）はホームディレクトリ（Terminal上では~）
- ・ホームディレクトリの中にある“compphys2”というディレクトリへ移動してみる
 - ・以下のコマンドを実行

```
akiyama@:~$ cd compphys2
```

- ・カレントディレクトリが以下のように変わったことを確認

```
akiyama@:~/compphys2$
```

- ・“~/compphys2”は「ホームディレクトリの中のcompphy2」という意味

ディレクトリ内の確認：lsコマンドを使ってみる

- ・ディレクトリの中身の確認は“ls”で行う

```
akiyama@:~/compphys2$ ls
```

- ・lsコマンドを実行すると次のようになる

```
akiyama@:~/compphys2$ ls  
lecture1 README.md vnc  
akiyama@:~/compphys2$
```

- ・“README.md”はディレクトリではなくファイルなのでcdコマンドでは移動できない
 - ・試しに“cd README.md”と入力して実行してみると？

```
akiyama@:~/compphys2$ cd README.md  
bash: cd: README.md: Not a directory  
akiyama@:~/compphys2$
```

- ・端末がこのように教えてくれるので、コマンド実行後に端末が何か言ってきた際は必ずチェックする癖をつけましょう

新規ディレクトリの作成：mkdirコマンドを試してみる

- “lecture1”というディレクトリへ移動する（最後の/は無くてもいい）

```
akiyama@:~/compphys2$ cd lecture1/
```

- “lecture1”の中身を確認

```
akiyama@:~/compphys2/lecture1$ ls  
lecture_material_1.pdf
```

- 新しいディレクトリの作成は“mkdir ディレクトリの名前”で行う

- ここでは“project1”という名前のディレクトリを作ってみよう

```
akiyama@:~/compphys2/lecture1$ mkdir project1
```

- コマンド実行後, lsコマンドから確認

新規ファイルの作成：touchコマンドを使ってみる

- ・まず, 今作った“project1”へ移動する

```
akiyama@:~/compphys2/lecture1$ cd project1/
```

- ・新しいファイルの作成は“touch ファイルの名前”で行う
 - ・ここでは“myname.txt”という名前のテキストファイルを作ってみよう

```
akiyama@:~/compphys2/lecture1/project1$ touch myname.txt
```

- ・コマンド実行後, lsコマンドから確認

コマンドは他にもたくさんある

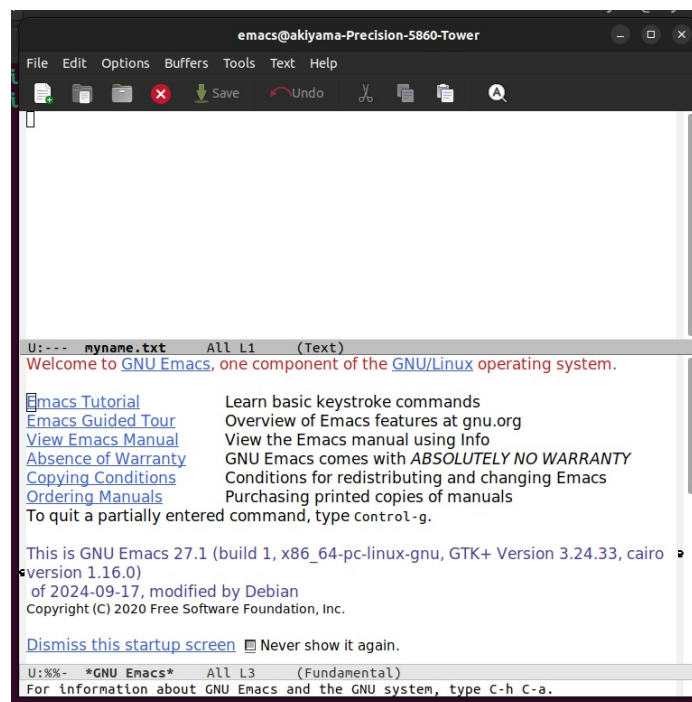
- ・他にも色々なコマンドが今後授業で登場すると思いますが
 - ・全てのコマンドを片っ端から覚える必要は全くありません
 - ・忘れたら, その都度ネット等で調べる癖をつけましょう
 - ・使っていく中で, 使用頻度の高いものから自然と覚えていくのが理想です

ファイルの編集：emacsを使ってみる

- ここではemacsというエディタを使って“myname.txt”を編集してみる
 - 以下のコマンドを実行するとemacsによって“myname.txt”が開く

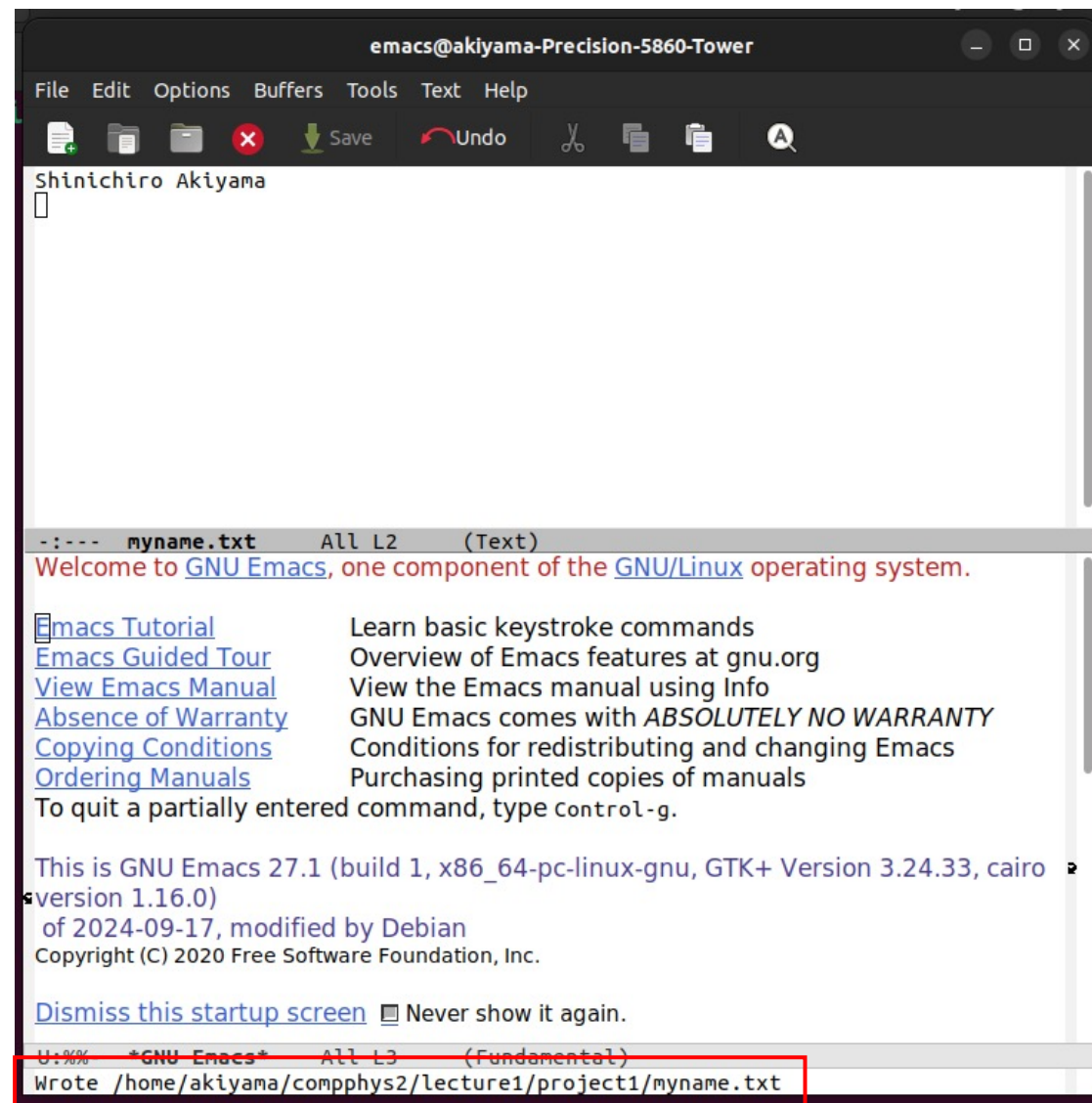
```
akiyama@:~/compphys2/lecture1/project1$ emacs myname.txt
```

- 以下のような画面が立ち上がればOK



ファイルの編集：emacsを使ってみる（続き）

- 自分の名前を入力してみよう
- 入力したらファイルを保存する
 - “Ctrl+X+S”を同時に押す
 - 保存できたら赤枠のメッセージが現れる
- 保存後, emacsを終了する
 - “Ctrl+X+C”を同時に押す
 - emacsの画面が消えればOK



テキストファイルの確認：catコマンドを使ってみる

- ・テキストファイルの確認は“cat ファイルの名前”で行う

```
akiyama@~/compphys2/lecture1/project1$ cat myname.txt
```

- ・先ほど入力した自分の名前が表示されればOK

```
akiyama@~/compphys2/lecture1/project1$ cat myname.txt  
Shinichiro Akiyama
```

ファイルの編集：viを使ってみる

- ・ 今度はviを使ってファイルを編集してみる
 - ・ 以下のコマンドを実行すると，“myname.txt”の編集が可能になる

```
akiyama@~/compphys2/lecture1/project1$ vi myname.txt
```

- ・ Terminalが以下のような画面に変わればOK

```
Shinichiro Akiyama
```

```
□
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

ファイルの編集：viを使ってみる（続き）

- viにはいくつかのモードがある
- 起動時は「ノーマルモード」になっている
- 文字を入力するには「挿入モード」へ移る
 - “i”を押すと文字入力が可能になる
 - 何か入力してみよう
- 「挿入モード」の終了方法
 - “Esc”を押す

```
Shinichiro Akiyama  
Computational Physics II
```

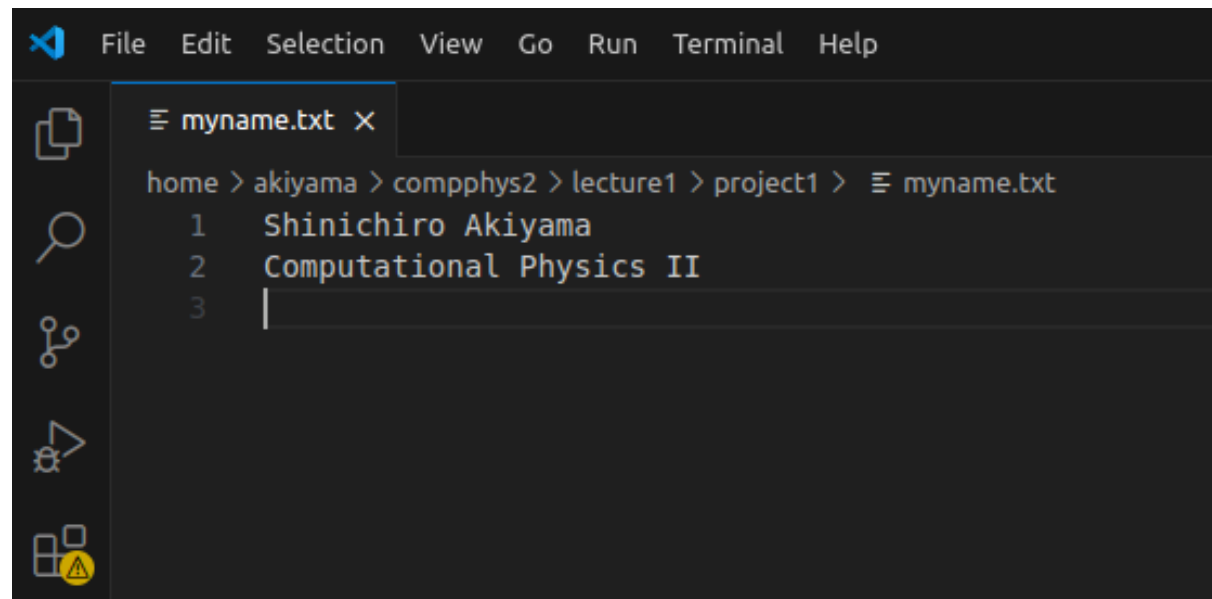
```
-- INSERT --
```


ファイルの編集：VS codeを使ってみる

- ・ 今度はVisual Studio Code (VS code)を使ってファイルを編集してみる
 - ・ 以下のコマンドを実行するとVS codeによって“myname.txt”が開く

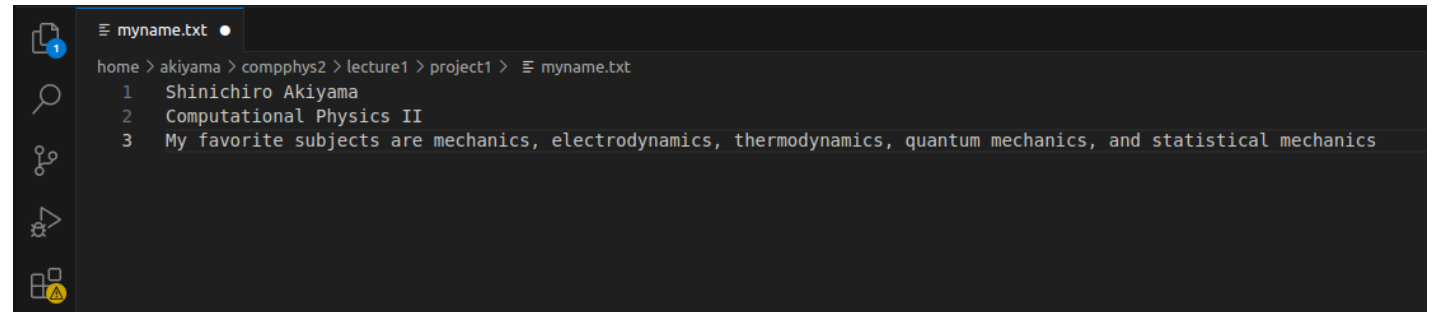
```
akiyama@:~/compphys2/lecture1/project1$ code myname.txt
```

- ・ 以下のような画面が立ち上がればOK



ファイルの編集：VS codeを使ってみる（続き）

- ・何か文字を入力してみよう
- ・入力したらファイルを保存する
 - ・“Ctrl+S”を同時に押す
- ・保存後, VS codeを終了する
 - ・“Ctrl+Q”を同時に押す
 - ・VS codeの画面が消えればOK
- ・catコマンドを使い, 編集内容が保存されているか確認してみよう



```
home > akiyama > compphys2 > lecture1 > project1 > myname.txt
1  Shinichiro Akiyama
2  Computational Physics II
3  My favorite subjects are mechanics, electrodynamics, thermodynamics, quantum mechanics, and statistical mechanics
```

エディタには色々な種類がある

- emacs, vi以外にも色々あります
 - 全てを使いこなす必要はありません
 - 自分が使いやすいと思うものを自由に使ってください
- 使い方がなかなか覚えられない？
 - 無理に覚えようとしなくて大丈夫です
 - 使い方が分からなくなったら、その都度ネット等で調べる癖をつけましょう
 - AIに尋ねるのも有効です
- コード開発等の作業を手元のパソコン（ローカル）で行う場合, VS codeは便利
 - 様々な機能を個人のニーズに合わせて導入できる

ディレクトリ間の移動

- “project1”から“lecture1”のディレクトリへ戻ってみる
 - “cd ../”で一つ上のディレクトリへ戻ることができる

```
akiyama@:~/compphys2/lecture1/project1$ cd ../
```

- 実行すると次のようになる

```
akiyama@:~/compphys2/lecture1$
```

- “cd”だけ実行すると、どのディレクトリにいてもホームディレクトリに戻る
 - ホームディレクトリに戻ってみてください

```
akiyama@:~/compphys2/lecture1$ cd
akiyama@:~$
```


Tabキーによる補完

- ・ 再び“lecture1”ディレクトリへ移動してみますが, 少し楽をしていきましょう
 - ・ まず“cd”と打ってスペースを入れる
 - ・ 次にTabキーを押してみる
 - ・ すると, 今いる場所から移動できるディレクトリの一覧が表示される

```
akiyama@:~$ cd
.cache/      Desktop/    .dotnet/    Dropbox/    .intel/    .local/    .nv/        Public/    .ssh/       Videos/
compphys2/   dev/       Downloads/  .dropbox-dist/ intel/      .mozilla/  Pictures/   slurm-24.11.3/ Templates/  .vscode/
.config/     Documents/ .dropbox/   .emacs.d/   .julia/    Music/     .pki/       snap/      .thunderbird/ work/
akiyama@:~$ cd
```

Tabキーによる補完（続き）

- “compphy2”へ移動するには“cd compphy2”と打ってもいいですが…
- “cd c”まで打ってTabキーを押してみましょう

```
akiyama@:~$ cd c
```

- “cd compphy2”と補完されるはず（補完されない場合はもう一文字打ってみる）

```
akiyama@:~$ cd compphys2/
```

- その状態でもう一度Tabキーを押してみましょう

```
akiyama@:~$ cd compphys2/  
.git/      lecture1/ vnc/  
akiyama@:~$ cd compphys2/
```

- また移動できる候補一覧が表示されます

Tabキーによる補完（続き）

- “l”だけ打ってTabキーを押してみましょう

```
akiyama@:~$ cd compphys2/l
```

- 再び補完されて次のようになるはず

```
akiyama@:~$ cd compphys2/lecture1/
```

- これでEnterで実行すれば“lecture1”に移動できたはず
- Tabキーはどんどん使いましょう
- Tabキーによる補完はコマンドに対しても有効です
 - 例えば, 今日の最初に演習資料を開くために使ったコマンドは？
 - “xdg”まで打ってTabキーを押してみましょう. コマンド候補が閲覧できるはず

ここまで終わったら

- ・お疲れ様でした
- ・GitHubへのリンクはTeamsのチャンネルにもピン留めしてあります
- ・Teamsコードはmanabaから確認してください
- ・今日使ったLinuxデスクトップへのリモートアクセスを設定する方法が掲載してあるので各自で授業後にやってみてください

そもそもUbuntuやLinuxとは？

- UbuntuとはLinuxディストリビューションの一種であり, Linuxとはそのコアとなるオペレーティングシステム
- Ubuntuの操作
 - Linuxコマンドによる操作（キーボード主体）
 - デスクトップ操作（マウス主体）
- スーパーコンピュータ（Ex. 富岳） は基本Linuxで動いている
- MacはLinuxとは異なるOSだが, 共にUNIXから派生しているため同様のコマンド操作が可能（余談）