

## 1-3 UNIX

# 続) UNIXの基礎の理解 / Linux導入

仲里 猛留

NAKAZATO, Takeru

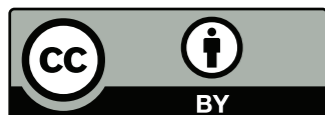


@chalkless



情報・システム研究機構 ライフサイエンス統合データベースセンター

Database Center for Life Science (DBCLS), Research Organization of Information and Systems (ROIS)



2014/9/2

いよいよ最初のコマンド

ターミナルを立ち上げて

えるえす

**ls**

**ls (list) : ファイルのリストを表示する**

```
x v ^ admin1409@BioLinux1409[~]
admin1409@BioLinux1409[admin1409] ls [ 9:24AM]
amelieff Documents genome Music Public Templates
Desktop Downloads igv Pictures R Videos
admin1409@BioLinux1409[admin1409] █ [ 9:24AM]
```



次のことをそれぞれやってみましょう

```
ls igv
```

```
ls -l
```

```
ls -la
```

```
ls -la igv
```

```
admin1409@BioLinux1409[admin1409] ls -la igv
```

ユーザー名

マシン名

いまいるところ  
(カレントディレクトリ)

プロンプト (prompt)

※ 自分流に設定可能



\$ と略す

例：\$ ls -la igv

※ \$ は入力しない

スーパーユーザー (root) のときは # と略す

例：# reboot

\$ ls -la igv

コマンド    オプション    (対象)

オプション

ひきすう

引数 (argument)

実際の命令

実行条件



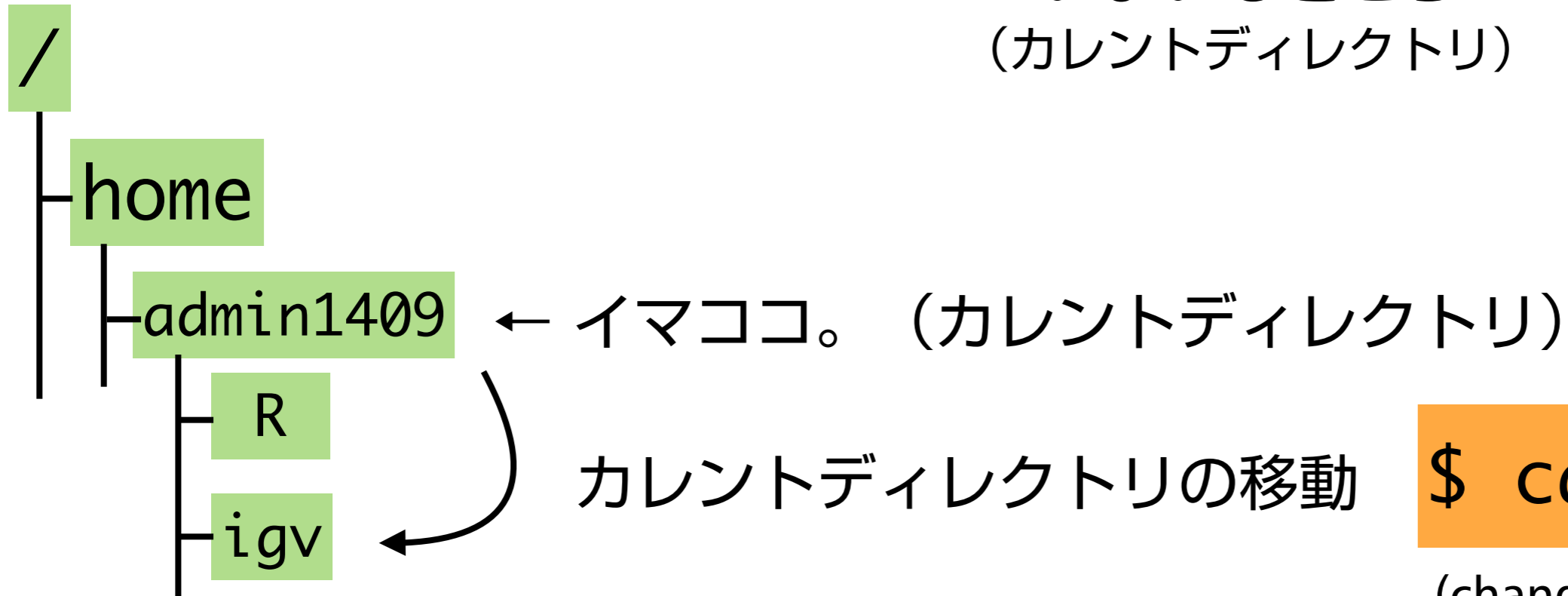
# ディレクトリ構造





```
admin1409@BioLinux1409[admin1409] ls -la igv
```

いまいるところ  
(カレントディレクトリ)



← イマココ。(カレントディレクトリ)

カレントディレクトリの移動

```
$ cd 移動先
```

(change directory)

```
$ pwd
```

```
/home/admin1409
```

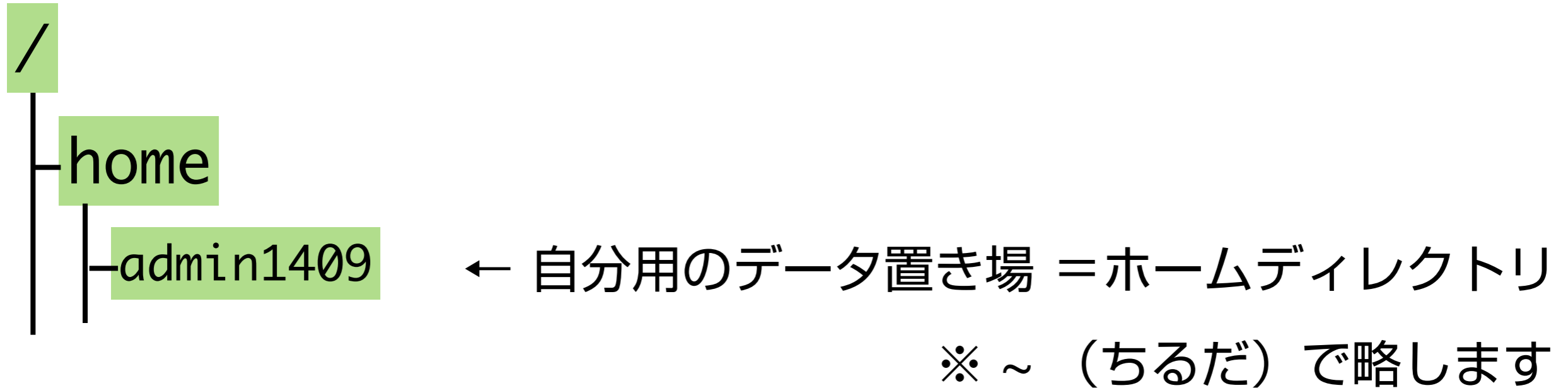
```
$ cd igv
```

```
$ pwd
```

← カレントディレクトリ表示 (positioning working directory)

```
/home/admin1409/igv
```

# ディレクトリの指定あれこれ



`$ cd` → ホームディレクトリに戻ります。つまり `$ cd ~`

カレントディレクトリは .

ひとつ上のディレクトリは ..

例： `$ cd ..`

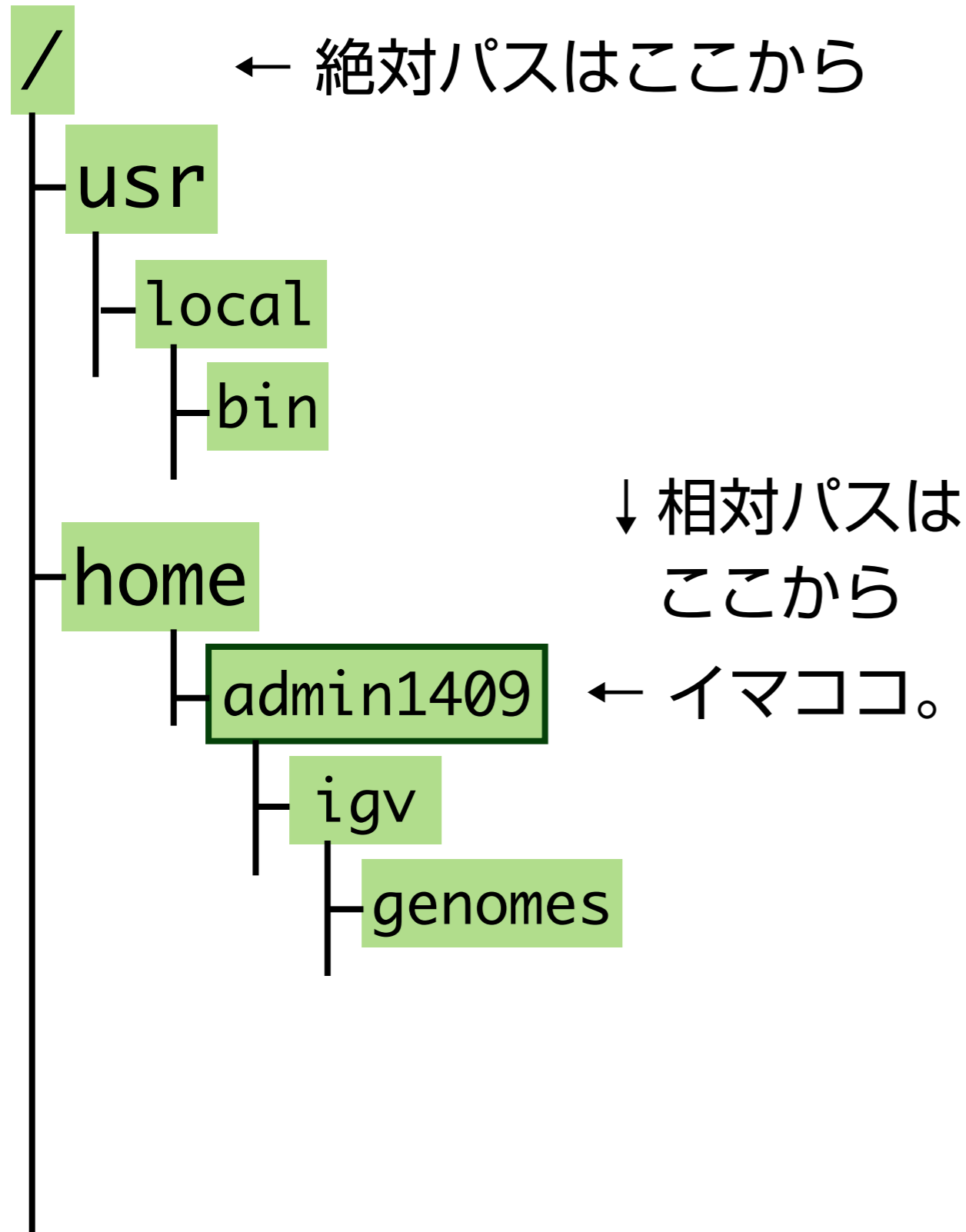
→ 1つ上のディレクトリへ

ディレクトリの区切りには / を用います。

例： `$ cd ~/igv/genomes`

# 絶対パスと相対パス

※ pass でなくて path です



## 絶対パス

```
$ cd /usr/local/bin
```

↑ / から始まる

ルートからの指定

## 相対パス

```
$ cd igv/genomes
```

↑ / から始まらない

カレントディレクトリからの指定

# 基本コマンド・その1

- `ls`, `cd`, `pwd`
- `mv`, `cp`, `rm`
- `mkdir`
- `cat`, `more/less`, `head/tail`

# 権限 (Permission)

```
$ ls -l igv
drwxrwxr-x 2 admin1409 admin1409 4096 Jul 16 16:50 bam
-rw-rw-r-- 1 admin1409 admin1409 7058 Jul 30 14:26 igv.log
```

↑ 頭が r, w, x の3文字 3つ組

↑ 頭が d だとディレクトリ

- ユーザー、グループ、その他で3組
- Read, Write, eXecute で3文字
- chmod : 権限変更、chown : 所有者変更
- su : root になる、sudo : 一時的にrootになる

プログラムを書いても実行権限をつけないと動かないのです

```
$ chmod 755 tekitou.pl
```

各数字は...            read = 4  
                         write = 2        として足し算する  
                         execute = 1

× ユーザー/グループ/その他 で3つ組

この例だと `rwxr-xr-x` になる

※ ユーザーは読み書き実行、グループと一般は読みと実行で書けない

※ `$ sudo chmod +x tekitou.pl` という書き方もある

※ 自分に `chmod` する権限がないときは  
`$ sudo chmod 755 tekitou.pl`

# 基本コマンド ・ その2

- grep
- sort/uniq
- シンボリックリンク ln

# リダイレクト、パイプ

```
$ bowtie2 -x hg19. -U SRR1294107.fastq  
@SQ SN:ILN:230218  
@SQ SN:II LN:813184  
...
```

画面に出力されちゃった。

```
$ bowtie2 -x hg19. -U SRR1294107.fastq > SRR1294107.sam
```

ファイルに出力 ↑

```
$ sort result.txt | uniq
```

↑ プログラム処理結果を次のプログラムに渡す



# ソフトのインストールなど

ダウンロードする

きちんとダウンロードできたか確認

解凍

インストール

# ダウンロード

\$ wget ファイルのありか

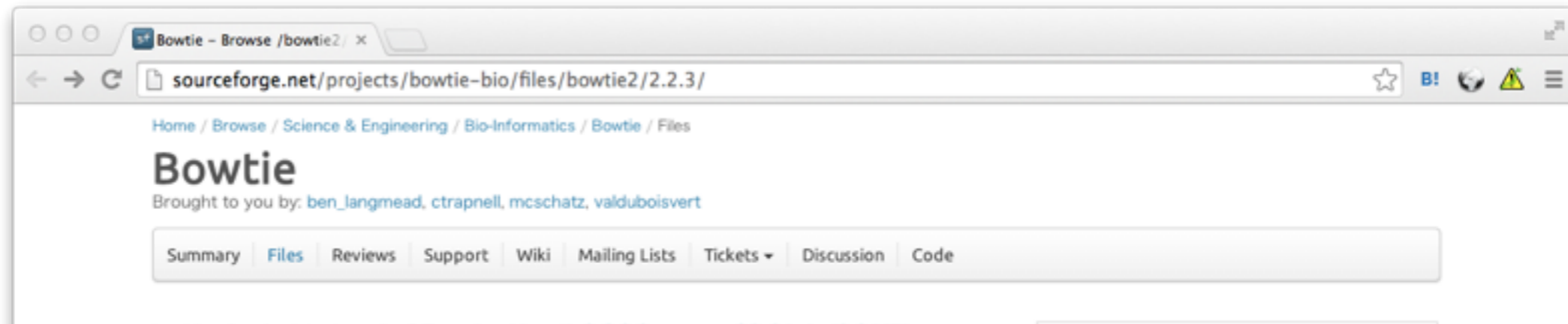
```
$ wget http://cufflinks.cbcb.umd.edu/downloads/cufflinks-2.2.1.tar.gz
```

\$ curl ファイルのありか

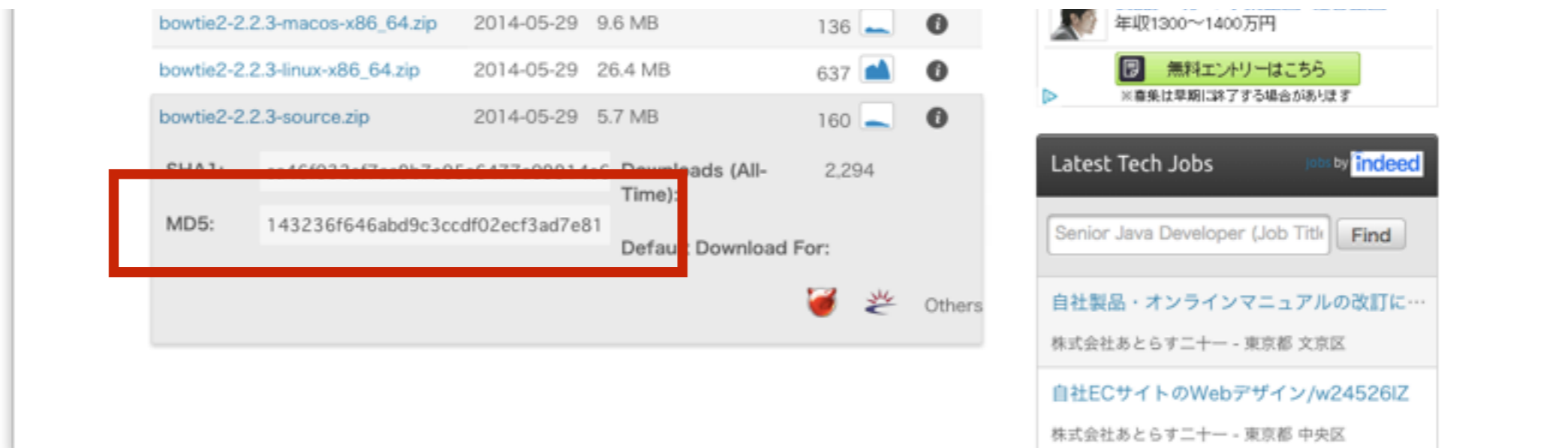
どちらか一方でよいが、どちらかしかない機能も

# ダウンロードの確認

チェックサム (check sum)



```
$ md5sum bowtie2-2.2.3-source.zip  
143236f646abd9c3ccdf02ecf3ad7e81 bowtie2-2.2.3-source.zip
```



# ファイルの解凍

```
$ gunzip hoge.gz
```

```
$ tar xvzf hoge.tar.gz
```

```
$ unzip hoge.zip
```

# プログラムのインストール

通称：ソースから入れる

まずはドキュメント（INSTALL等）を見るべし！

↓ これは一般例で例外も多々あります

```
$ wget ファイル名
```

```
$ md5sum
```

```
$ tar xvzf
```

```
$ cd hoge
```

```
$ ./configure ← えてして --help つけると
```

```
$ ./make 設定可能内容を表示
```

```
$ sudo ./make install
```

# パッケージのインストール

configure | make | make install はメンドい。  
依存関係もややこしい。



パッケージとしてまとめてくれていたり。

```
# apt-get install tekitou      (今回は =Ubuntu)
```

※ ディストリビューションにより apt-get でなく  
yum, port, zypper など

将来、一度は出会います

```
$ bwa  
zsh: command not found: bwa
```

プログラムがないと言われる

```
$ bwa  
Program: bwa (alignment ...  
Version: 0.6.1-r104
```

新しいバージョンを入れたのに  
古いのが実行される

# それはパス (path) です

プログラム等をさがす場所と順番

```
$ ./amelieff/ngs/bwa-0.7.10/bwa  
Program: bwa (alignment ...  
Version: 0.7.10-r789
```

場所も指定して実行

```
PATH="~/amelieff/ngs/bwa-0.7.10:$PATH"
```

設定ファイルに  
書いておく

# 設定ファイル

・ から始まるファイル

zsh だと .zshrc など

bash だと .bashrc など

```
PATH="/usr/local/src/ngsplot/bin:$PATH"
```

すでにある設定 ↑

\$ export すると現在の設定が確認できる

ls で -l オプションをつけると表示される



# エディター (editor)

The image shows a Linux desktop environment with three text editors open, each displaying a Perl script. The script in all three editors is:

```
my @nuc = ("A", "T", "G", "C");  
my $i = int(@nuc);  
while($i>0){  
    $i = $i - 1;  
    print $nuc[$i], "\n";  
}
```

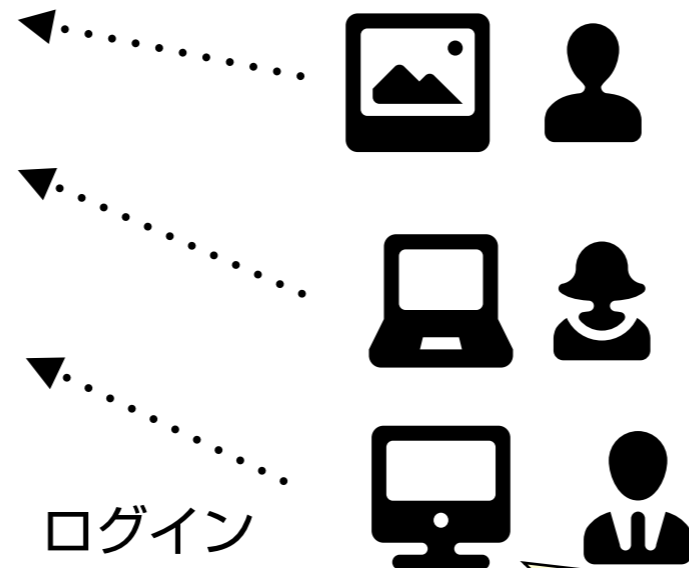
The editors are labeled as follows:

- emacs**: The top-left window, titled "emacs@BioLinux1409". It has a menu bar (File, Edit, Options, Buffers, Tools, Help) and a toolbar with various icons.
- gedit**: The middle window, titled "perl7.pl (~/amelieff/perl/pl\_answer) - gedit". It has a menu bar (Open, Save, Undo) and a toolbar.
- vi**: The bottom window, titled "perl7.pl (~/amelieff/perl/pl\_answer) - VIM". It has a status bar at the bottom showing "Perl", "Tab Width: 8", "Ln 1, Col 1", "INS", "1,1", and "All".

# 基本コマンド ・ その3

- fg, bg
- top, ps, kill, jobs

# 他のマシンにログインして作業



計算はここで

表示だけ


ログイン           \$ ssh ユーザー名@マシン名

ファイルコピー   \$ scp コピー先 ユーザー名@マシン名:コピー先

ログアウト       \$ exit

# screen コマンド



.....~~X~~.....   接続 切れちゃった



以後の計算はウヤムヤに...



 .....   \$ screen

 .....~~X~~.....   接続 切れちゃった

 .....   \$ ssh ...  
\$ screen -r



続きの作業ができる

\$ screen -d で  
おまかせ可能

# 困ったときは

\$ `man コマンド名`  
(manual)

(今日 紹介したような  
Linuxのコマンドの場合)

\$ `コマンド名 -help` とかなんとか

(自分でインストールした  
プログラムの場合)

ググれ。