

情報実験・第4回 (2020/06/12)

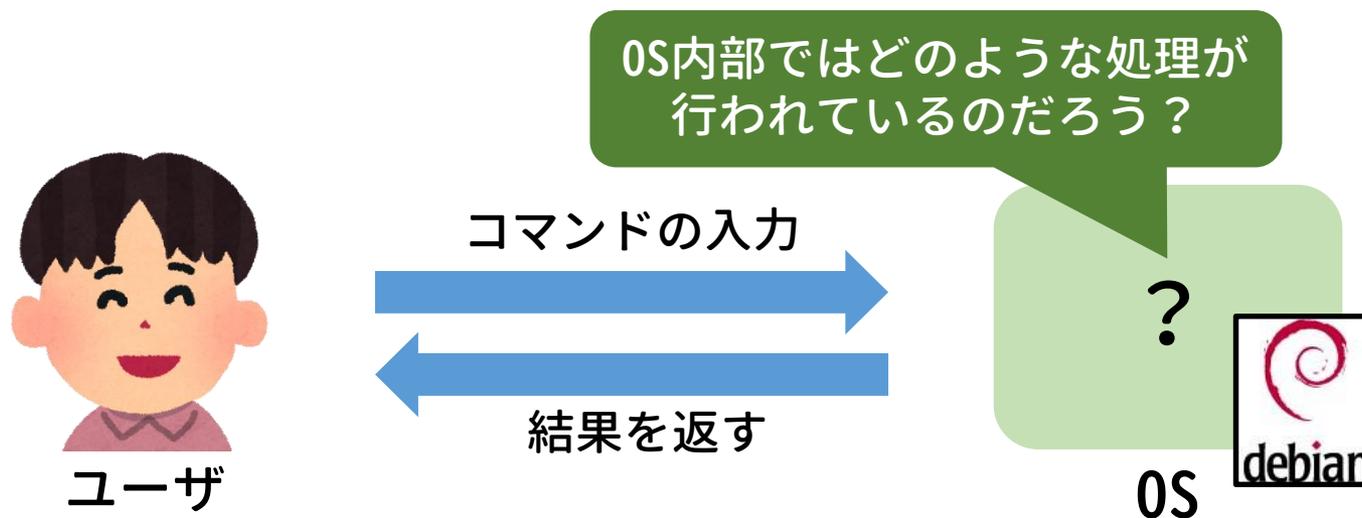
最低限Unix(Linux) II ～シェル・テキストエディタ～

北海道大学大学院 理学院 宇宙理学専攻
修士課程 1年
茂木 遥平 / Moteki Yohei



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

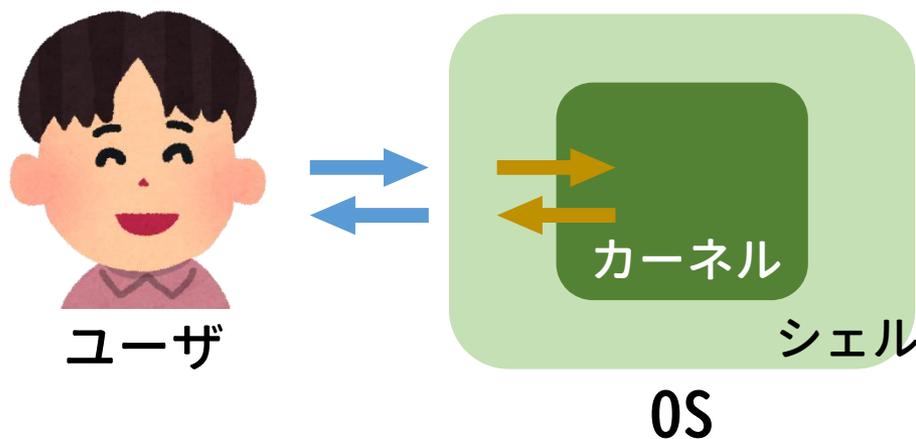
復習：OS(Operating System)

■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

■OSの大まかな構造

カーネル(Kernel:核)とシェル(Shell:殻)の二段階構造。



目次

1. カーネルとシェル
2. シェルスクリプト
3. テキストエディタ

1. カーネルとシェル

カーネル

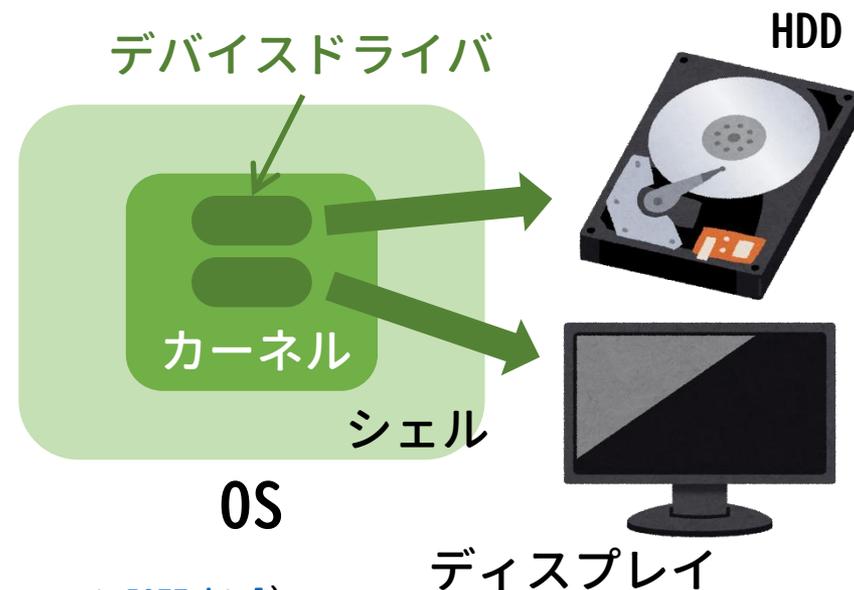
カーネル (Kernel, 核)

- OSの中核をなすソフトウェア。
- ソフトウェアの要求に対して、デバイスドライバを介して (第六回参照) 必要なハードウェアを制御する
e.g. CPUに計算を実行させる, HDDにデータを保存する

ユーザはカーネルを直接操作できない。

→ ユーザとカーネルを仲介するソフトウェアが必要。

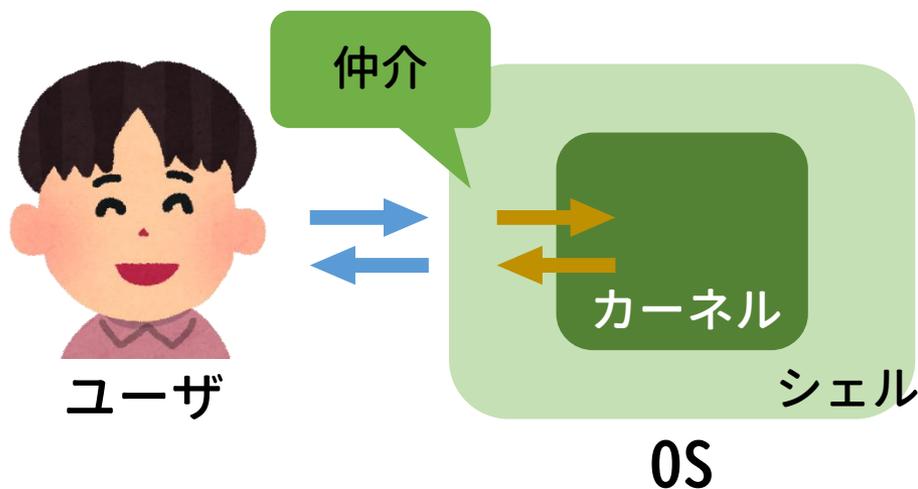
↓
シェル



シェル

シェル (Shell, 殻)

- ユーザとカーネルを仲介するソフトウェア。
- ユーザはシェルを通して計算機に作業を要求する。
- ユーザインタフェース (UI, ユーザとの境界部分) を実装。





ユーザインタフェースの例：GUI

0715_lec3.pptx

サインイン

ファイル ホーム 挿入 デザイン 画面切替 アニメーション スライド 校閲 表示 アドイン ストーリー Iguana 操作アシスト

新しいスライド

クリップボード スライド フォント 段落

自動保存

1 2 3 4 5 6 7 8 9

カーネル

カーネル (Kernel, 核)

- OSの中核をなすソフトウェア。
- ソフトウェアの要求に対して、デバイスドライバを呼び出す。

(第五回参照)

e.g. CPUに計算

ユーザはカーネルを操作できない。

→ ユーザとカーネルを分離するソフトウェア

ノートを入力

日本語

66%

ファイル(E) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

0 - 1703.06718.pdf ja

https://www.jstage.jst.go.jp/article/geoj/123/9/733-745/0

733 (1 / 13) 自動ズーム

地質学雑誌 第123巻 第9号 733-745ページ, 2017年9月
Jour. Geol. Soc. Japan, Vol. 123, No. 9, p. 733-745, September 2017
 doi: 10.5575/geosoc.2017.0034

総説

変成岩組織と鉱物組成累帯構造からの情報抽出：フォワードモデルと逆解析
 Information extraction from metamorphic rock textures and compositional zoning of minerals: Forward models and inversion analyses

岡本 敦*1 桑谷 立*2, *3

Sushi Okamoto*1 and Tetsu Kuvvatani*2, *3

17年4月15日受付
 17年6月9日受理
 東北大学大学院環境科学研究科
 Graduate School of Environmental Studies,
 Tohoku University, 6-6-20, Aramaki, Sendai,
 980-8579, Japan
 国立研究開発法人海洋研究開発機構
 Japan Agency for Marine-Science and Technology,
 2-15, Natsushima-cho, Yokosuka,
 Kanagawa, 237-0061, Japan
 国立研究開発法人科学技術振興機構
 PRESTO, Japan Science and Technology
 Agency (JST), 4-1-8 Honcho, Kawaguchi
 332-0012, Japan

Corresponding author: A. Okamoto,
 okamoto@mail.kankyotohoku.ac.jp

はじめに

変成岩には、地球内部の物理化学的な環境と岩石形成のプロセスに関する様々な情報が記録されている。変成岩研究の大きな主題の一つは、岩石の経験した温度圧力を推定することであり、多成分多相系である複雑な岩石に相平衡熱力学を適用することで大きな成功を取ってきた。特に、1990年

などの地球内部のダイナミックな現象の理解に大きな役割を果たしてきた(Kerrick and Connolly, 2001; Hacker et al., 2003 など)。

一方、変成岩をよく観察すると、鉱物が部分的に置換された仮像組織、鉱物粒子の化学組成が変化する累帯構造、岩相境界に発達する反応帯など、露頭スケールから薄片、鉱物粒子スケールまでさまざまな空間的不均質性が観察される。すなわち、変成岩には「多相系」だけでなく「不均質系」

Abstract

Metamorphism refers to the reactions that proceed in rocks in response to the dynamic environmental factors of the Earth's interior, which include temperature, pressure, and chemical composition. Because it is difficult to obtain time-series data of metamorphic processes directly, it is necessary to extract information from the spatial patterns of the final states of the rocks (e.g., the textures of metamorphic rocks) to understand the conditions and processes of metamorphism. The complications of metamorphic rock textures range widely, from simple problems that can be modeled based on rigid theoretical backgrounds, to complex problems in which several processes interact in nonlinear ways, and forward models themselves are still being developed based on combinations of empirical laws. In this paper, we review recent progress in the analysis and modeling of metamorphic rock textures, with a particular focus on the thermodynamic analysis of zoned minerals and forward modeling of reaction-induced fracturing, employing the distinct element method (DEM). We show that stochastic inversion analyses based on Bayesian inference can be a powerful tool for solving various petrological problems characterized by parameters with undefined values and noise. By effectively using algorithms of machine-learning, the approach of data assimilation, which combines numerical simulations and observed data, is likely to yield a breakthrough in terms of deciphering the complex textures of metamorphic rocks.

Keywords: textures of metamorphic rocks, forward modeling, inversion analyses, compositional zoning, reaction-induced fracturing

ユーザインタフェースの例：CUI

```
Debian GNU/Linux 10 Buster tty1
```

```
johoXX login: ymoteki
```

```
Password:
```

```
ymoteki@johoXX: ~$ ls -F
```

```
190322_Arc/  190422_Arc/  IMG_1377.JPG  dthesis.bak/
```

```
ymoteki@johoXX: ~$
```

GUIとCUIの特徴

GUIの特徴

- 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（CUIよりも計算機の動作が複雑）

CUIの特徴

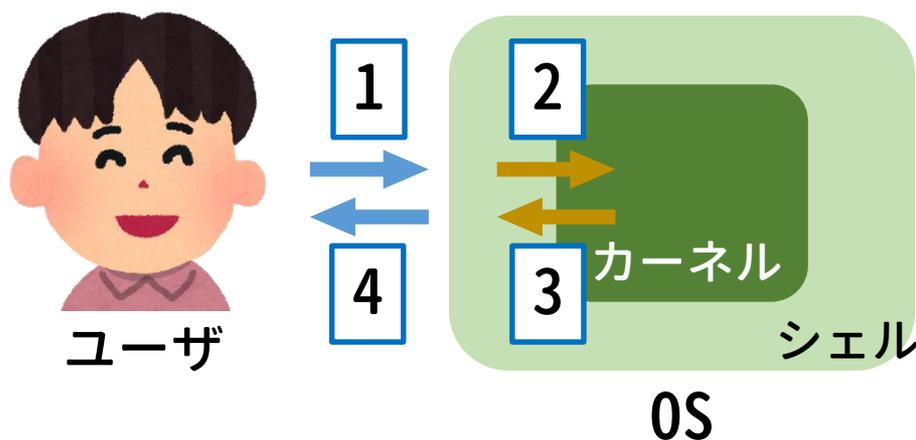
- コマンドを覚えればキーボードだけで何でもできる
- 計算機への負荷が小さい
 - サーバ業務，トラブル対処に強い
- 単純な繰り返し作業に向く（本日の課題にも関連）

シェルの基本的機能

- コマンドインタプリタ (Command Interpreter)
- 環境設定

コマンドインタプリタ

1. UIを通してユーザのコマンドを受け取る
2. 適切なアプリケーションソフトウェアに引き渡す
3. アプリケーションソフトウェアから実行結果を受け取る
4. 実行結果をUIを通してユーザに返す



コマンドインタプリタの仕事



ユーザ

```
hoge@johoXX: ~$ █
```

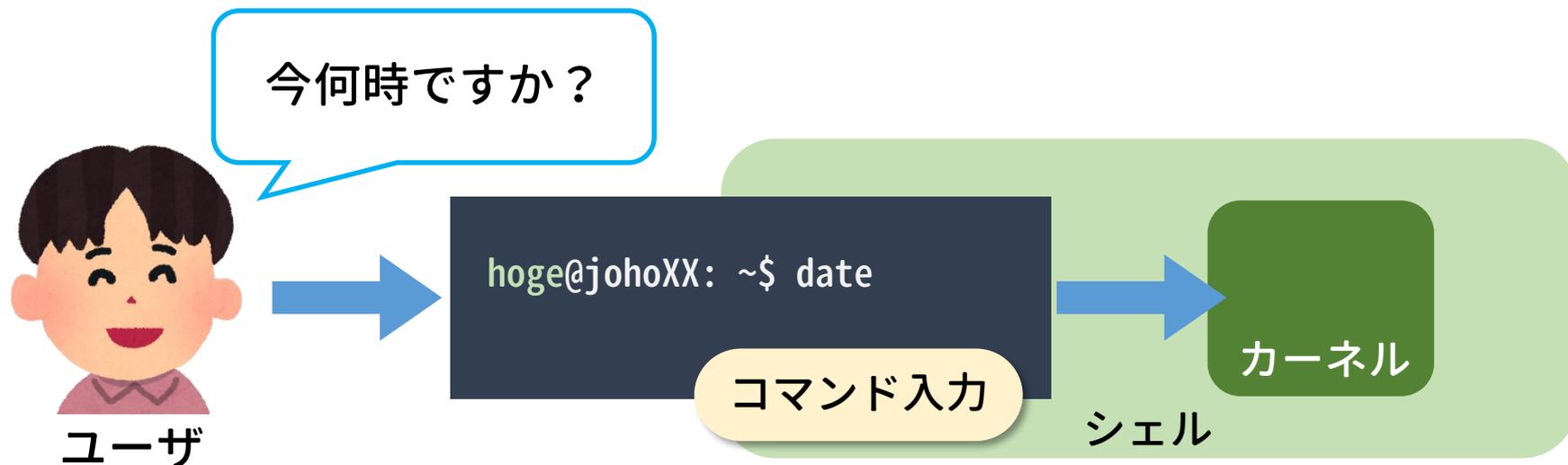
プロンプト
(待ち受け表示)

カーネル

シェル

コマンドを待ち受ける (これも仕事の一つ)

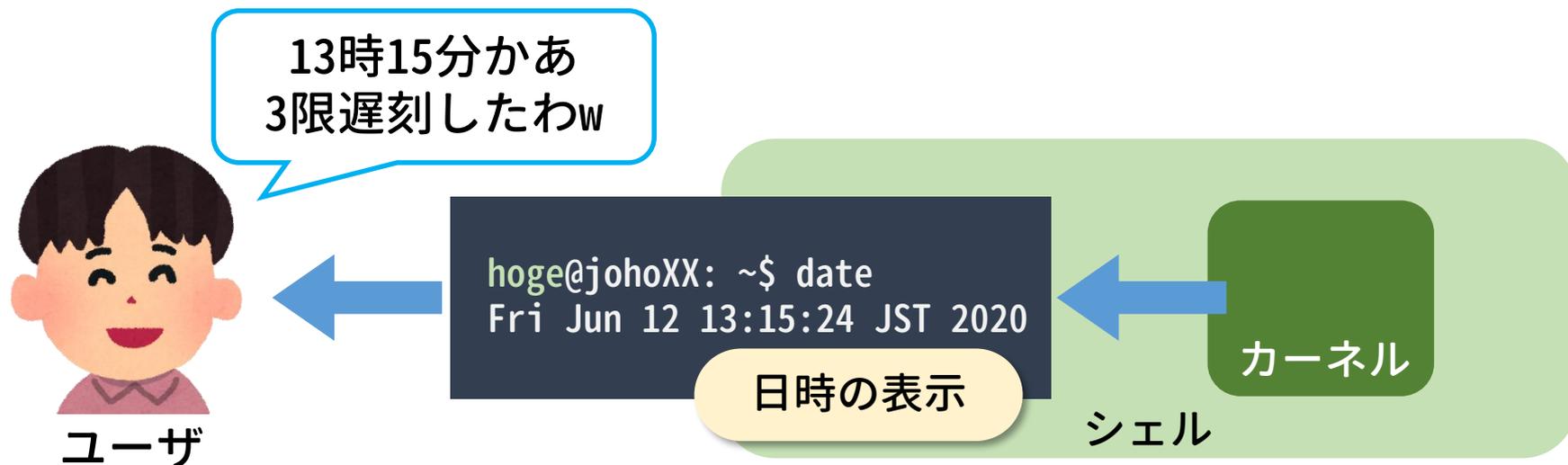
コマンドインタプリタの仕事



ここでのシェルの機能

- 文字列 `date` を受け取る
- `date` というコマンドを探し出し、アプリケーション（カーネル）に実行を依頼

コマンドインタプリタの仕事



ここでのシェルの機能

- アプリケーション（カーネル）から結果を受け取る
- 結果を UI に表示する

シェルの基本的機能

- コマンドインタプリタ (Command Interpreter)
- 環境設定

環境設定

環境

- アプリケーションソフトウェア間で共有される設定情報
(e.g. 言語)
- 各アプリケーションソフトウェアはシェルから与えられた環境下で動く

環境変数

- 設定内容を格納する変数
変数 `LC_ALL` に代入されている値の例：`ja_JP.UTF-8`
- 起動時に自動設定されるが，手動で書き換えることもできる
例：`export LC_ALL=ja_JP.UTF-8`

環境設定

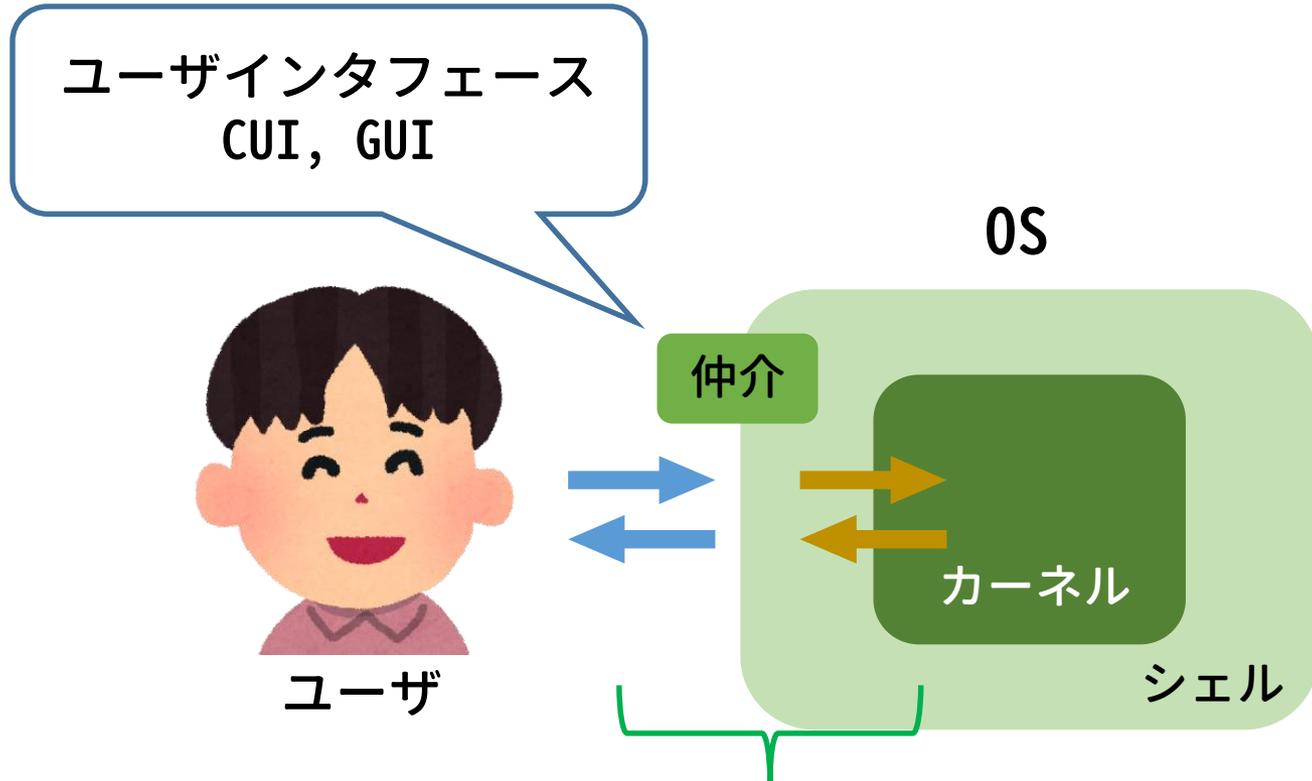
```
ymoteki@joho27:~$ date  
Fri Jun 12 17:48:46 JST 2020  
ymoteki@joho24:~$
```

```
ymoteki@joho27:~$ date  
Fri Jun 12 17:48:46 JST 2020  
ymoteki@joho27:~$ export LC_ALL=ja_JP.UTF-8  
ymoteki@joho27:~$
```

```
ymoteki@joho27:~$ date  
Fri Jun 12 17:48:46 JST 2020  
ymoteki@joho27:~$ export LC_ALL=ja_JP.UTF-8  
ymoteki@joho27:~$ date  
2020年 6月 12日 金曜日 17:49:38 JST  
ymoteki@joho27:~$
```



シェルの基本的機能のおさらい



- シェルの基本的な設定
- コマンドインタプリタ
 - 環境設定

シェルの種類

多様なシェルが存在

sh, bash, csh, dash, tcsh, zsh ...

INEX では主に **bash** を使う

色々便利な機能を持った標準的なシェル

ヒストリ機能, 補完機能,
リダイレクト...

シェルの種類

多様なシェルが存在

sh, bash, csh, dash, tcsh, zsh ...

INEX では主に **bash** を使う

色々便利な機能を持った標準的なシェル

ヒストリ機能, 補完機能,
リダイレクト...

詳しくは実技編で！！

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

利用する意義

- 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

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

- バックアップを取るスクリプトを作る
- 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

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

```
#!/bin/bash  
  
NICHI=`date '+%Y-%m-%d'`  
  
cp source.txt backup_${NICHI}.txt
```

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=`expr $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
- 通常の記事からプログラム、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, emacs, nano, メモ帳, Terapad, 秀丸エディタ, …)

vi ～困ったときに頼れるアイツ～

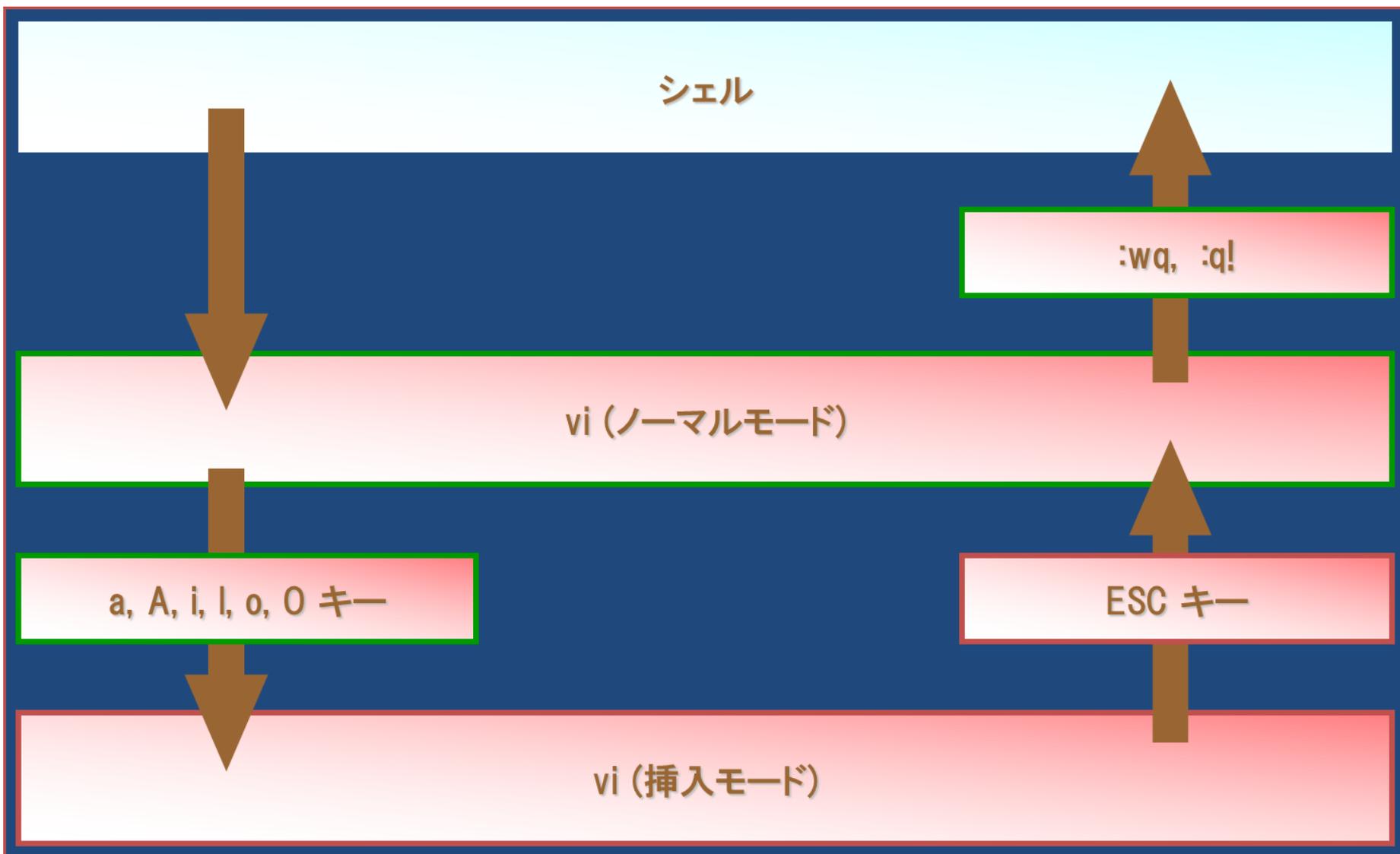
テキストエディタの一つ

Unix黎明期から使われている由緒正しいエディタ

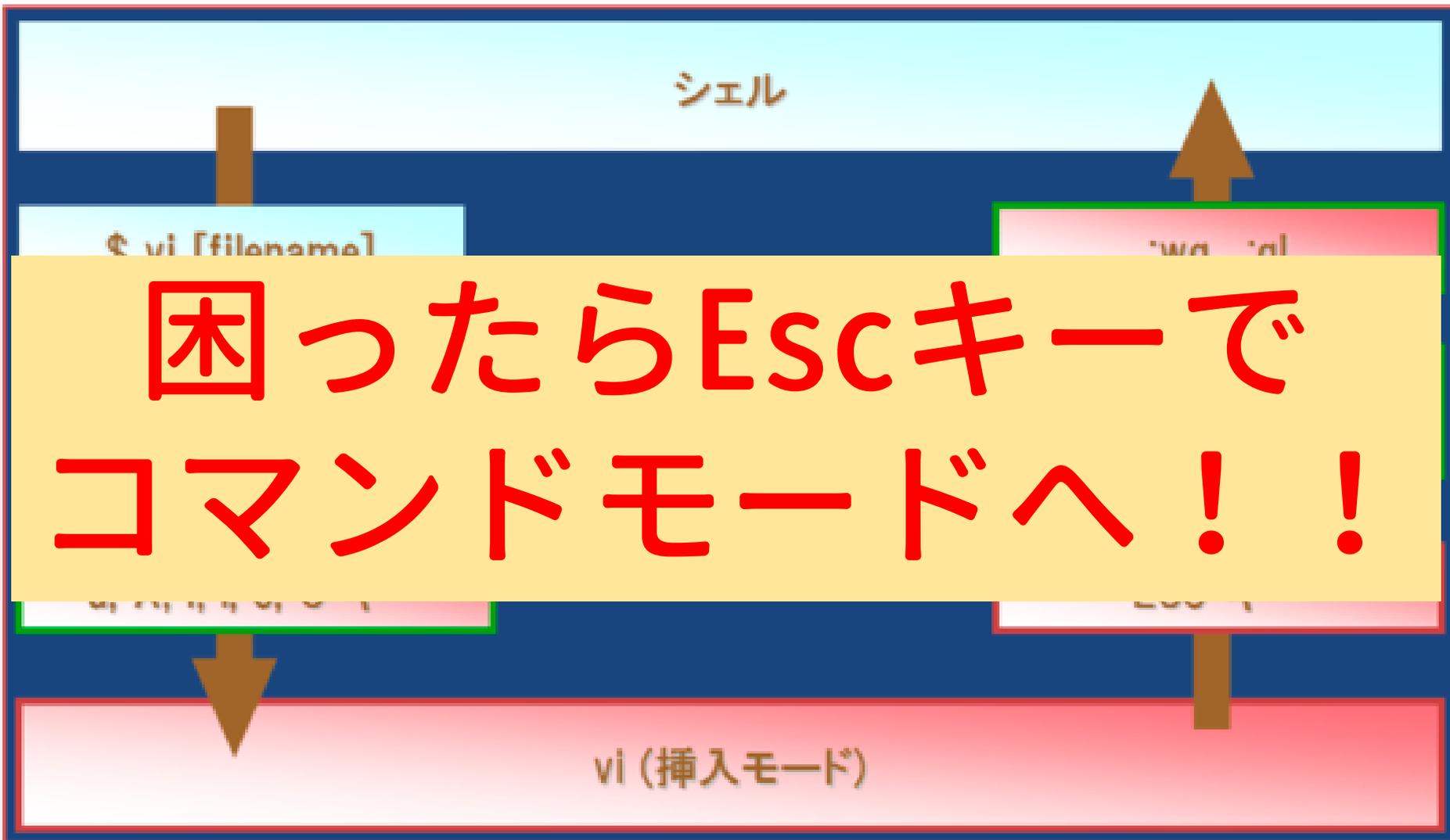
特徴

- 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



まとめ

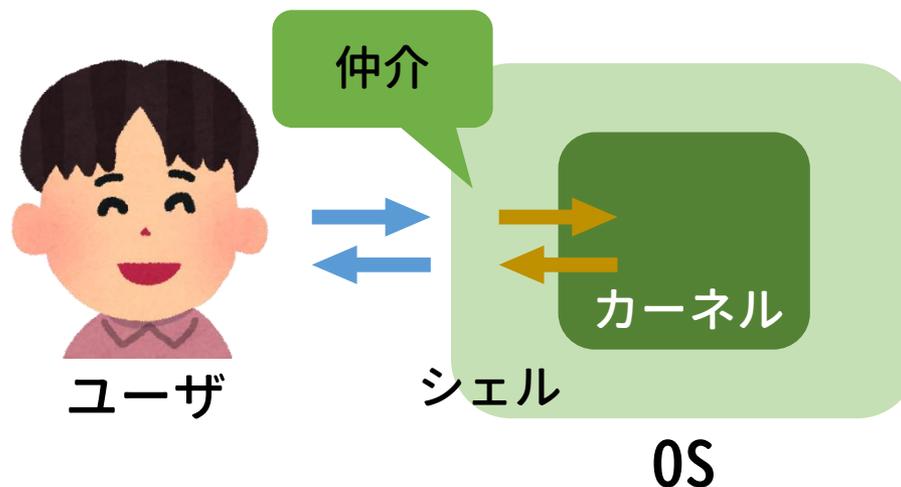
まとめ

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

- ユーザとカーネルを仲介するソフトウェア
- ユーザはシェルを通して計算機に作業を要求



まとめ

シェルの仲介者としての役割

- ユーザインタフェースの提供（ユーザと接する境界）

シェルの基本的機能

- コマンドインタプリタの提供
- 環境を設定可能
 - 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

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

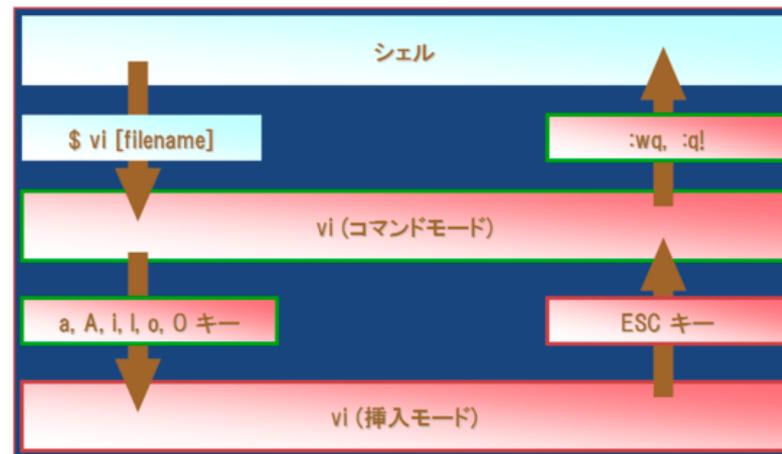
まとめ

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



参考文献

- INEX2019 最低限UNIX/Linux [II]
<http://www.ep.sci.hokudai.ac.jp/~inex/y2019/0426>
- INEX2018 最低限UNIX/Linux [II]
<http://www.ep.sci.hokudai.ac.jp/~inex/y2018/0427>
- INEX2017 最低限UNIX/Linux [II]
<http://www.ep.sci.hokudai.ac.jp/~inex/y2017/0428>
- INEX2016 最低限UNIX/Linux [II]
<http://www.ep.sci.hokudai.ac.jp/~inex/y2016/0506>
- INEX2019 最低限UNIX/Linux [I]
<http://www.ep.sci.hokudai.ac.jp/~inex/y2019/0419>
- IT用語辞典 e-words
<http://e-words.jp>
- 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

本日の実習では…

シェルに慣れる

- シェル（特にbash）の各種機能を試してみよう！

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！