UNIX 入門

基礎生物学研究所
ゲノムインフォマティクストレーニングコース

2015

西出 浩世

hiroyo@nibb.ac.jp

UNIX を使う理由

- UNIX でしか使えないアプリケーション
  - 最新の研究用ソフト、並列化や巨大メモリに対応したソフト
- たくさんの処理を一度に行なう
  - スクリプトを用いたコマンドの連続実行
- 独自のプログラムを作成
  - Perl, Ruby等のスクリプト言語、豊富な開発ユーティリティ
- WWWサーバやデータベースサーバを立ち上げたい
  - サーバとしての高い安定性、apache や postgres などのフリーウェア

PCでUNIXを使うには

MacOS X	OS自体がUNIX #1	アプリケーション→ユーティリティ→ターミナルでUNIX端末が使用できる
	リモートログイン	UNIXサーバへリモートログイン ターミナルからsshを使用する
Windows	Cygwin	Windows上で動作するUNIXライクな環境
	VMware + Linux	仮想マシンを構築してLinuxそのものをインストールする
	リモートログイン	UNIXサーバへリモートログイン TeraTermからsshを使用する

#1) フリーウェアなどのインストールが必要な場合は「MacOSXでのUNIX環境構築方法」を参照してください

キーボード配置とUNIX環境の確認

_（アンダースコア）

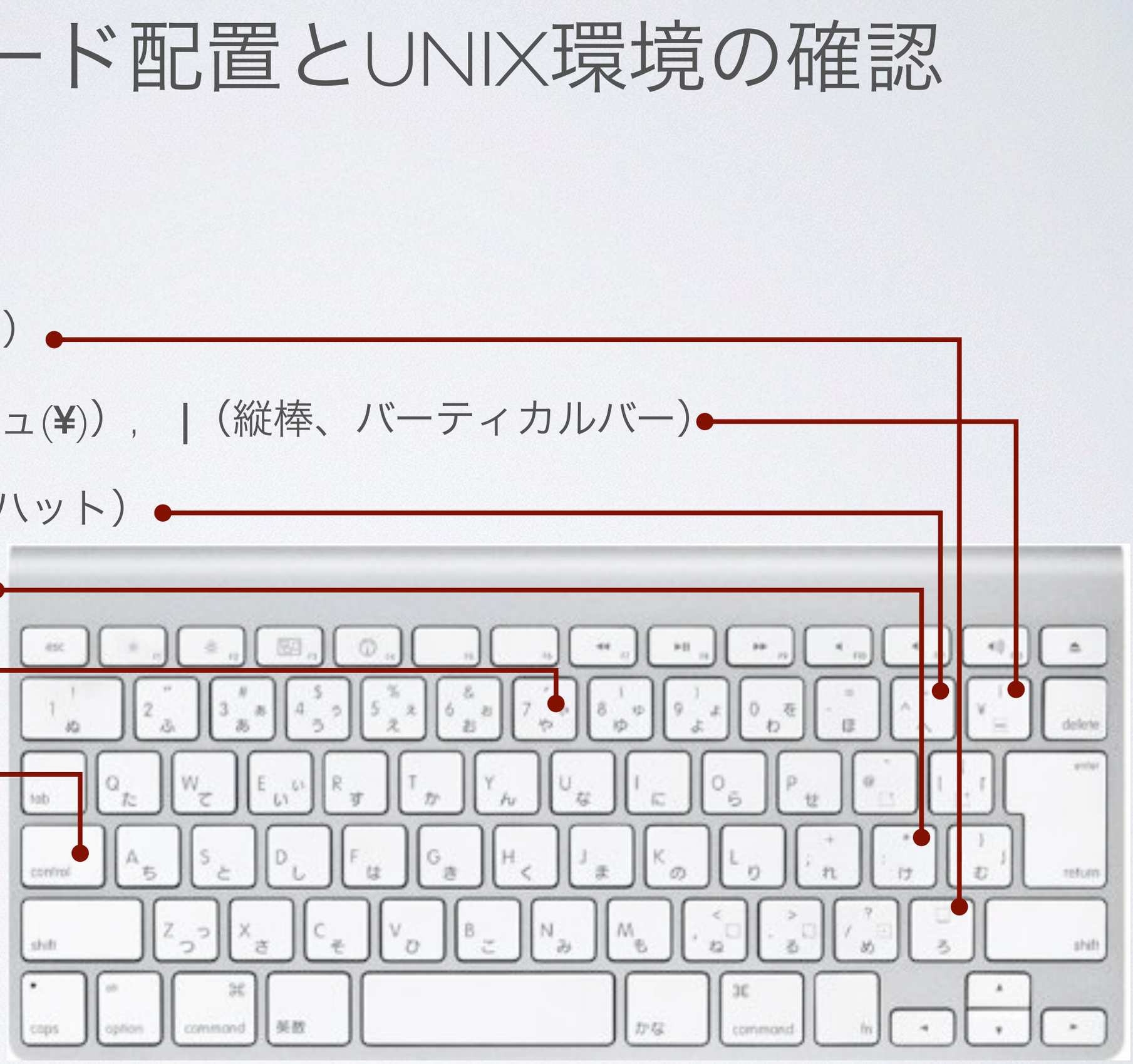
\（バックスラッシュ(¥)），　|（縦棒、バーティカルバー）

~（チルダ），^（ハット）

*（アスタリスク）

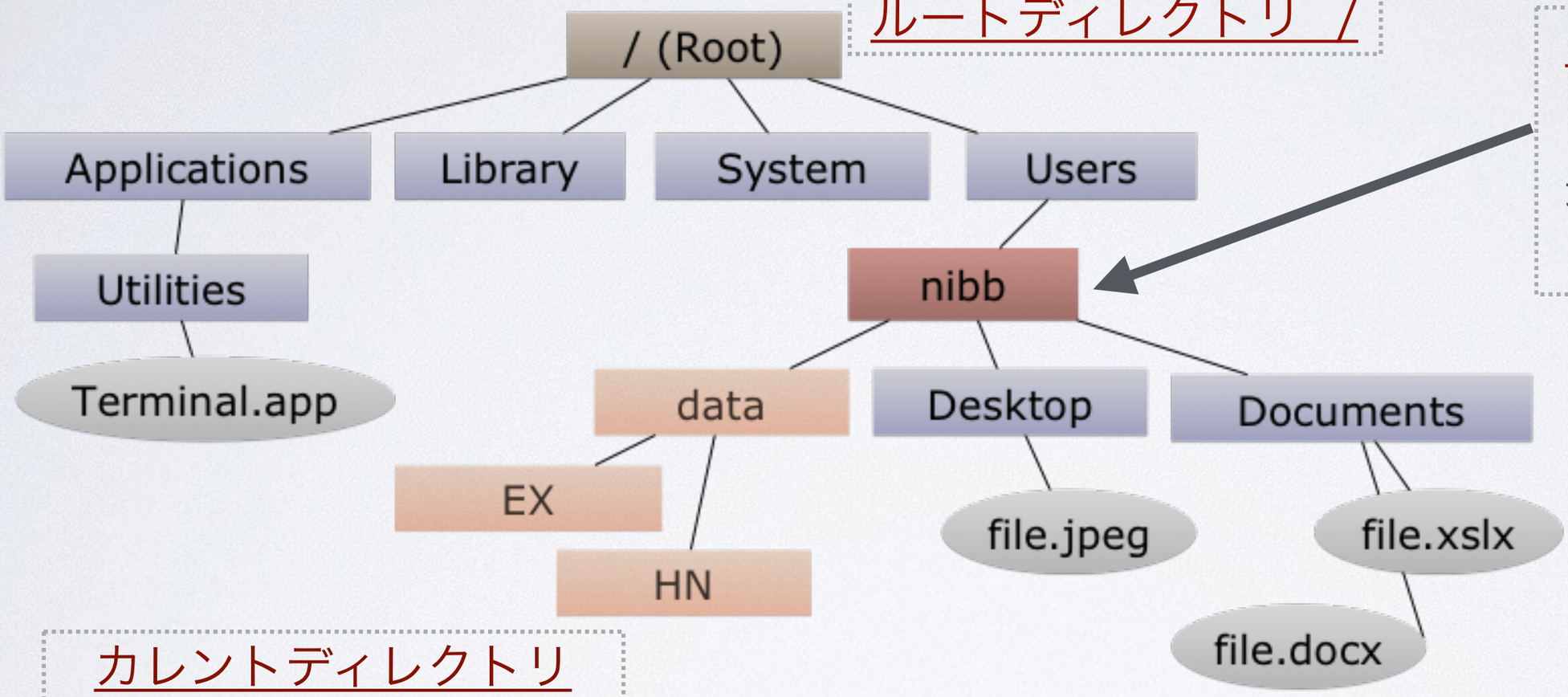
‘（引用符）

Control



UNIXのファイルシステム：階層型ディレクトリ

トップのルートディレクトリ下に、子ディレクトリ、孫(…)ディレクトリがあり、ファイルを配置する



ルートディレクトリ　/

ホームディレクトリ
ログイン時の
最初のディレクトリ
/Users/nibb

カレントディレクトリ
現在の作業ディレクトリ

Finderでの見え方

UNIXにおけるディレクトリの表現

UNIXではディレクトリを上へ下へ移動しつつ作業する

名前	表現方法	説明
ルートディレクトリ	／（スラッシュ）	ファイルシステムの頂点
カレントディレクトリ	．（ドット）	起点とするディレクトリ 今いる場所
親ディレクトリ	..（ドット２つ）	カレントディレクトリの上の ディレクトリ
ホームディレクトリ	～（チルダ）	ユーザ用のディレクトリ ログイン直後にいる場所

プログラム, コマンドが作成するファイルは、
使った時のカレントディレクトリに作成される

ファイル/ディレクトリ名の指定方法

- パス（Path）
  - ファイルやディレクトリを指定する記述
  - UNIXではディレクトリ名をスラッシュ（ / ）で区切る
  - 絶対パスと相対パスの２つの記述法
- 絶対パス
  - ルートディレクトリから目的のファイル・ディレクトリへの道筋の記述
  - 行頭は必ずルートディレクトリ（ / ）となる
  - 例：`/Users/nibb/data/HN/readme.txt`
- 相対パス
  - 現在位置から目的のファイル・ディレクトリへの道筋の記述
  - 例：`data/HN/readme.txt`

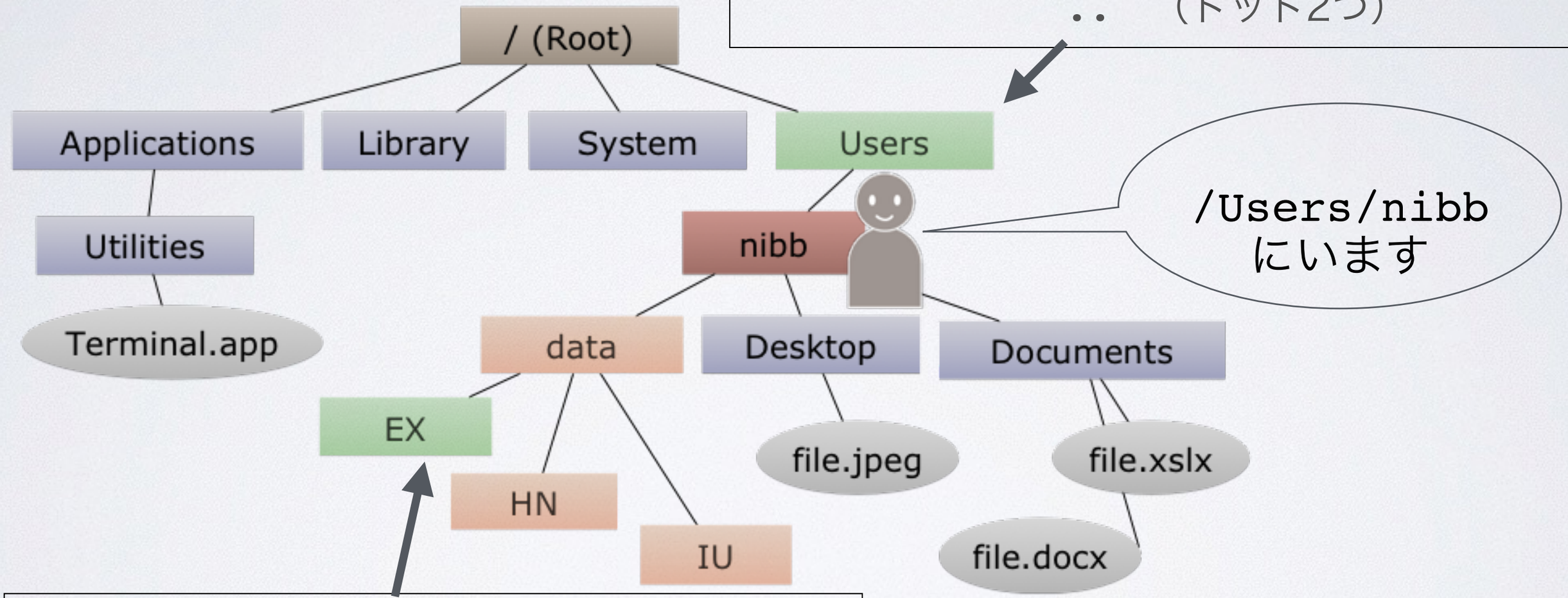
ファイル/ディレクトリ名の指定方法

絶対パス　（ルートディレクトリからのパス）

/Users

相対パス　（カレントディレクトリからのパス）

..　（ドット2つ）



絶対パス　（ルートディレクトリからのパス）

/Users/nibb/data/EX

相対パス　（カレントディレクトリからのパス）

data/EX

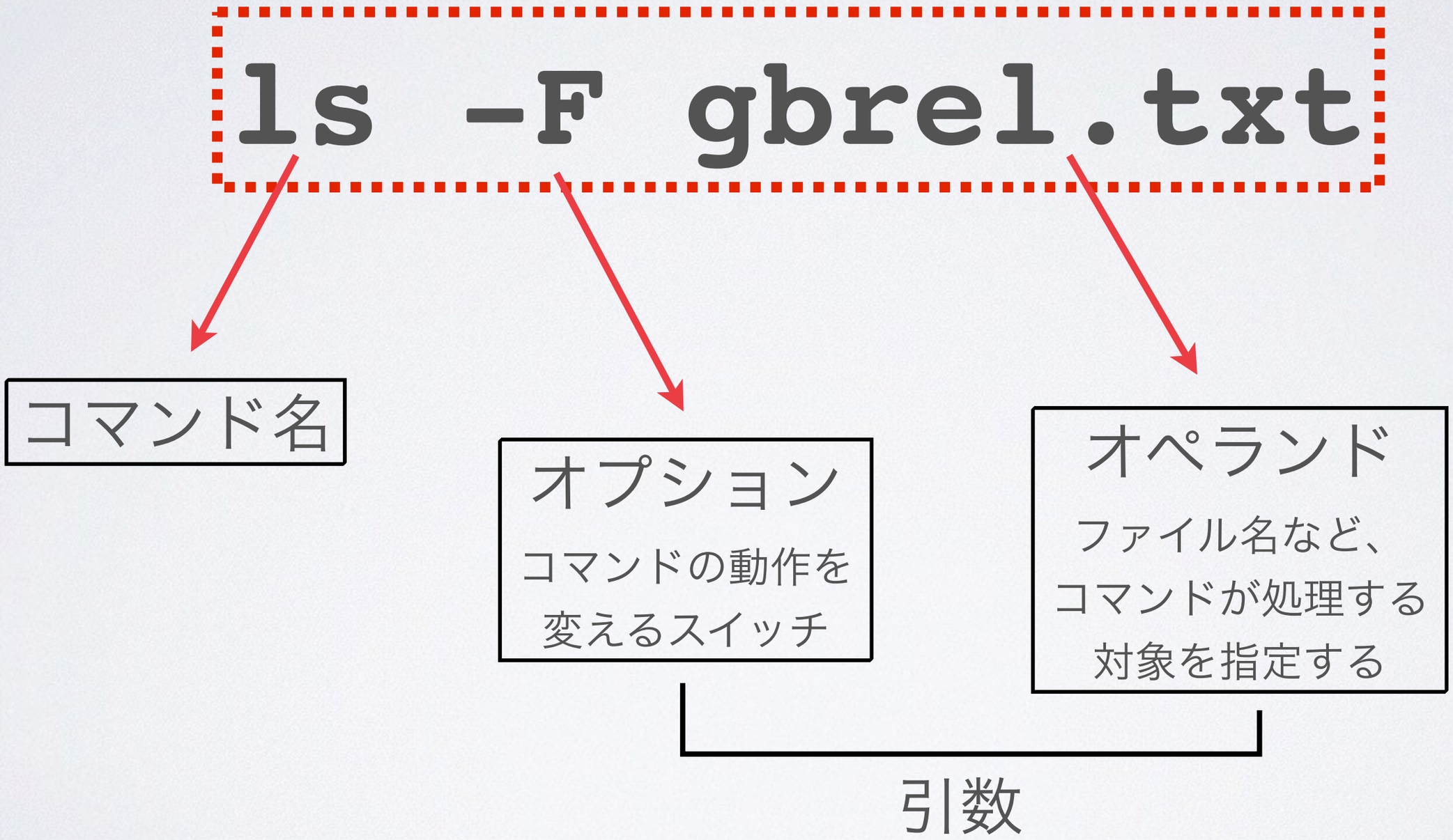
ディレクトリの中身を見る (ls)

- **ls**
  - カレントディレクトリの内容（ファイル名のリスト）を表示
- **ls** ディレクトリ名
  - 指定したディレクトリの内容を表示

```
$ ls data        ディレクトリdataの内容を表示
$ ls /           ルートディレクトリ / の内容を表示
$ ls ..          一つ上のディレクトリの内容表示
$ ls .           ただの ls と同じ
```

- **ls -F**
  - ファイル名の末尾にファイルの種類に応じた記号をつけて表示
    /ディレクトリ、@シンボリックリンク、*実行権付きファイル
- **ls -a**
  - ファイル名の先頭がドット（ .）で始まる隠しファイルも表示する

UNIX コマンドの基本形



補完機能

- コマンド名やファイル/ディレクトリ名の補完
  - ファイル名を途中まで入力して、タブ（Tab）キーを押す

    `$ ls d` と入力してタブキーを押す
  - 一意に決まらない場合は、一意に展開できるところまでを展開する

    `$ ls data/HN/sprot/143` と入力してタブキーを押す。
  - それ以上展開できない場合は、再度（つまり2回）タブキーを押すと、候補ファイルのリストを表示する

    `$ ls data/HN/sprot/1433` の状態で2回続けてタブキーを押す

・入力を著しく効率的にするので、積極的な活用を！
  ・複雑なファイル名の入力ミスを防ぐ
  ・うろ覚えのファイル名でも入力できる
・ファイル名だけでなく、コマンド名でも同様の補完機能が使える

コマンドヒストリと編集

- 過去に実行したコマンドの呼び出しとコマンドの編集
  - Control キー と 同時に p を押す…

キー操作（十字キー代替）	動き
Control + p（↑キー）	1 つ前のコマンド
Control + n（↓キー）	1 つ後のコマンド
Control + b（←キー）	カーソルを左に移動
Control + f（→キー）	カーソルを右に移動
Control + a	カーソルを行頭に移動
Control + e	カーソルを行末に移動

- 引数の影響で長くなってしまったコマンドを再実行する際に便利

ディレクトリを移動する (cd)

- **cd** ディレクトリ名
  - 指定したディレクトリに移動（カレントディレクトリの変更）

```
$ cd data                  dataディレクトリに移動
$ cd ..                    一つ上のディレクトリに移動
$ cd ~/data/HN             ホームディレクトリ以下の/data/HN に移動
```

- **cd**
  - ディレクトリ名を省略すると、ホームディレクトリに移動する
- **pwd**
  - 現在のディレクトリ名の確認

ワイルドカード

- ファイル名やディレクトリ名を指定するときに使う「任意の文字」
- パターンマッチによって複数のファイル名を一度に指定できる。
  - *****　任意の文字列（０文字以上）とマッチ
    - 例）**`ls *.fasta`**　例）**`ls *_HUMAN*`**
  - **?**　任意の１文字とマッチ
    - 例）**`ls 1A2?_HUMAN.fasta`**
  - **[1-5]**　1 から 5までの数字とマッチ
    - 例）**`ls 1A2[1-5]*.fasta`**
  - **{文字列１，文字列２}**　"文字列１"または"文字列２"とマッチ
    - 例）**`ls 1A25_HUMAN.{fasta,phylip}`**

ファイルの内容を一括表示する(cat)

- **cat** ファイル名
  - ファイルの内容を表示する。

```
$ cat 1A25_HUMAN.fasta
```

  - ファイル名を複数指定すると内容を連結して表示する。

```
$ cat *.fasta
```

ファイルの部分表示 (head, tail)

- **head** [-行数] ファイル名
  - ファイルの先頭から指定した行数（指定しないと10行）だけ出力する

```
$ head 1A25_HUMAN.sprot
```
(最初の10行を出力)

  - 大きなファイルの内容をとりあえず確認したい場合などに便利

- **tail** [-行数] ファイル名
  - ファイルの最後から指定した行数を出力する

```
$ tail -20 1A25_HUMAN.sprot
```
(最後の20行を出力)

  - -n +行数 とすると、先頭から数えて指定された行以降を出力する

```
$ tail -n +2 1A25_HUMAN.fasta
```
(先頭行を除去して2行目以降を出力)

ファイルの内容を見る (less)

- **less** ファイル名
  - ファイルの内容をページビューで見る
  - ページの移動には独自のキー操作が必要

キー操作	動き
スペースキー, f	1ページ先へ進む
b	1ページ前に戻る
j	1行先へ進む
k	1行前に戻る
g	先頭行へ移動
G	最終行へ移動
/[パターン]	現在位置から後にパターン検索をして移動
?[パターン]	現在位置から前にパターン検索をして移動
n	検索を再実行して次のヒットに移動
N	逆方向に検索を再実行して次のヒットに移動
q, zz	lessを終了
h, ?	ヘルプを表示する

実習10

ディレクトリの作成と削除 (mkdir, rmdir)

- **mkdir** ディレクトリ名
  - 新規ディレクトリの作成
    - **$ mkdir unixtest**

- **rmdir** ディレクトリ名
  - ディレクトリの削除。
  - ただし、ディレクトリ内にファイルがあると削除できない

ファイルのコピー（cp）

- **cp** file1 file2
  - file1 のコピーを file2 として作成

- **cp** file1 [file2...] dir
  - file1 (, file2, ...) のコピーを、dir 内に同一名のファイルとして作成
  - 同一名ファイルが既に存在しているときは上書きされる

- **cp -r** dir1 dir2
  - dir1をdir2としてディレクトリごとコピーを作成
  - dir2が存在する場合は、dir2以下にdir1として作成

ファイル/ディレクトリ名の変更
ファイル/ディレクトリの移動（mv）

- **mv** file1 file2
  - file1 の名前を file2 に変更する

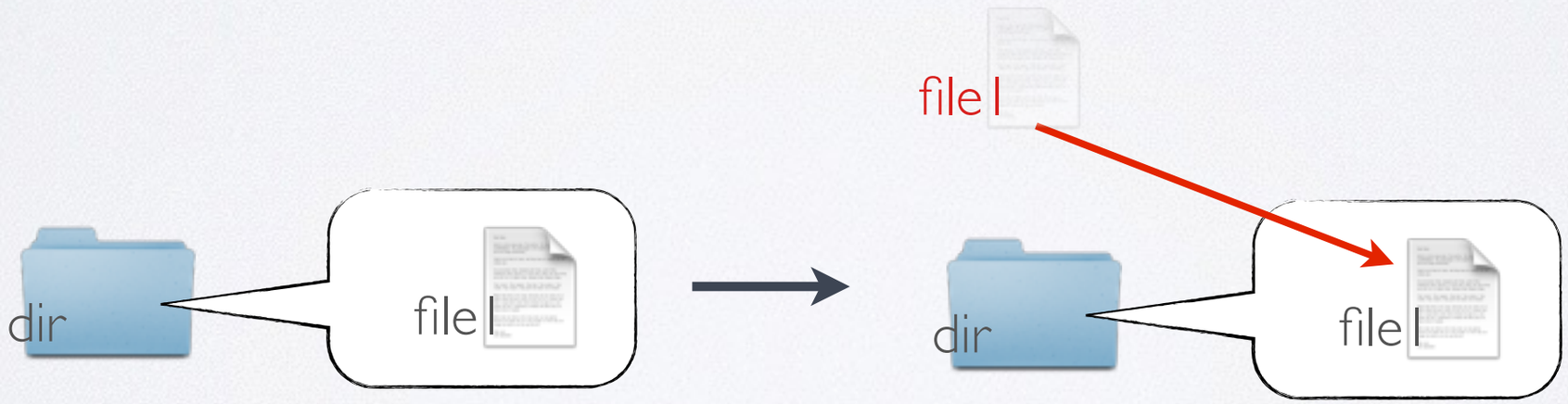
- **mv** file1 [file2...] dir
  - file1 (, file2, ...) を、dir 内に移動する。
  - 同一名ファイルが既に存在しているときは上書きされる

シンボリックリンクの作成（ln -s）

- **ln -s** ファイル/ディレクトリ名 シンボリックリンク名
  - ファイル/ディレクトリ の 「別名」 を作成する。
  - Macのエイリアス、Windowsのショートカットに相当

```
$ ln -s dir/file1 .
```

  - dir 内のfile1のシンボリックリンクをカレントディレクトリ（.）に作成

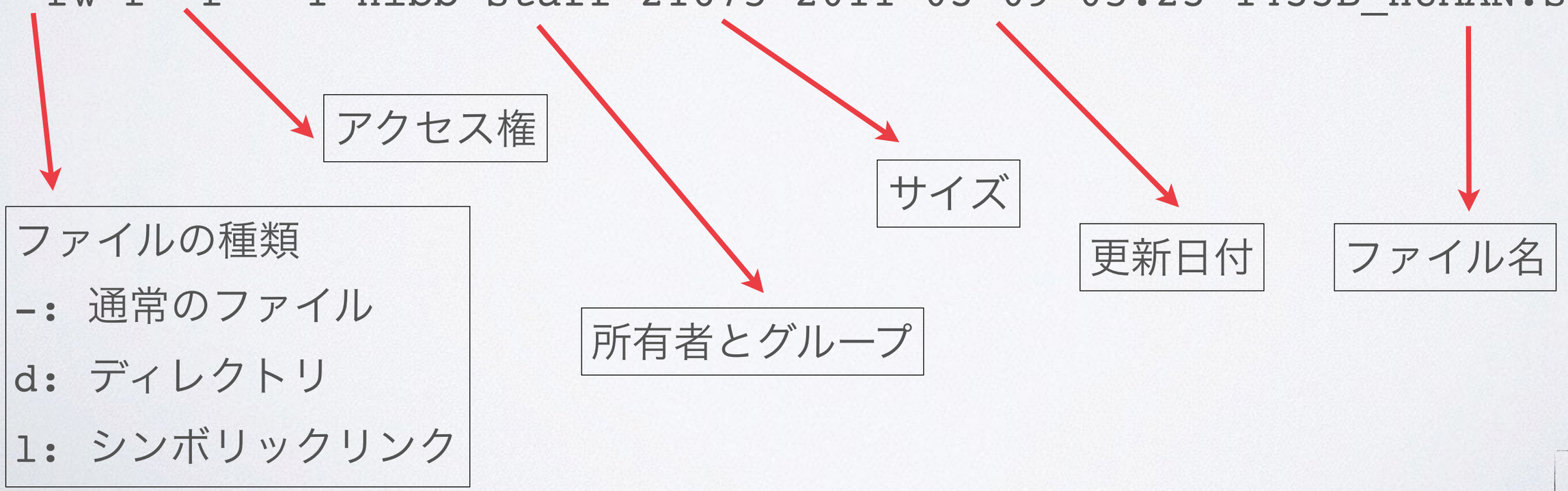


  - オリジナルファイルが移動・消去されると、シンボリックリンクを通した参照はできなくなる。

ファイル情報の表示 (ls -l)

- **ls -l** ファイル/ディレクトリ名
  - ファイルの大きさや最終更新日などの情報のリスト表示（名前順）
- **ls -lt** ファイル/ディレクトリ名
  - 上記と同様だが日付順に表示

```
$ ls -l 1433B_HUMAN.sprot
-rw-r--r-- 1 nibb staff 21675 2011-03-09 05:23 1433B_HUMAN.sprot
```



ファイルのアクセス権（permission）

- ファイル（ディレクトリ）について、どんな人が、何をできるか
  - どんな人？
    - ファイルの所有者 **(u)**
    - 同一グループの利用者 **(g)**
    - その他の利用者 **(o)**
  - それぞれ何ができる？
    - 読み出し **(r)**
    - 書き込み **(w)**
    - 実行 **(x)**
      - ファイルをプログラムとして実行
      - ディレクトリの場合は、そこに移動できる

アクセス権の確認

- **ls –l** ファイル/ディレクトリ名

```
$ ls -l myfile
-rwxr-xr-x 1 nibb cproom 21675 2011-03-09 05:23 myfile
```

–　r w x　r – x　r – x

ファイルの種類
- **-:** 通常のファイル
- **d:** ディレクトリ
- **l:** シンボリックリンク

所有者**(u)**
- **r:** 読み出し
- **w:** 書き込み
- **x:** 実行

同グループの人**(g)**
- **r:** 読み出し
- ~~**w:** 書き込み~~
- **x:** 実行

その他の人**(o)**
- **r:** 読み出し
- ~~**w:** 書き込み~~
- **x:** 実行

アクセス権の変更 (chmod)

- **chmod** [アクセス権] ファイル名
  - ファイルのアクセス権の変更
  - アクセス権の書式：「どの人に」に続けて「＋」で付与、「ー」で削除

```
$ chmod u+x file          所有者に実行権を与える
$ chmod go-rwx file       所有者以外から全てのアクセス権を除く
```

  - ８進数３桁で各レベルを列挙する

```
$ chmod 600 file          所有者のみ読み書き可能
```

	所有者(u)			グループ(g)			その他(o)		
パーミッション	r	w	x	r	w	x	r	w	x
8進数	4	2	1	4	2	1	4	2	1
設定値	合計値			合計値			合計値		

ファイルの削除 (rm)

- **rm** ファイル名
  - 指定したファイルを削除する
- **rm -rf** ディレクトリ名
  - ディレクトリの中身を確認なしですべて消去した上でディレクトリを削除
- **rm -ri** ディレクトリ名
  - ディレクトリの中身を確認しながら消去

- **「ごみ箱へ移動」ではない**
- **完全に削除してしまう**
- **削除したファイルを復活することは不可能！**

コマンドマニュアルの参照（man）

- **man** コマンド名

```
$ man ls
```

[ ]で囲まれている引数は省略可能

```
ls - list contents of directory

SYNOPSIS
ls [-RadLCxmlnogrtucpFbqisf1AMSDP] [names]

DESCRIPTION
For                                                    the conte
    each file argument, ls repeats its name and any other information
    requested.  The output is sorted alphabetically by default.  When
arguments are not given, the current directory is listed.  When several
    arguments are given, the arguments are first sorted appropriately, but
    file arguments appear before directories and their contents.  ls
  processes supplementary code set characters according to the locale
specified in the LC_CTYPE and LC_COLLATE environment variables [see LANG
    on environ(5)], except as noted under the -b and -q options below
```

指定可能なオプション

下線付きは変数

行数・単語数のカウント（wc）

- **wc** ファイル名
  - ファイルの行数、単語数、文字数を出力する

```
$ wc 1433B_HUMAN.fasta
6   25 481 1433B_HUMAN.fasta
```

    - ファイルは、6行、25単語、481文字からなる。
  - ファイル名を省略すると、端末から入力待ちの状態になる。適当な文章を入力して Controlキー＋d を押すと、入力した文章に対して wc が実行される。

```
$ wc
This is a pen.
(Control+d)
1       4       15          （結果、１行、４単語、１５文字）
```

パターン検索 (grep)

- **grep** パターン ファイル名
  - ファイル中でパターンを含む行を出力する

```
$ grep GO 1433B_HUMAN.sprot
```

    - 1433B_HUMAN.sprot から GO を含む行を検索する。

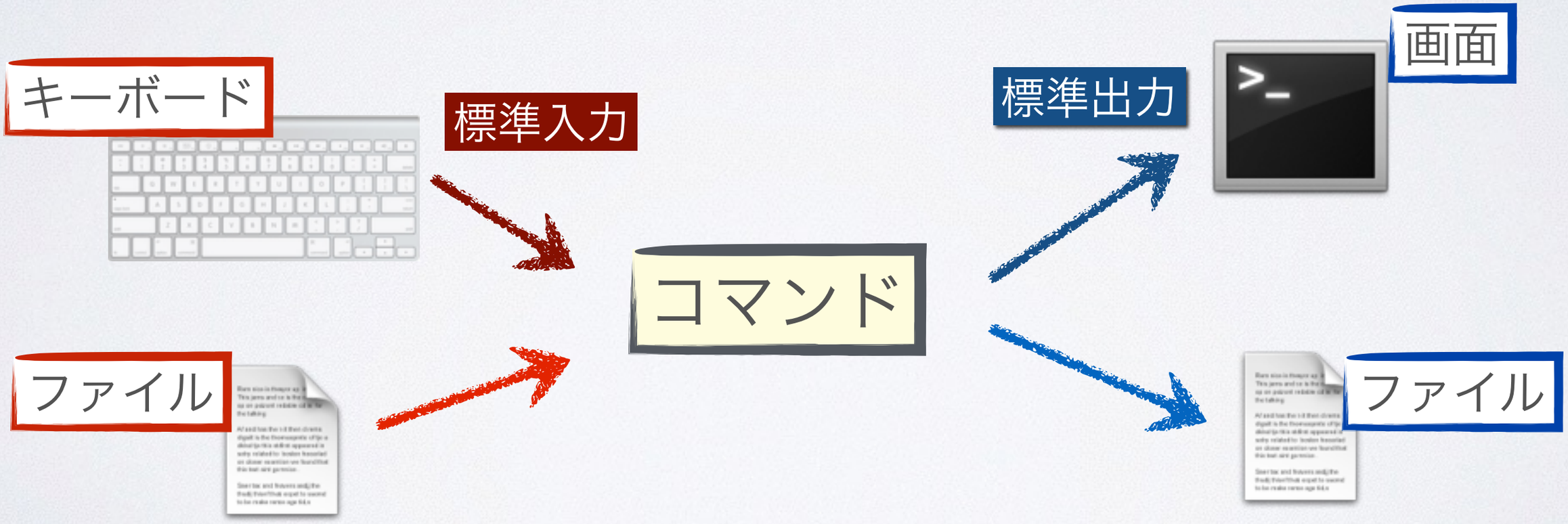
```
$ grep ^FT 1433B_HUMAN.sprot
```

    - 1433B_HUMAN.sprot から FT で始まる行を検索する
    - (“^” は行の先頭を意味する)。

- **grep –v**　とすると（オプション -v）パターンを含まない行を出力する
- ファイル名を省略すると、やはり端末から文字列を読み込んでパターンを検索する

入出力のリダイレクション

- UNIXは、コマンドへの入力元 および 実行結果の出力先を切り替えられる「リダイレクション機能」を持つ
- 通常、標準入力は端末（キーボード）、標準出力は端末（画面）



エラー内容を出力する「標準エラー出力」もある

リダイレクションの例 (出力)

- 結果を画面に出力

```
$ grep GO 1433B_HUMAN.sprot
```

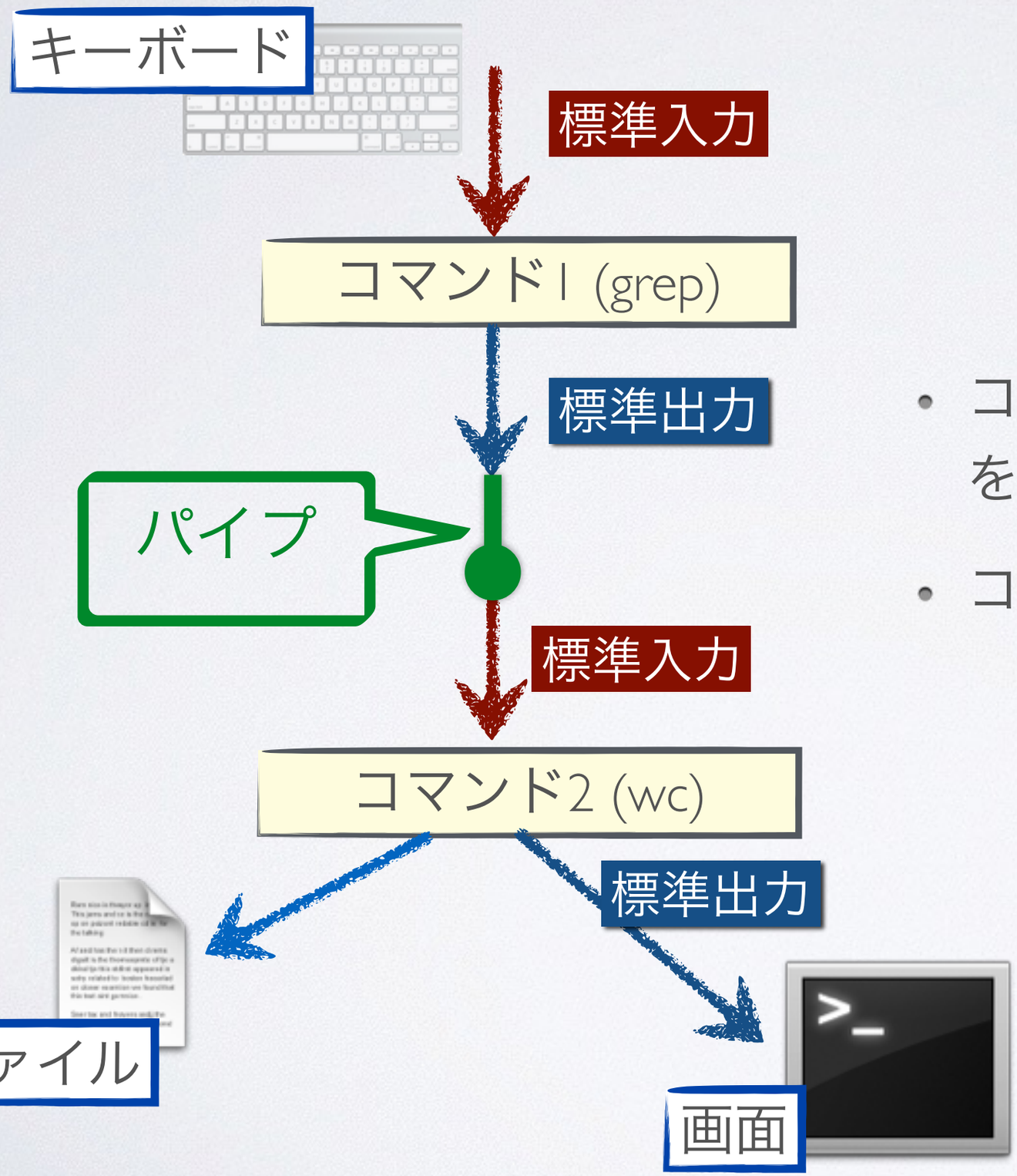
- 結果をファイルに保存：「**>**」
  - コマンドの右に　**> filename**　を加える
  - 結果が GO_count というファイルに格納
  - 同名ファイルがある場合は上書きされる

```
$ grep GO 1433B_HUMAN.sprot > GO_count
```

- すでに存在するファイルに追加書き：「**>>**」

```
$ grep GO 1433B_HUMAN.sprot >> GO_count
```

パイプによるコマンドの結合



- コマンド間の標準出力→標準入力を直接つなげる方法
- コマンドでは「|」で表す

パイプ (|) の例

- grep の出力行数をwcで数える

```
$ grep ^FT 1433B_HUMAN.sprot | wc
```

- grep の結果を less で見る

```
$ grep ^FT 1433B_HUMAN.sprot | less
```

- grepの結果をさらにgrepして絞り込む。

```
$ grep ^FT 1433B_HUMAN.sprot | grep HELIX | less
```

- パイプによって、コマンドはいくつでも結合できる

シェルスクリプト

- 一連のコマンドを記述したファイル（スクリプトファイル）
  - 例）~/data/HN/sprot/testpg の内容

```
#!/bin/sh
pwd
ls -la
```

- 実行権を与え、コマンドとして実行できる。

```
$ chmod +x testpg
```

- 実行

```
$ ./testpg
```

  - コマンドパスの通ったディレクトリに保存すれば、コマンド名のみで実行できる。

コマンドパス

- 打ったコマンドは「コマンドパス」と呼ばれるパスのリストの中から探して実行される
  - 例）**ls** コマンドの場所は **/bin/ls**
- コマンドパスは、それぞれのユーザが持つ設定「PATH変数」に格納されている
  - **`echo $PATH`**　で確認できる。
    - コマンドパスはコロン(：)区切りで複数のパスをつなげて指定する。
    - 同じコマンドが複数のパスに存在するときは、最初のパスのものが優先して実行される
- シェルが、実行プログラムを見つけられない場合、"Command not found" のメッセージが返る。この場合、以下の可能性がある。
  - プログラムが（正しく）インストールされていない
  - コマンドパスが正しく設定されていない

実習２３

コマンドパスの設定

- 一時的にパスを設定する場合

```
$ PATH=$PATH:~/bin
```

  - ホームディレクトリ配下の bin ディレクトリをパスに追加

- 恒常的にパスを設定する場合
  - ~/.bash_profile に下記２行を追記

```
PATH=$PATH:~/bin
export PATH
```

  - ログイン時に読み込まれる初期設定ファイルに、環境変数（シェルから起動されるプログラムにも引き継がれる変数）として PATH 変数を設定する

エイリアス

- **alias** 別名=コマンド名
  - エイリアスとはコマンドの別名のこと
  - エイリアスを新たに作成する

    $ alias 別名='コマンド名'
  - エイリアスを取り消す

    $ unalias 別名

    **$ **別名
- 現在設定されているエイリアスの一覧を確認

  $ alias

- コマンドの初期設定を変更する際に使用
  - 例えば ls の表示結果をカラー対応にする場合

    $ alias ls='ls -G'

メタキャラクタ

- コマンド内で特別な意味を持つ記号
  - 例）**`* ? [ ] < > | ! $ ; &`** など
- コマンドの引数（ファイル名など）にスペースや記号があるときは要注意
- 引数全体を ' ' で囲むことにより、スペースやメタキャラクタの特別な意味を抑止できる

```
$ grep '^>' 1A01_HUMAN.fasta
```

  - 「>」で始まる行を検索。 ' ' をつけないと、リダイレクトと解釈され、 1A01_HUMAN.fasta が上書きされる

- 基本的に、ファイル名には英数字と「.」「_」だけを用い、それ以外の記号は使わないこと

ジョブの中止と中断

- コマンドを起動すると、端末は待ち状態となり、コマンドを受け付けなくなる
  - 実行中のコマンド（ジョブ）を途中で中止
  - コントロールキーと「c」を同時に押す

```
$ Control + c
```

  - ジョブを一時的に中断

```
$ Control + z
```

  - 一時中断したジョブを再開する

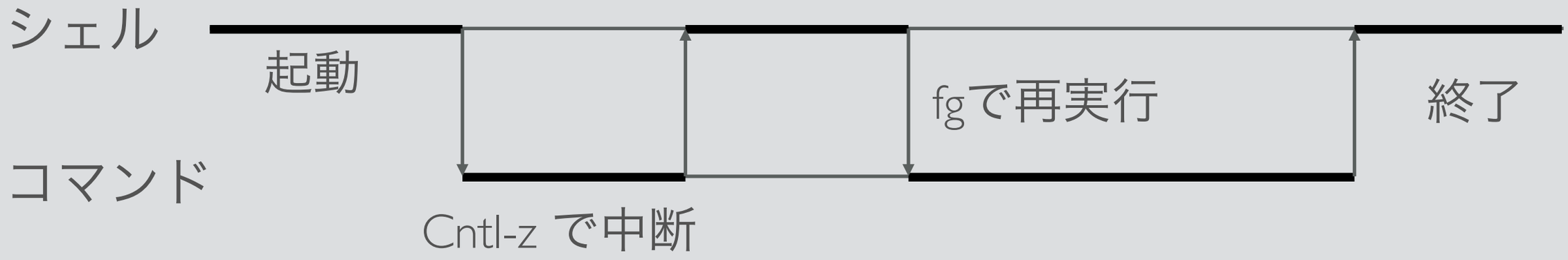
fg　　フォアグラウンドで再開（端末は結果待ちの状態）

bg　　バックグラウンドで再開（端末から入力可能）

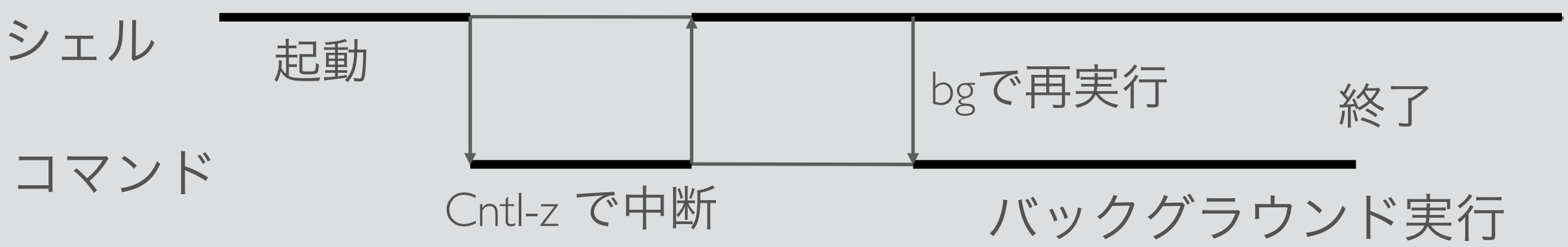
- はじめからバックグラウンドでジョブを走らせるにはコマンドの最後に **&** をつける

フォアグラウンドとバックグラウンド

フォアグラウンド実行（実行中はコマンドを受け付けない）



バックグラウンド実行（実行中でもコマンド実行が可能）



マシンの稼働状況を把握する (top)

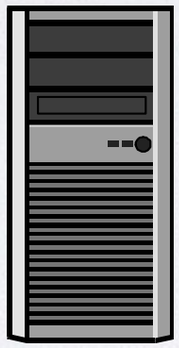
```
$top
Processes: 262 total, 2 running, 7 stuck, 253 sleeping, 1643 threads
18:51:35
Load Avg: 2.98, 1.81, 1.53  CPU usage: 1.40% user, 1.14% sys, 97.44% idle  SharedLibs: 103M resident, 0B data, 17M linkedit.
MemRegions: 125942 total, 9971M resident, 202M private, 2001M shared. PhysMem: 15G used (1918M wired), 8918M unused.
VM: 765G vsize, 1312M framework vsize, 4619(0) swapins, 10377(0) swapouts. Networks: packets: 14584362/19G in, 3809549/380M out. Disks: 3655689/628G read, 3609878/822G
written.

PID    COMMAND      %CPU TIME     #TH    #WQ  #PORTS #MREGS MEM    RPRVT  PURG   CMPRS  VPRVT  VSIZE  PGRP  PPID  STATE    UID  FAULTS     COW      MSGSENT    MSGRECV
99935  Calendar     0.0  00:29.88 4      1    196    1198   88M    79M    0B     116K   518M   3227M  99935 180   sleeping 501  140706     1242     276213     106575
94491  ssh          0.0  00:00.34 1      0    19     48     1032K  660K   0B     0B     63M    2429M  94491 1742  sleeping 501  786        125      36         12
86585- Microsoft Wo 0.0  00:17.71 4      1    151    806    99M    68M+   4096B  0B     362M+  1263M  86585 180   sleeping 501  63647      2262     23468+     22257
85695  GitHub Condu 0.0  00:00.36 4      1    130    133    5996K  4672K  0B     0B     266M   2677M  85695 180   sleeping 501  5499       375      4738       1130
83080  com.apple.hi 0.0  00:00.31 2      0    44     84     2864K  2296K  0B     0B     261M   2627M  83080 1     sleeping 501  4881       197      3375       2251
83009  Keynote      0.0  28:29.73 6      1    492    11182  804M   653M   91M    1552K  1209M  6050M  83009 180   sleeping 501  9963602    46302    5801889    1631017
82764  com.apple.Me 0.0  00:10.42 3      1    82     277    110M   109M   0B     0B     326M   2719M  82764 1     sleeping 501  110668     672      5533       2833
82734  Google Chrom 0.0  00:48.58 10     0    101    515    63M    62M    568K   364K   396M   3544M  72524 72524 sleeping 501  70728      1650     301744     83094
81883  com.apple.ap 0.0  00:16.37 4      1    204    334    43M    37M    28K    0B     278M   2941M  81883 1     sleeping 501  126725     599      92332      44584
81366- mi           0.0  00:09.05 4      1    134    428    35M    26M+   0B     528K   252M+  975M   81366 180   sleeping 501  45749      640      119888     48991
79429  Google Chrom 1.5  24:49.84 12     0    103    1450   132M   135M   4672K  628K   468M   3628M  72524 72524 sleeping 501  466539     1705     506245+    308343+
77397  com.apple.hi 0.0  00:00.01 2      0    34     58     1408K  856K   0B     32K    139M   2505M  77397 1     sleeping 501  1688       209      135        63
77257  com.apple.hi 0.0  00:00.01 2      0    34     53     1316K  764K   0B     24K    106M   2472M  77257 1     sleeping 501  1634       208      117        51
77188  Evernote     0.0  00:51.01 16     2    290    1663   104M   82M    0B     1676K  540M   4255M  77188 180   stuck    501  245407     2002     391446     171586
75296  com.apple.au 0.0  00:00.07 2      1    47     83     2168K  1468K  0B     0B     241M   2609M  75296 1     sleeping 501  2462       269      415        183
75295  com.apple.au 0.0  00:00.01 2      1    28     55     1376K  820K   0B     0B     112M   2479M  75295 1     sleeping 501  1668       190      114        41
75293- com.apple.qt 0.0  00:00.14 2      0    72     133    4592K  3032K  0B     0B     217M   822M   75293 1     sleeping 501  2645       406      1603       759
75158  com.apple.We 0.0  00:05.97 17     1    290    1131   75M    71M    132K   0B     668M   3972M  75158 1     sleeping 501  39398      1628     30708      17195
```

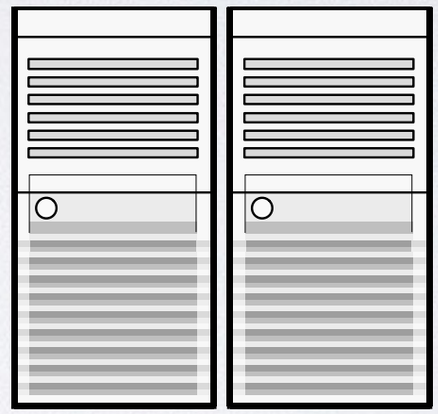
- リアルタイムで稼働状況を表示
- q キーで終了

ネットワークを介したサービス

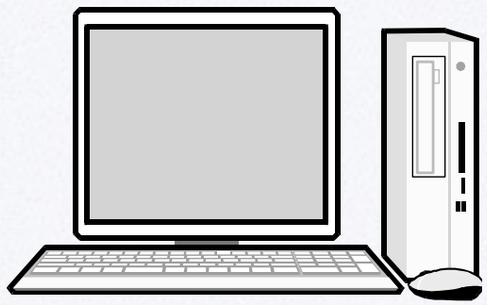
Webサーバ、FTPサーバ



計算サーバ等



ローカル



アカウントが不要なサービス

・Web閲覧

・ファイル転送

（アノニマスftp, curl, wget）

アカウントが必要なサービス

・リモートログイン（ssh）

・ファイル転送（sftp, scp）

ファイル転送 (ftp / sftp)

- **ftp** ホスト名　　または　　**sftp** ホスト名
  - ftpログインしてからのコマンド

コマンド	用途
ls	ホストのファイルの一覧
lls	ローカルのファイルの一覧
cd	ホストのディレクトリ移動
lcd	ローカルのディレクトリ移動
get	ファイルをホストから手元に取得
mget	複数のファイルを取得、ワイルドカード利用
put	ファイルをローカルからホストに送信
mput	複数のファイルを送信、ワイルドカード利用
help	使用できるコマンドの簡易ヘルプ

データ転送 (curl)

- **curl** URL
  - HTTPやFTP経由のファイル取得

    $ curl http://www.nibb.ac.jp
    - http://www.nibb.ac.jp のトップページを標準出力に書き出し

オプション	用途
-o filename url	urlのデータをfilenameに保存
-O url	url上のファイル名で保存（http://hoge.com/fig.jpg のように「ファイル名がある」urlのみ）
-O url[start-end]	http://www.hoge.com/[01-10].jpg 等とすることで、01.jpg, 02.jpg, 03.jpg … 10.jpg のファイルを全て取得
-L url	urlのリダイレクトを追う
-L -Z number url	リダイレクトを追う回数
-h	ヘルプを表示

ファイルの圧縮と解凍

- データ圧縮
  - 一定のアルゴリズムを使い、情報を失わずにファイルサイズを小さくする
  - 圧縮ファイルは使用する前に元に戻す（解凍）必要がある
- **gzip** :
    - 拡張子 **.gz** または **.Z**
    - 圧縮 : **gzip** ファイル名　（ファイル名.gzというファイルができる）
    - 解凍 : **gunzip** ファイル名.gz　または **gzip –d** ファイル名.gz
- **bzip2** :
    - 拡張子 **.bz2**
    - 圧縮 : **bzip2** ファイル名　（ファイル名.bz2というファイルができる）
    - 解凍 : **bunzip2** ファイル名.bz2　または **bzip2 –d** ファイル名.bz2

アーカイブの作成と展開 (tar)

- アーカイブ
  - 複数のファイルやディレクトリを1つのファイルにまとめること
  - 圧縮と組み合わせて、データ配布やバックアップの保存などに用いる
  - 通常アーカイブファイルには拡張子 .tar をつける（自動ではつかない！）
- アーカイブの作成・追加

  $ tar cvf archive.tar directory
  - directory を archive.tar としてアーカイブ
- アーカイブの展開

  $ tar xvf archive.tar
  - archive.tar をカレントディレクトリに展開
- 圧縮・解凍の同時実行
  - オプションに z (gzip)または j (bzip2) をつけることで圧縮・解凍を同時に行うこともできる

    $ tar xzvf archive.tar.gz （gzipで解凍しつつ展開）

プログラムのインストール

- アノニマスFTPまたはWebからファイルをダウンロード
- 圧縮ファイルを解凍してアーカイブファイルを展開
- README, INSTALL などのファイルを見てインストール方法や注意点を確認
- ソースコードをダウンロードした場合はコンパイルする。一般的には以下のコマンドを順に実行する

$ configure	計算機環境に合った設定を自動的に行う
$ make	プログラムをコンパイル
$ sudo make install	プログラムを適当な場所にコピー

SAMtools1.2のインストール

- SAMtoolsのダウンロードと解凍
  - URL　: http://samtools.sourceforge.net/
  - File　: samtools-1.2.tar.bz2
  - 解凍 : tar xvfj samtools-1.2.tar.bz2
- SAMtoolsのコンパイル、インストール、テスト

```
$ cd
$ mv Downloads/samtools-1.2.tar.bz2 unixtest
$ cd unixtest
$ tar xvfj samtools-1.2.tar.bz2
$ cd samtools-1.2
$ less INSTALL
$ make prefix=~ install
$ samtools
```

実習２５

基礎生物学研究所 生物情報解析システム