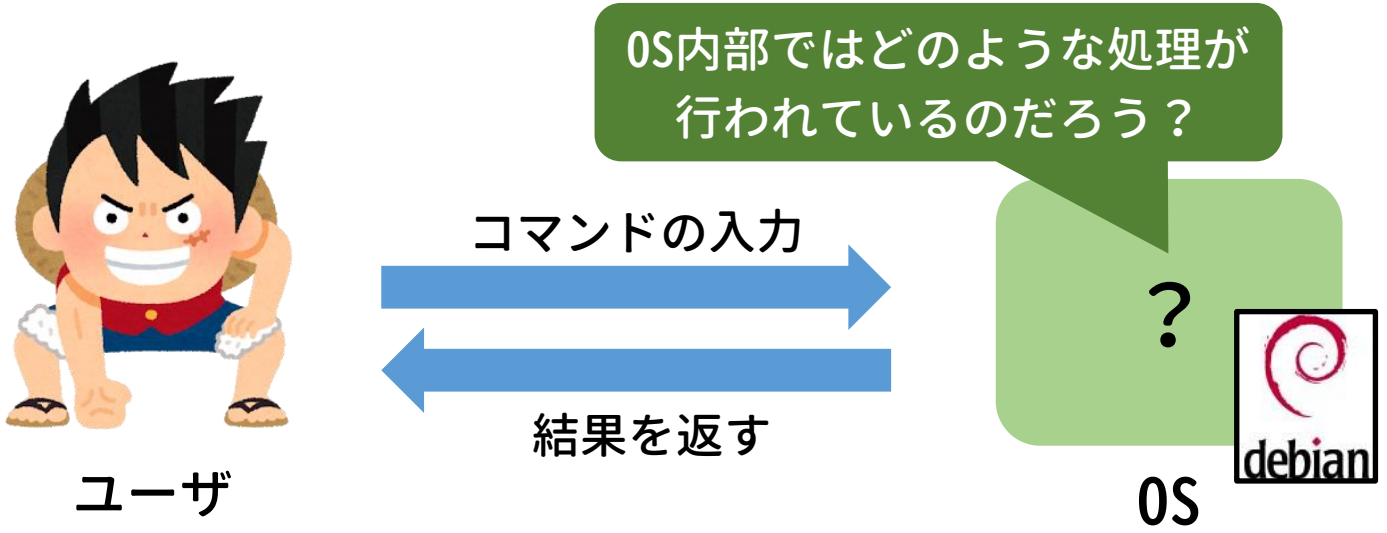
情報実験・第3回（2025/05/02）

最低限 Unix（Linux）II
～シェル・テキストエディタ～

北海道大学大学院 理学院 宇宙理学専攻
修士課程 2年
山本 峻大／Yamamoto Takahiro

本日の内容

■コマンドが実行される仕組みについて



■テキストエディタについて

- テキストエディタとは？
- viの使い方

イラスト：いらすとや（ https://www.irasutoya.com/2021/01/onepiece.html ）

目次

1. OS とカーネルとシェル

復習：OS (Operating System)

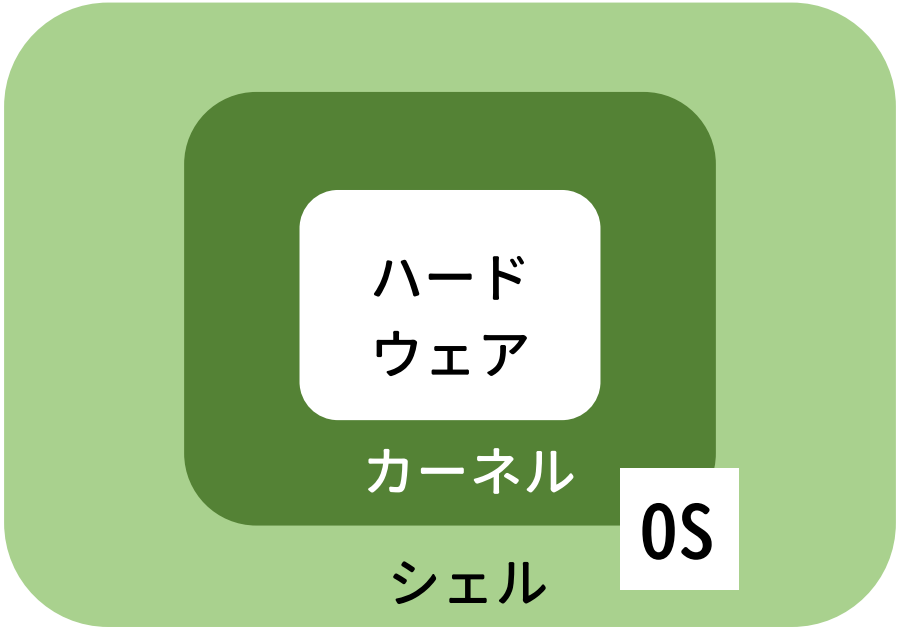
■OSとは？

- 計算機を管理，操作するための基本ソフトウェア
- アプリケーションソフトウェアとハードウェアの仲介役

イラスト：いらすとや（ https://www.irasutoya.com/search/label/%E5%91%A8%E8%BE%BA%E6%A9%9F%E5%99%A8 ）

OS の大まかな概念図

カーネル(kernel:核)とシェル(shell:殻)で
ハードウェアを包んでいる



OS はハードウェアにかぶせた服

カーネルとシェル

カーネル（kernel，核）

- ハードウェアを直接制御する OS の中心部
  - プロセス管理（CPU のスケジューリングなど）
  - メモリ管理（主記憶メモリの配分など）
  - ファイルシステム管理
  - プロセス間通信（リダイレクト、パイプなど）
  - デバイス制御（第５回講義）

シェル（shell，殻）

- 入力文字列をコマンドとして解釈する窓口
  - コマンドインタプリタの一つ

OS の大まかな構造

各種アプリケーション（コマンド群など）

ウィンドウシステム
（第１回講義）

シェル

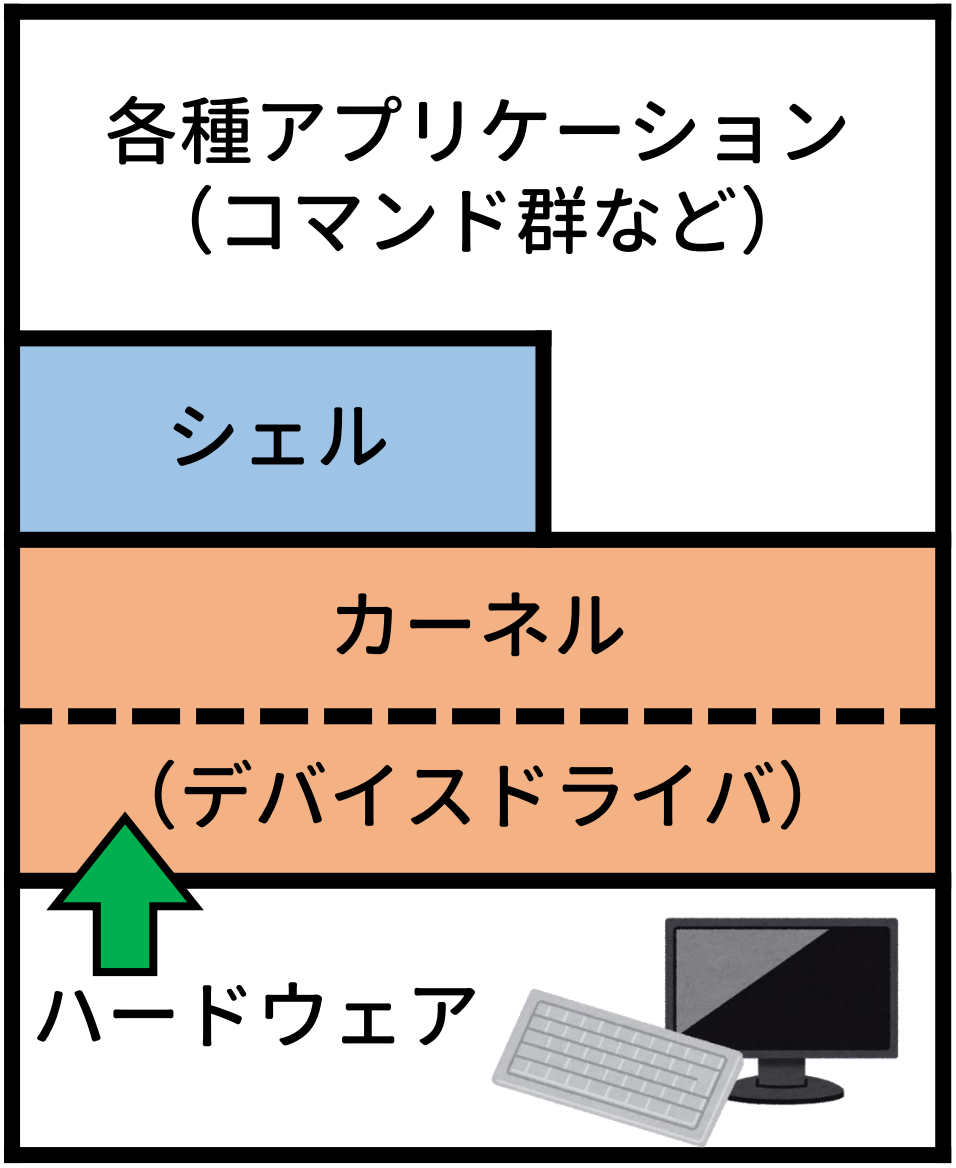
カーネル

（デバイスドライバ）

ハードウェア

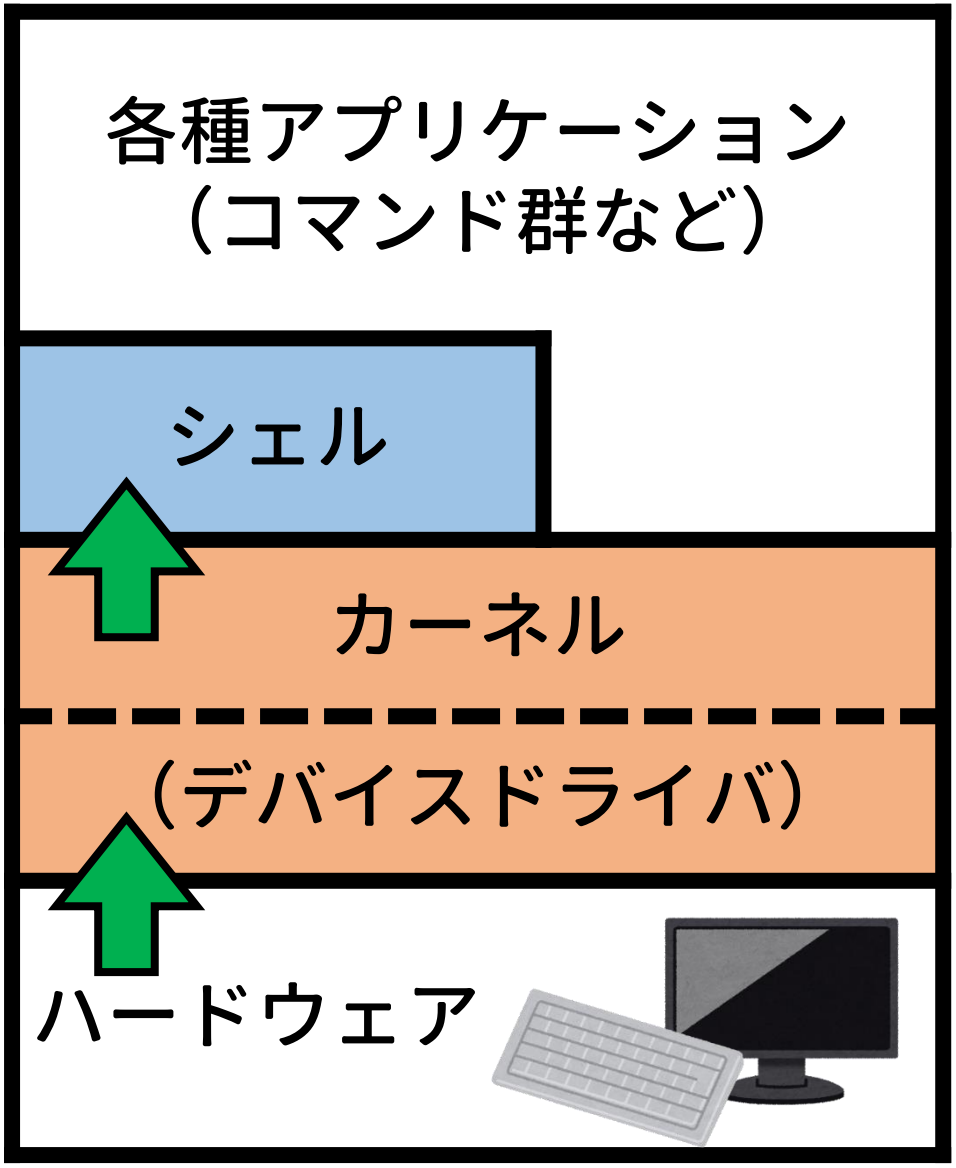
図は新 The UNIX Super Text 上に基づく

OS のおしごと



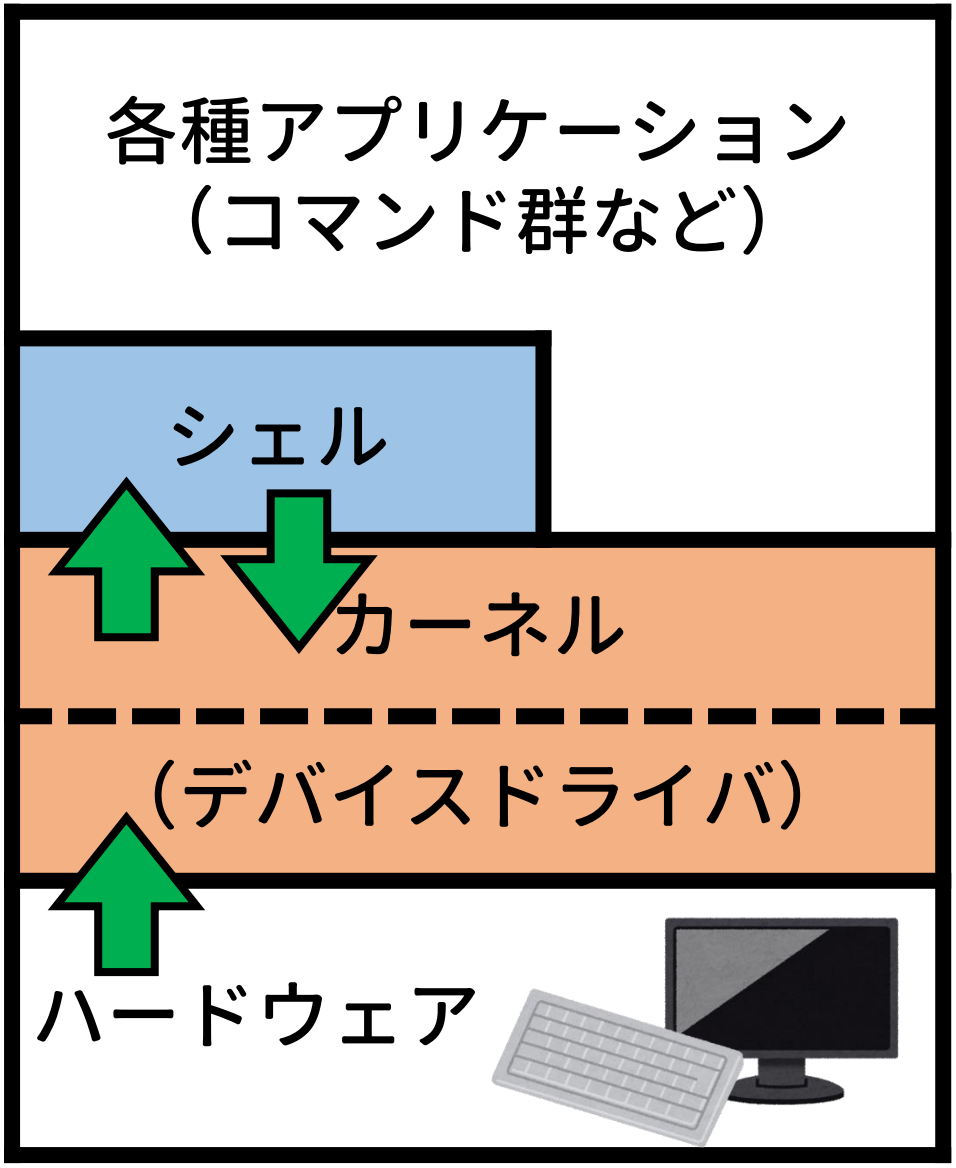
1. キーボード入力
   ($ cat himitsu.txt
   とか)

OS のおしごと



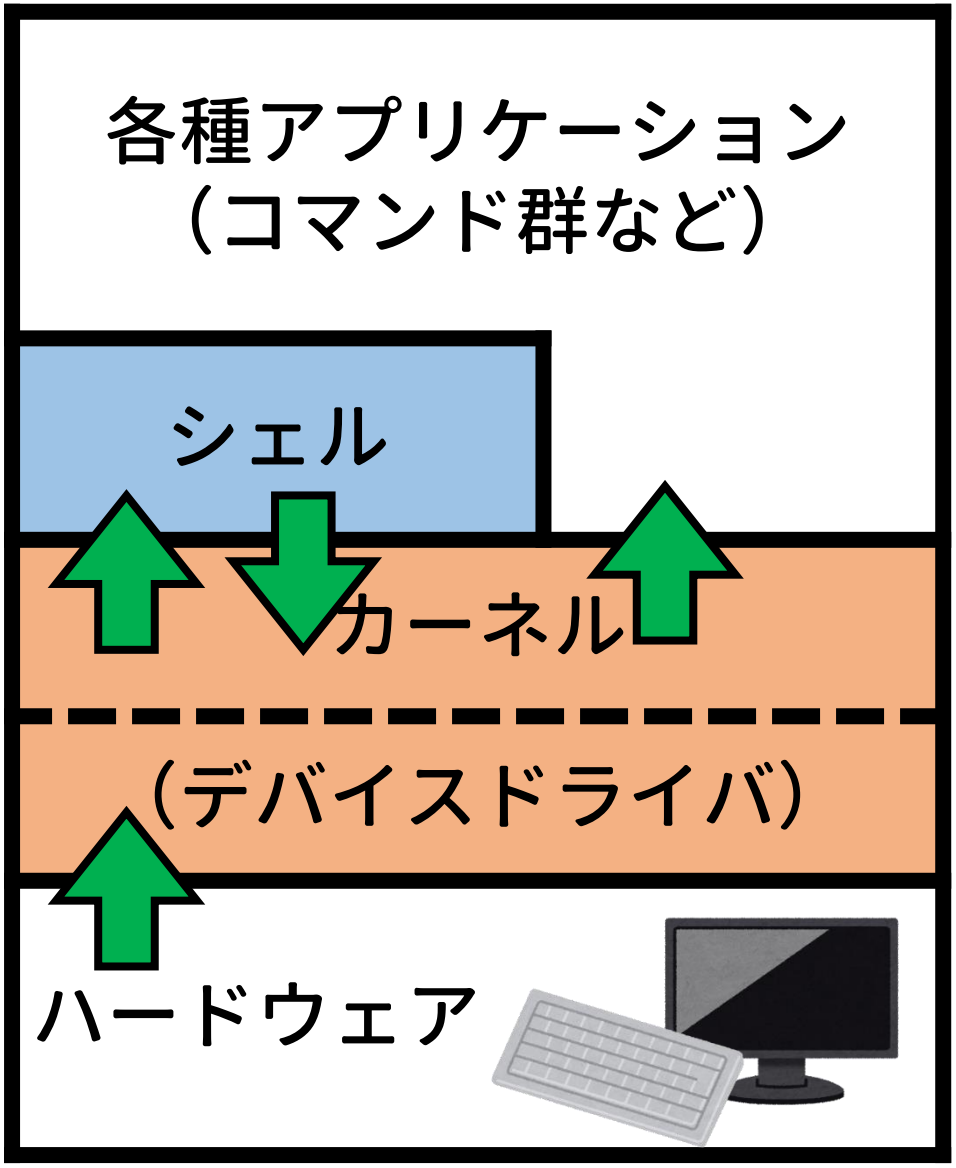
1. キーボード入力
2. カーネルがシェルへ入力を送る

OS のおしごと



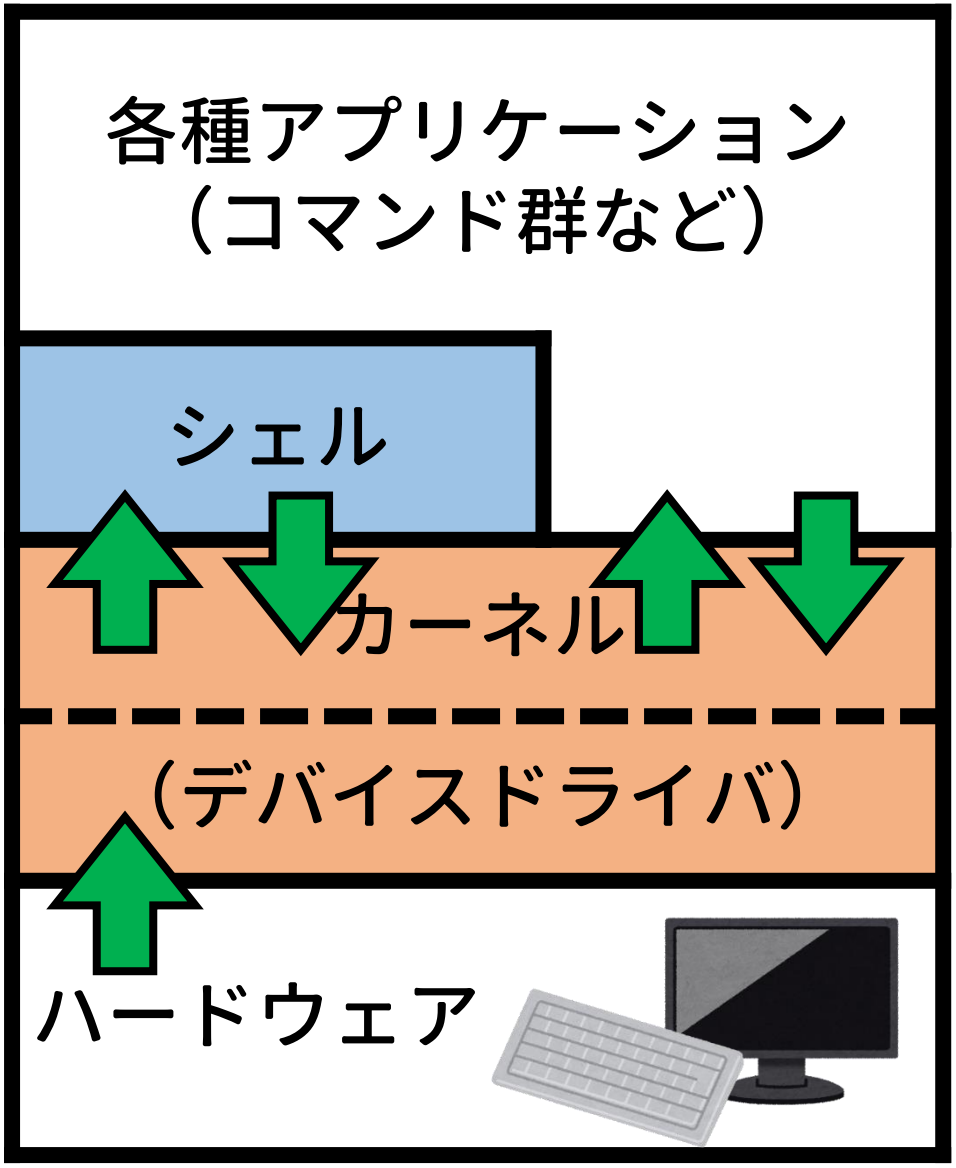
1. キーボード入力
2. カーネルがシェルへ入力を送る
3. シェルが入力をコマンドに解釈し、起動指示
   (コマンド: cat,
    引数 : himitsu.txt)

OS のおしごと



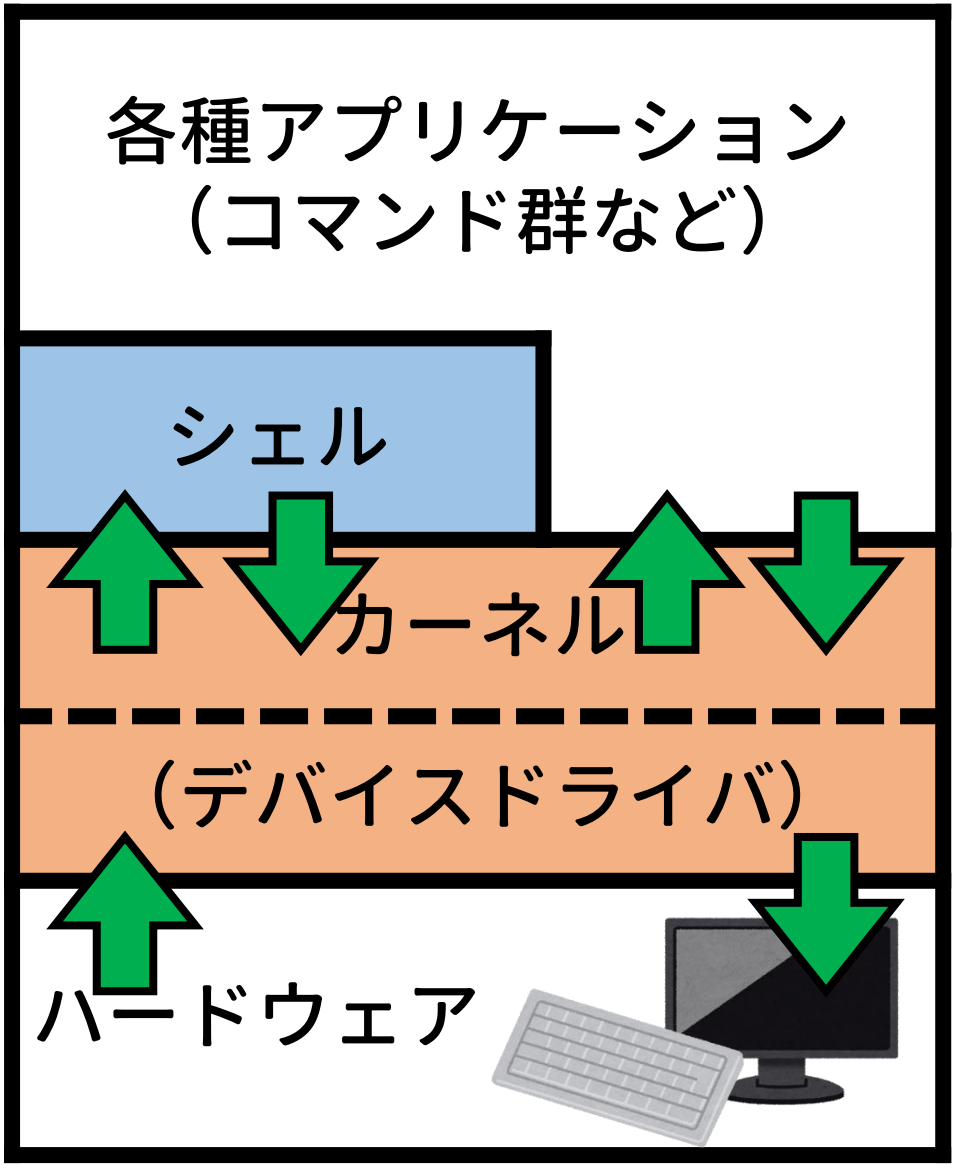
1. キーボード入力
2. カーネルがシェルへ入力を送る
3. シェルが入力をコマンドに解釈し、起動指示
4. カーネルがアプリケーションにコマンドの実行要求
   （cat コマンドで引数を実行して）

OS のおしごと



1. キーボード入力
2. カーネルがシェルへ入力を送る
3. シェルが入力をコマンドに解釈し、起動指示
4. カーネルがアプリケーションにコマンドの実行要求
5. アプリケーションが実行結果を返す

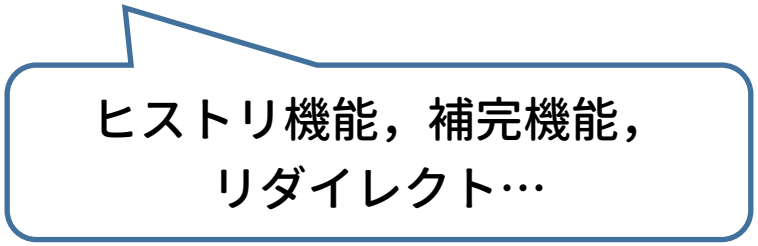
OS のおしごと



1. キーボード入力
2. カーネルがシェルへ入力を送る
3. シェルが入力をコマンドに解釈し、起動指示
4. カーネルがアプリケーションにコマンドの実行要求
5. アプリケーションが実行結果を返す
6. カーネルがディスプレイに結果を表示
(秘密のあれこれが表示！)

シェルの特徴

1. CLI であることが多い
2. 多様なシェルが存在
   - sh, bash, csh, dash, tcsh, zsh …



INEX では **bash** を使う
色々便利な機能を持った標準的なシェル

復習：ユーザインタフェース

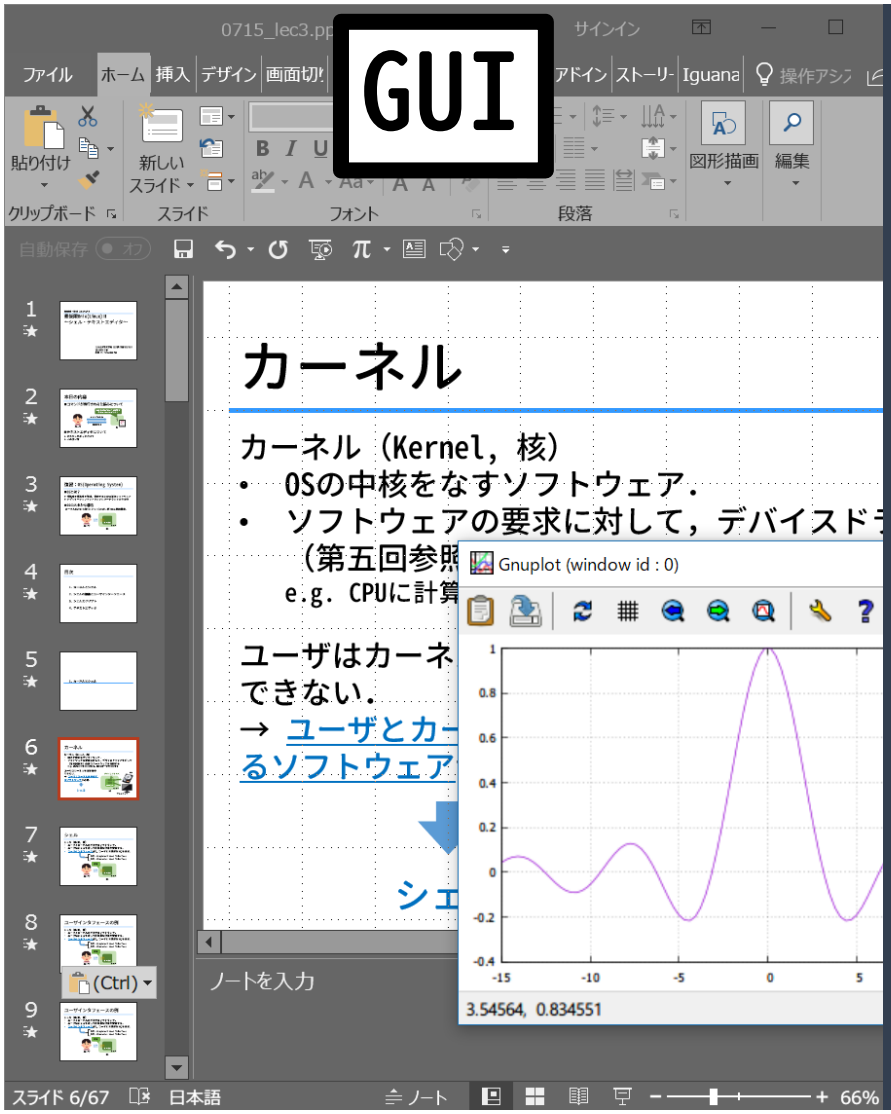
ユーザインタフェース（UI）

- ユーザとコンピュータの間で情報をやりとりする
  ソフトウェア・ハードウェア
  - 例：音声入力、タッチパネル、画面のデザインなど

- 計算機に対する UI は主に２種類存在
  - GUI: Graphical User Interface
    - ウィンドウやボタンなどをマウス操作
  - CLI: Command-Line Interface
    - キーボードで文字列を入力して操作
    - CUI (Character User Interface) という
      似た意味の言葉もある

ユーザインタフェースの例

GUI



CLI

```
Debian GNU/Linux 12 tty1

joho24 login: tyama
Password:

tyama@joho24: ~$ pwd
 /home/tyama

tyama@joho24: ~$ rm -rf /
```

GUI と CLI の特徴

GUI の特徴

- X Window System など（第１回）
- マウス・タッチパネル等を使って直感的に作業できる
- 計算機への負荷が大きい（CLI よりも計算機の動作が複雑）

CLI の特徴

- ほとんどのシェルは CLI
- コマンドを覚えればキーボードだけで何でもできる
- 計算機への負荷が小さい

  → サーバ業務，トラブル対処に強い
- 単純な繰り返し作業に向く（本日の課題にも関連）

シェル（Bash）の環境設定

環境

- 言語やパスなど設定情報（$ env で確認してみよう）
- 各アプリケーションソフトウェアは、呼び出されたシェルの環境下で動く

環境変数

- 設定内容を格納する変数の一種
  変数 LC_ALL に代入されている値の例：ja_JP.UTF-8
- 設定ファイルで設定される（~/.bashrc など）
- 手動で書き換えることもできる（シェルによって違う）
  例：export LC_ALL=en_US.UTF-8（Bash の場合）
  （実技編で実際に書き換えてみましょう）

環境設定

```
tyama@joho24:~$ echo $LC_ALL   変数 LC_ALL に代入されている値の確認
en_US.UTF-8   言語は英語・国、地域はアメリカ・文字コードは UTF-8
tyama@joho24:~$
```

環境設定

```
tyama@joho24:~$ echo $LC_ALL
en_US.UTF-8
tyama@joho24:~$ date
Fri May 10 12:44:46 JST 2024
tyama@joho24:~$
```

現在の時間が英語で表示される

環境設定

```
tyama@joho24:~$ echo $LC_ALL
en_US.UTF-8
tyama@joho24:~$ date
Fri May 10 12:44:46 JST 2024
tyama@joho24:~$ export LC_ALL=ja_JP.UTF-8
tyama@joho24:~$
```

変数の値を変更

環境設定

```
tyama@joho24:~$ echo $LC_ALL
en_US.UTF-8
tyama@joho24:~$ date
Fri May 10 12:44:46 JST 2024
tyama@joho24:~$ export LC_ALL=ja_JP.UTF-8
tyama@joho24:~$ echo $LC_ALL  変数 LC_ALL に代入されている値の確認
ja_JP.UTF-8
tyama@joho24:~$
```

環境設定

```
tyama@joho24:~$ echo $LC_ALL
en_US.UTF-8
tyama@joho24:~$ date
Fri May 10 12:44:46 JST 2024
tyama@joho24:~$ export LC_ALL=ja_JP.UTF-8
tyama@joho24:~$ echo $LC_ALL
ja_JP.UTF-8
tyama@joho24:~$ date
2024年　5月 10日 金曜日 12:45:22 JST 現在の日時が日本語で表示される
tyama@joho24:~$
```

環境設定

```
tyama@joho24:~$ echo $LC_ALL
en_US.UTF-8
tyama@joho24:~$ date
Fri May 10 12:44:46 JST 2024
tyama@joho24:~$ export LC_ALL=ja_JP.UTF-8
tyama@joho24:~$ echo $LC_ALL
ja_JP.UTF-8
tyama@joho24:~$ date
2024年  5月 10日 金曜日 12:45:22 JST
tyama@joho24:~$
```

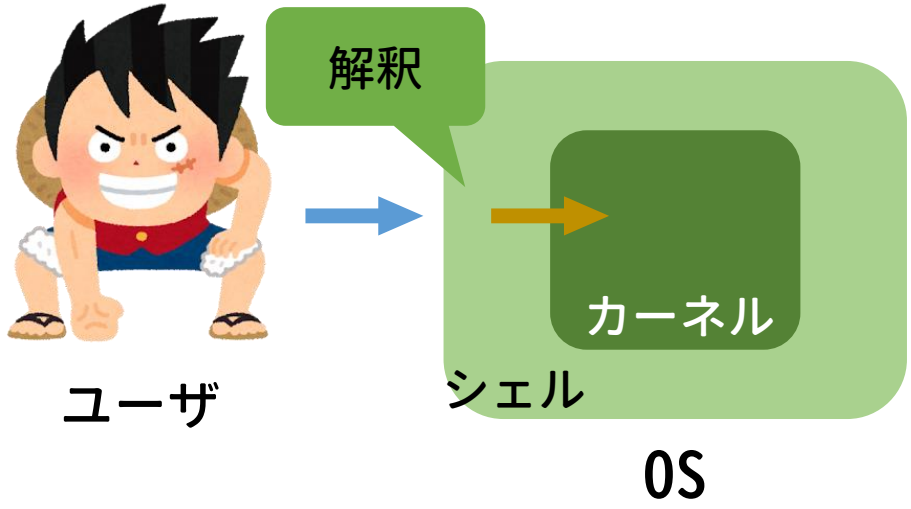
前半のまとめ

カーネル

- OSの中核のソフトウェア
- ソフトウェアの要求に対して必要なハードウェアを制御

シェル

- 入力からコマンドを解釈するソフトウェア
- ユーザインタフェースとして CLI を提供
- ユーザはシェルを通して計算機に作業を要求



前半のまとめ

シェルの基本的機能

- コマンドインタプリタの提供
- 環境の設定
  - アプリケーションソフトウェア間で共用される情報を設定できる

前半の実習では…

シェルに慣れる

- シェル（bash）の各種機能を試してみよう！

2. シェルスクリプト

シェルスクリプト

シェルスクリプト

… コマンドを実行順に並べて記述したファイル
(script = 台本)

台本を読むように
コマンドを連続して実行できる

スクリプトを書くメリット

- 繰り返し作業の手間を省ける
  - 制御構造を利用したプログラミングが可能
- 人為的ミスを防げる
- 似たような作業をするときに再利用ができる
  - スクリプトファイルの資源化が可能

なぜシェルスクリプトか？

他言語と比較したメリット

- 高水準の処理対象を簡単に扱える
  - ファイルやディレクトリの操作に強い
- 移植性が高い
  - POSIX に準拠すれば多くの Linux で動く
  - どんな Linux システムでもインストールされている

POSIX…IEEE が定めた規格。シェルに限らず、ライブラリ、コマンド、セキュリティ、他言語 (C, Fortran, Ada) を標準化

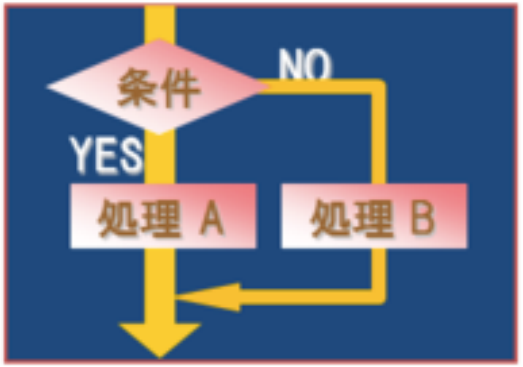
制御構造 ～アルゴリズムの基本～

順次構造

一方向に順序立てて処理を行う構造

選択構造

条件に応じて処理を分岐する構造



反復構造

同じ処理を反復する構造

これらの組み合わせで
色々な作業が可能となる！

シェルスクリプトの具体例 ①

要求

- source.txt のバックアップを取る
- 大切なファイルを別名でも保存する
- 念のため元のファイルは残す

手法

- 日付をシェル変数に格納
- ファイル名を backup_日付.txt としてコピー

```
1 #!/bin/bash
2
3 NICHI=$(date '+%Y-%m-%d')
4
5 cp source.txt backup_${NICHI}.txt
6
```

シェルスクリプトの具体例 ②

要求

- 通し番号が付くファイルの作成
- 1-50，51-100 で名前の付け方を変える

手法

- 通し番号の変数を利用する
- 選択構造（if）と反復構造（while）を組み合わせる

```
#!/bin/bash

num=1

while [ $num -le 100 ]
do
	if [ $num -le 50 ]; then
		echo "${num}!!!" > small_${num}.txt
	elif [ $num -ge 51 ]; then
		echo "${num}!!!" > large_${num}.txt
	fi
	num=$((num + 1))
done

```

3. テキストエディタ

テキストエディタとは？

- テキストファイル（テキストデータのみからなるファイル）の編集を目的とするアプリケーションソフトウェア
  — プログラム編集用のソフトウェアが起源

- 通常の文章からプログラム，各種設定ファイルの作成，編集まで幅広く使える

- 種類が豊富にある
  （例：vi，vim，emacs，nano，メモ帳，Terapad，
  秀丸エディタ，VS Code，NeoVim など…）

本日の実習では **vi** を用いる

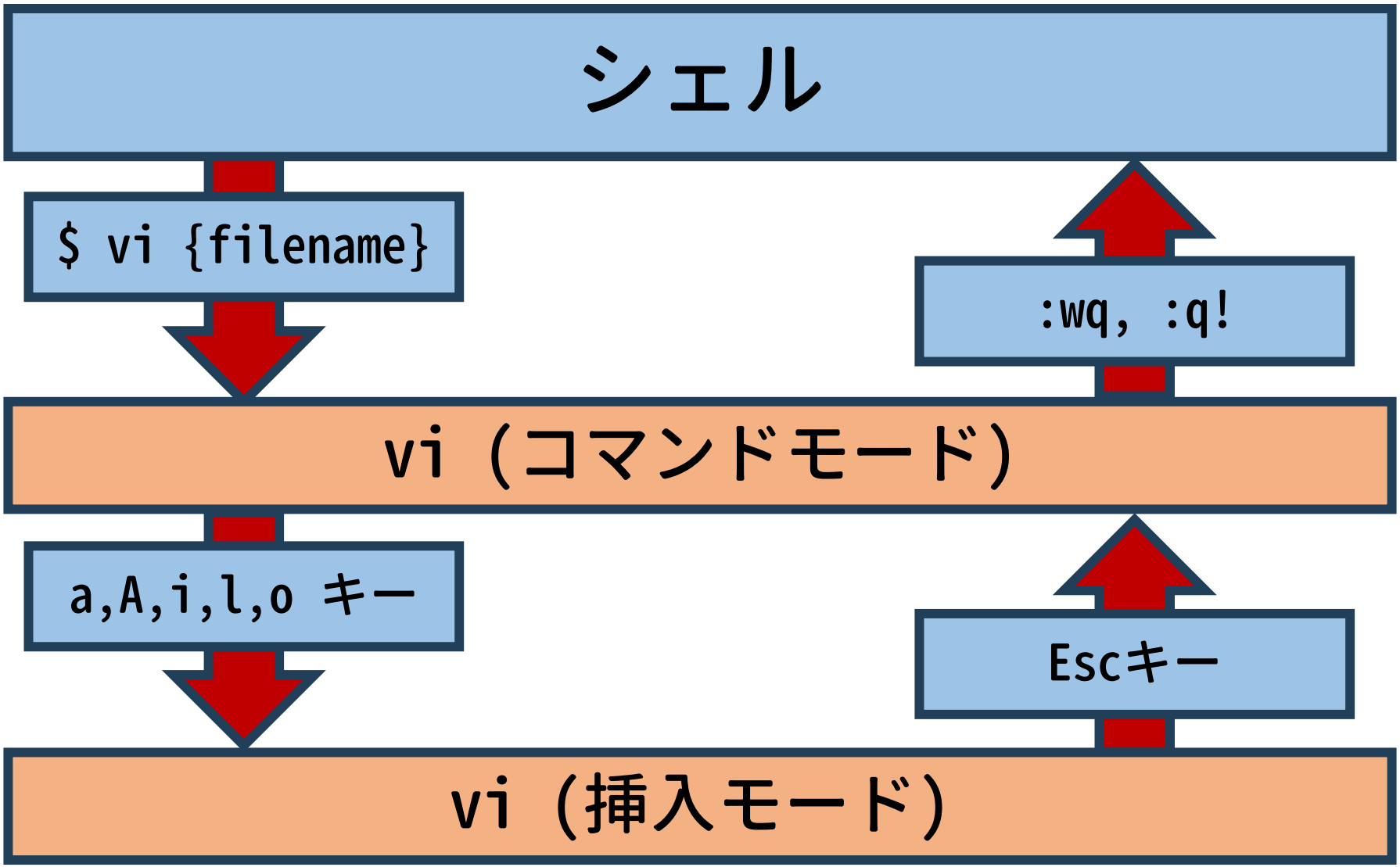
vi ～困ったときに頼れるアイツ～

テキストエディタの一つ
Unix黎明期から使われている由緒正しいエディタ

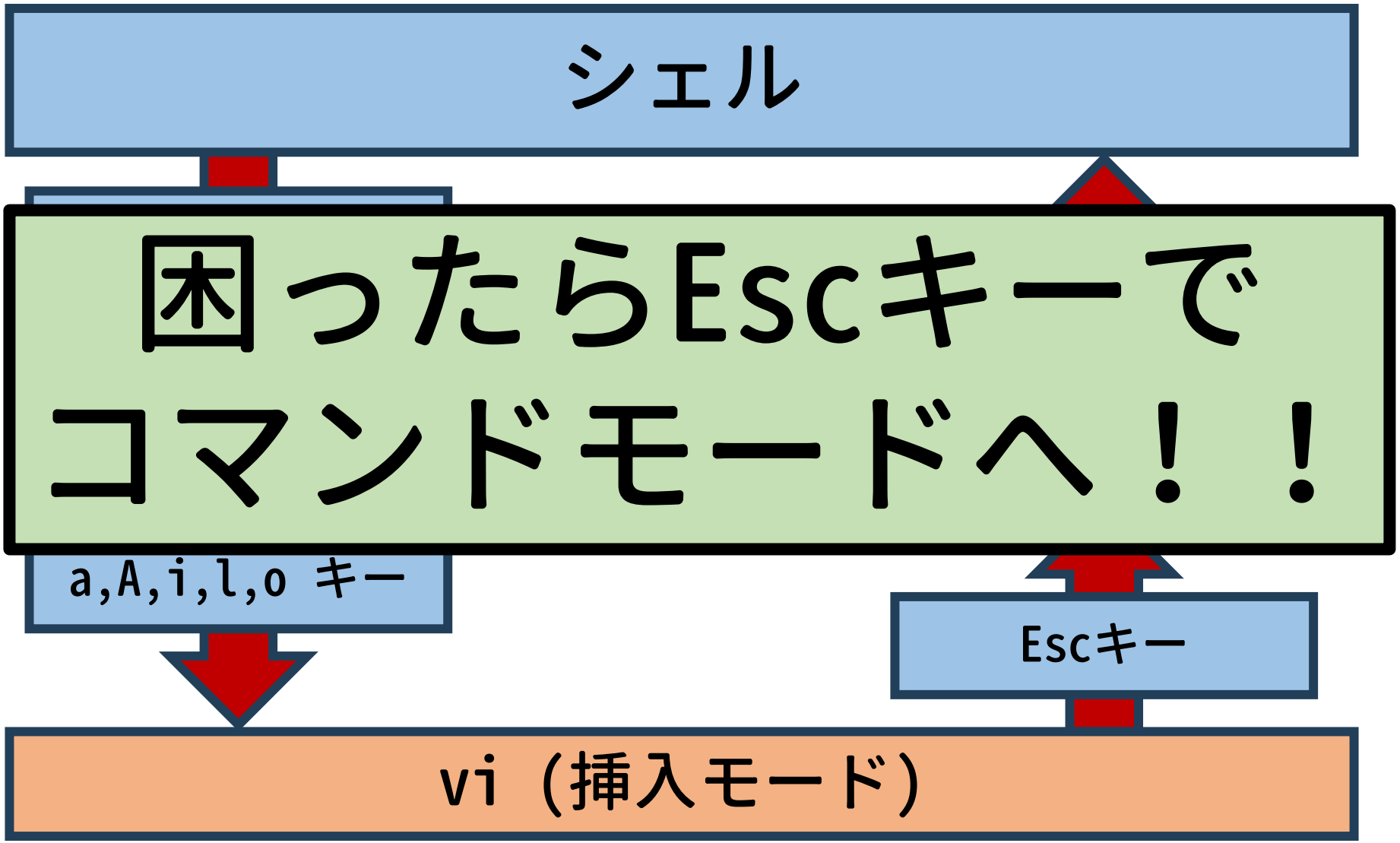
特徴

- 動作が軽快
- どの Linux でもほぼ確実にインストールされている
  - トラブル時に役立つ → 管理者にとって必修のエディタ
- 操作方法がかなり独特で，慣れが必要

viの操作概略



viの操作概略



後半のまとめ

シェルスクリプト

- コマンドを実行順に並べて記述したファイル
- 制御構造を利用したプログラミングが可能

テキストエディタ

- テキストファイルを編集するためのソフトウェア
- 通常の文書，プログラム，設定ファイルの作成/編集が可能

vi

- Linux に標準的にインストールされているテキストエディタ
- 動作が軽快で，いざという時に必須となるツール
- コマンドモードと挿入モードがある

後半の実習では…

viを使えるようにする

- 最低限のテキスト編集技術を身につけよう！

シェルスクリプトを書いてみる

- 煩雑な作業をスクリプトを書くことで効率化しよう！

参考文献

- INEX2024 最低限UNIX(Linux) [II]
  http://www.ep.sci.hokudai.ac.jp/~inex/y2024/0510/
- 魚田勝臣 著，共立出版，コンピュータ概論 第7版，2017年
- 三宅英明・大角祐介，新しいLinuxの教科書，SB Creative，2015年
- Arnold Robbins & Nelson H.F. Beebe 著，日向 あおい 訳，詳解シェルスクリプト，オライリー・ジャパン，2019年
- 山口和紀・古瀬一隆 監修，新 The UNIX Super Text 上 改訂増補版，技術評論社，2003年
- 山口和紀・古瀬一隆 監修，新 The UNIX Super Text 下 改訂増補版，技術評論社，2003年
- Cameron Newham & Bill Rosenblatt 著，株式会社クイープ 訳，入門 Bash 第3版，オライリー・ジャパン，2022年
- Abraham Silberschatz，Peter Baer Galvin，Greg Gagne 著，土居 範久 監訳，オペレーティングシステムの概念，共立出版，2010年