OSの構造とシェル

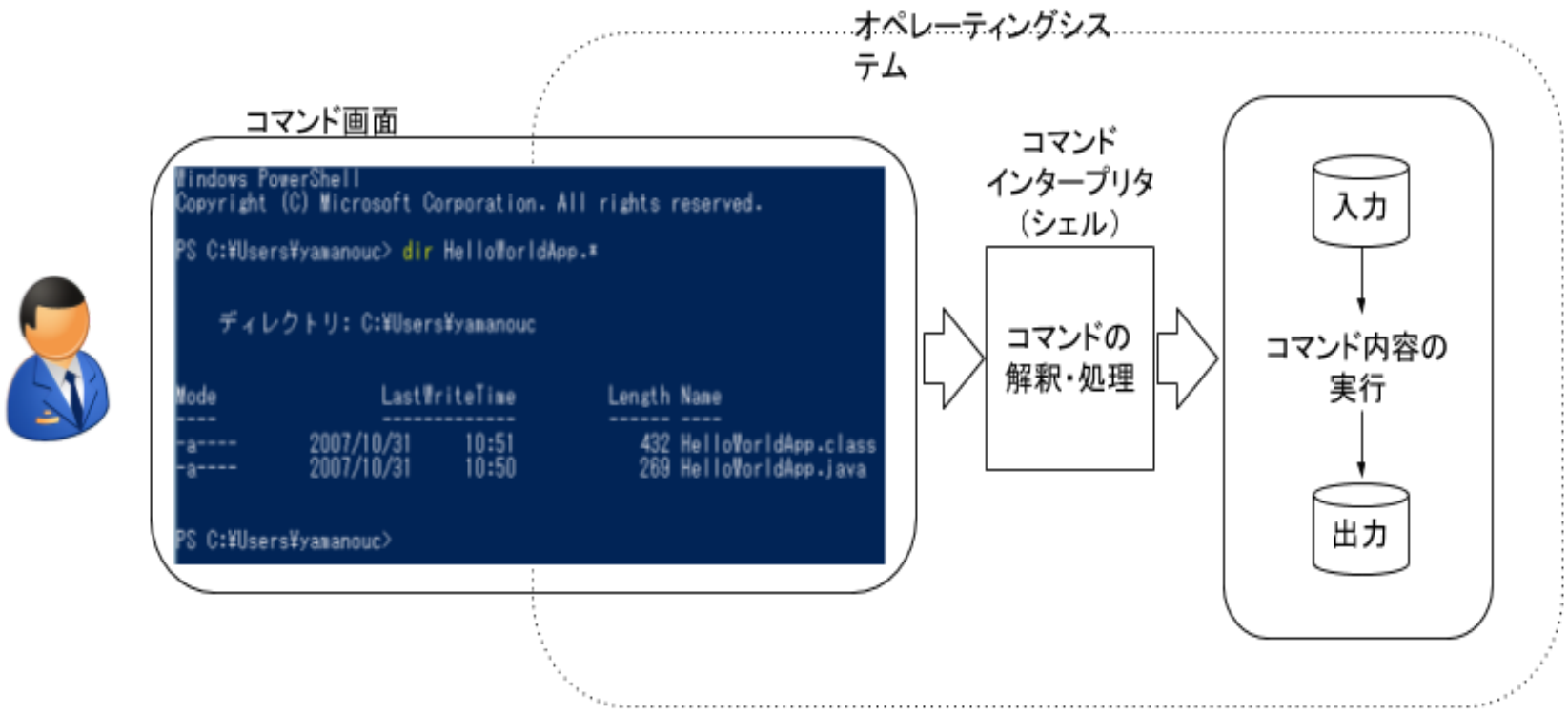
スケジュール

- 6/10 RNA-seqやVariant Callの流れ：意識合わせ
- 6/17 受け渡すデータ形式／OSの知識とシェルコマンド操作
- 7/01 OSの知識とファイル操作・プログラムインストール
- (　) RNA-seq処理を試してみる（時間？）
- (　) Pythonプログラミングの基礎
- (　) Pythonライブラリのいろいろ

2019-06-17　　山内 長承

yamanouc@is.sci.toho-u.ac.jp

OSとシェル（コマンドインタープリタ）



- マウスで選択　vs　コマンド（文字列）で指示
  - 同じことをしている　　使い勝手が違う
  - マウス：とっつきやすい
    コマンド：繰り返しなどが容易　←　プログラム的

脱線：　MacOS　vs　Linux　vs　Windows

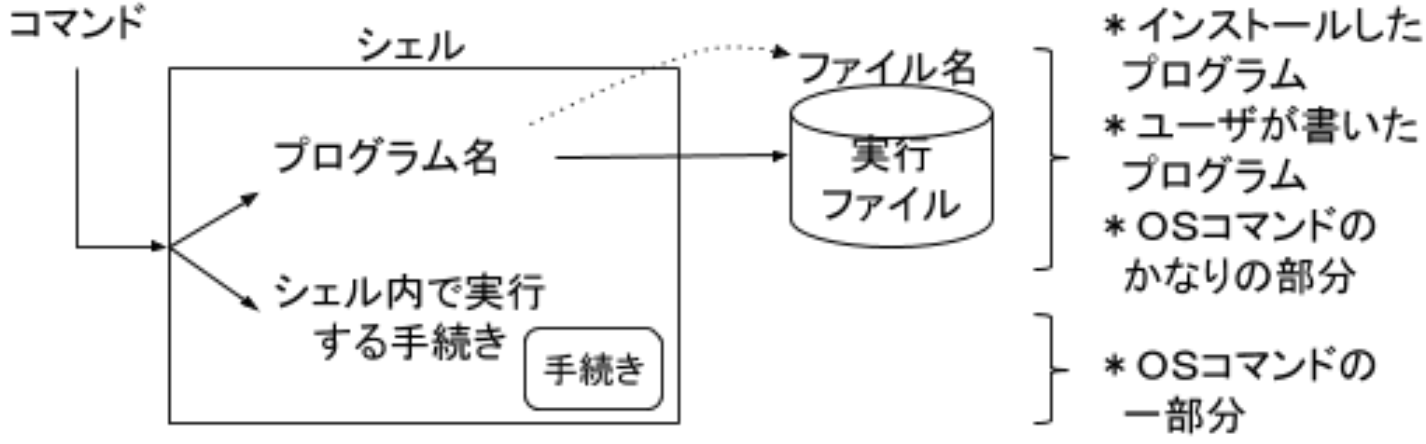
全体の構造は、どれも同じような形

- OS基本部分(カーネル) + OSのサービス部分　（+ アプリ）

OSカーネル： LinuxとMacOSは同系統、Windowsは全く別

サービス部分： LinuxとMacOSはコマンドはほぼ同系統で
GUI部分(ウィンドウ)が違う
Windowsはかなり違う

コマンドはどう実行されるか



- プログラム名　　たとえば　fastqc
  - 厳密にはプログラムのファイル名
- OSのコマンドだと思っているが実はプログラム名　例 ls
- シェルに作り付けのOSコマンド　例 pop, which ... 少ない

MacOS/Linuxにはどんなosコマンドがあるか

使い慣れるしかない！

- ファイル操作
  - ls:　リスト　ファイル一覧　ls *.py　'*'：ワイルドカード
  - rm:　消去　　remove
  - cat:　ファイルを結合したり表示したり
  - less/more:　文字ファイルの内容を閲覧
- ジョブ・プロセス操作
  - fg, bg:　プロセスをフォア/バックグラウンドに移す
  - Ctrl-Z:　プロセスを中断しシェルに戻る（bgとか可能）
  - kill -K プロセス番号:　プロセスを終了する
  - ps -aef :　プロセス一覧
  - Ctrl-C:　フォアグラウンドのプロセスを異常終了
  - ... ...

標準入出力とパイプ

プログラムの入力と出力の指定の仕方の１つ

　　プログラムにそのように書いておく/書いてある

指定しなければ、標準入力は端末入力、標準出力は端末出力

　　指定は起動時にコマンドを書く時に '<'、'>'

```
ls *.fasta > filelist.txt
```

```
cat < in.txt > out.txt
```

２つのプログラムPAとPBをパイプで繋ぐ

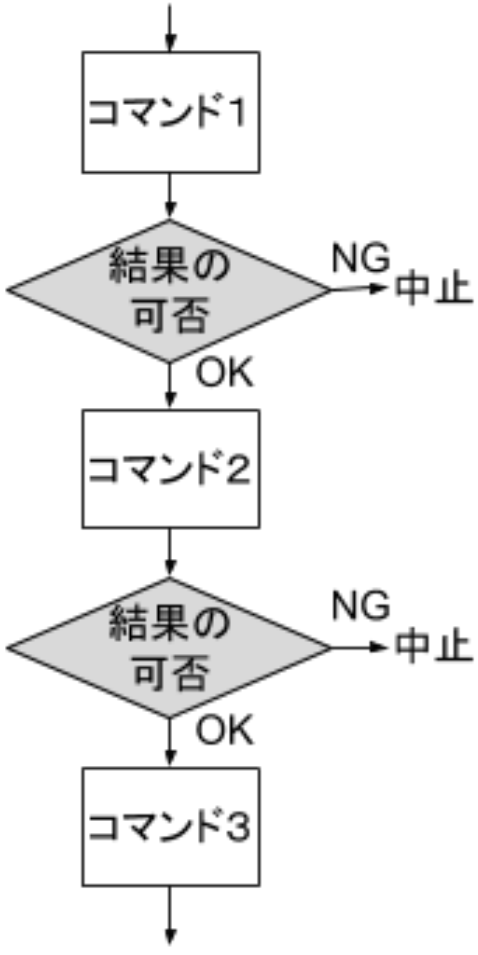
　　PAの標準出力を、PBの標準入力として、PAとPBを起動

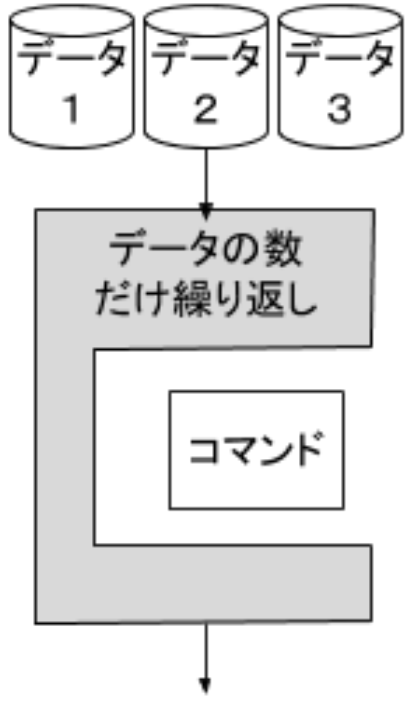
```
PA | PB
```

　　多数段をつなぐことができる。各段は並行して実行する

```
処理して結果生成 | 特定文字を含む行を抽出 | 列を抽出
```

シェルプログラミング





条件分岐や繰り返しが可能