

Linux 基本コマンド

千葉大学真菌医学研究センター
高橋 弘喜*

2013 年 11 月 26 日

1 基本コマンド

コマンドの入力方法

コマンド名 <オプション> <引数>

引数と区別するためにオプションには“-”を付けることが多い。オプションは複数付けることもでき、複数のオプションを1度に指定することもできる。

表 1: よく使うコマンド

ls	ディレクトリ内容を表示する
cd	特定のディレクトリに移動する
pwd	作業中のディレクトリを表示する
less	テキストをページ単位で表示する
cp	ファイルをコピーする
rm	ファイルを削除する
mv	ファイルの移動と名前変更をする
mkdir	ディレクトリを作成する
rmdir	ディレクトリを削除する

2 ワイルドカード

2.1 ワイルドカード

複数のファイル指定を行うために、ワイルドカードと呼ばれる特殊文字が使用できる

ワイルドカード

- * 任意の文字列
- ? 任意の1文字
- [...] [] 内に含まれる任意の1文字
- [?...] [] 内の... に含まれない任意の1文字
- {~} 内に含まれる任意の文字列

*hiroki.takahashi@chiba-u.jp, URL:<http://bioinfo.pf.chiba-u.jp>, URL:<http://www.facebook.com/chiba.takahashi.lab>

```
1 ls *fastq          # fastqファイルを表示
2 ls *bam            # bamファイルを表示
3 ls projectA*fastq # projectA ... fastqファイルを表示
```

3 基本コマンドの実行

3.1 ls

```
ls <オプション> <パス名>
```

主なオプション

- a ドットファイルも表示する
- A “.”と“..”は表示しない
- l ロング形式で表示する
- F ファイル種類記号を付加する
- t 最近更新された順に表示する
- R ディレクトリ以下の全てのサブディレクトリ内容を表示する
- h 合計サイズとファイルサイズを単位付きの短い数字で表示する

```
1 ls
2 ls -a -l # 別々に指定
3 ls -al   # まとめて指定
```

ディレクトリを指定する時は、

```
1 ls mydir
2 ls mydir/ # “/”はあってもなくてもよい
```

3.2 cd

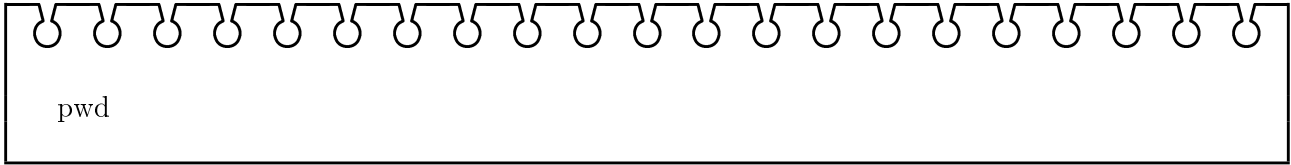
カレントディレクトリを移動する

```
cd
cd <ディレクトリ名>
```

```
1 cd          # ホームディレクトリに戻る
2 cd ~       # 同上
3 cd ~-      # 直前のディレクトリに戻る
4 cd ..      # ひとつ上のディレクトリに移る
5 cd ../dr1  # ひとつ上のディレクトリにあるサブディレクトリdr1に移る
```

3.3 pwd

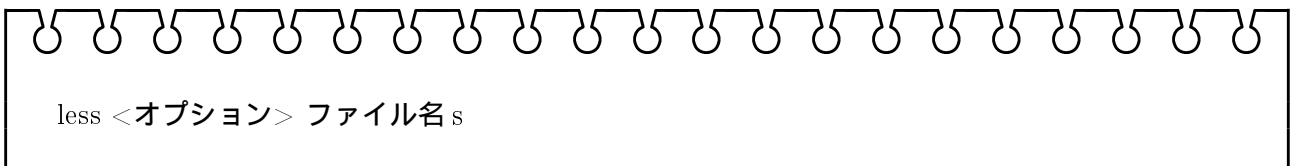
作業中のカレントディレクトリを表示する



```
1 | pwd
```

3.4 less

テキストをページ単位で表示する



主なオプション

- N 行番号も表示する
- 行数 1度にスクロールする行数を指定
- +行番号 指定行から表示開始

```
1 | less myfile
2 | ls -l | less # パイプで渡されたデータをless表示する
```

less 内部コマンド

操作	説明
e j <input type="button" value="Enter"/> <input type="button" value="↓"/>	次の1行を表示
k <input type="button" value="↑"/>	前の1行を表示
f <input type="button" value="Space"/> <input type="button" value="PageDown"/>	次の1画面を表示
b <input type="button" value="PageUp"/>	前の1画面を表示
g	1行目を表示
G	ファイルの最終行を表示
/文字列	文字列を検索
?文字列	文字列を先頭方向へ検索
q	lessを終了

3.5 cp

ファイルをコピーする

```
cp <オプション> ファイル名1 ファイル名2
cp <オプション> ファイル名s ディレクトリ名
cp -r <オプション> ディレクトリ名1 ディレクトリ名2
```

主なオプション

- i 確認しながらコピーする
- r ディレクトリ内容を全コピーする

```
1 cp file1 file2 # file1をfile2にコピー
2 cp file1 mydir2 # mydir2に同名でコピー
3 cp f1 f2 mydir2 # f1とf2をmydir2にコピー
4 cp mydir2/* . # mydir2の全ファイルをカレントディレクトリにコピー
```

3.6 rm

ファイルを削除する

```
rm <オプション> ファイル名s
```

主なオプション

- i 削除確認メッセージを表示する
- f 一切の確認メッセージなしで強制削除する
- r 指定ディレクトリをまるごと削除する

```
1 rm f1
```

3.7 mv

3.7.1 ファイル

ファイルの移動と名前変更をする

```
mv <オプション> ファイル名1 ファイル名2
mv <オプション> ファイル名s ディレクトリ名
```

主なオプション

- i 確認メッセージを表示する

ファイルの移動

```
1 mv file1 mydir # 1個のファイルをmydirディレクトリに移動
2 mv file1 mydir/file2 # 移動とともに名前を変更する
3 mv file1 file3 mydir # 複数のファイルを移動
4 mv mydir/file1 . # カレントディレクトリに移動
5 mv mydir/file1 file1 # 同上
```

ファイル名の変更

```
1 mv file100 file200
```

3.7.2 ディレクトリ

ディレクトリの移動と名前変更をする

```
mv ディレクトリ名1 ディレクトリ名2
```

引数にディレクトリを指定したときは、ディレクトリの移動または名前変更が行われる。ディレクトリ2が存在しない場合は、ディレクトリ名の変更が行われる。

```
1 mv dir1 dir2
```

3.8 mkdir

ディレクトリを作成する

```
mkdir <オプション> ディレクトリ名s
```

主なオプション

-p 途中のディレクトリも生成する

```
1 mkdir dir1
2 mkdir dir1 dir2
```

```
1 mkdir dir55
2 mkdir dir55/dir66
```

```
1 mkdir -p dir55/dir66 # dir55とその下のdir66を同時に作成する
```

3.9 rmdir

ディレクトリを削除する

```
rmdir <オプション> ディレクトリ名 s
```

主なオプション

-p 途中のディレクトリも削除する

```
1 rmdir dir1
2 rmdir dir2 dir3
3 rmdir dir44/dir55 # dir55だけを削除
4 rmdir -p dir44/dir55 # dir44もdir55も削除
```

4 便利なコマンド

4.1 wc

ファイルのバイト数・単語数・行数を表示する

```
wc <オプション> ファイル名 s
```

主なオプション

-c バイト数
-w 単語数
-l 行数

```
1 wc -l file1 # 行数
2 wc -w file1 # 単語数
```

4.2 grep

ファイルの中から指定された文字列を検索し、その行を出力する

```
grep <オプション> 文字列パターン ファイル名 s
```

主なオプション

- n 行番号も表示
- c マッチした行数だけ表示
- e str 文字列 str を検索する
- f file パターンを file から読み込む
- v マッチしない行を表示

```
1 grep ATGC file1 # file1の中から‘ATGC’のある行を表示
2 grep -v ATGC file1 # ‘ATGC’のない行を表示
3 grep -c ATGC file1 # 一致する行数を表示
4 grep -e -AA file1 # ‘-AA’のある行を表示
5 grep -e AAA -e TTT file1 # ‘AAA’または‘TTT’のある行を表示
6 grep AA file1 | grep TT # ‘AA’と‘TT’のある行を表示
```

4.3 sort

ファイル内容をソートする

```
sort <オプション> ファイル名 s
```

主なオプション

- r 逆順にソートする
- t 区切り文字指定
- k n1 [,n2] フィールド n1 から末尾を対象にソート
- o ファイル名 指定したファイル名に出力する

```
1 sort file1 # 先頭から比較してソート
2 sort -r file1 # 逆順にソート
3 sort file1 -o file2 # 結果をファイルに出力
4 sort file1 > file2 # 同上
```

4.4 uniq

ファイル内の重複行を削除する。

重複検査は隣接している行に対して行われる。対象ファイルはソートされている必要がある。

```
uniq <オプション> 入力ファイル名 出力ファイル名
```

主なオプション

- u 重複していない行だけを表示
- d 重複している行だけを表示
- c 重複している行数を表示

```
1 uniq file1 # 重複行を削除
2 uniq file1 file2 # file2に結果を出力
3 sort file1 | uniq # 未ソートの場合
```

4.5 awk

テキスト形式で記述されたデータを処理する簡単なプログラミング言語です

```
1 awk 'プログラム' file # コマンドラインからプログラムを指定
2 awk -f progfile file # progfileにプログラムを記述
3 awk 'NR%2==0' file # 偶数行取り出し
4 awk 'NR%2==1' file # 奇数行取り出し
5 awk 'NR%4==2' fastq # fastq内のシーケンスデータの取り出し
```

5 リダイレクトとパイプ機能

5.1 リダイレクト

入出力先を別のものにしたい時は、標準入力先、標準出力先の切り替えを指定できる。

- 標準入力 (stdin)
- 標準出力 (stdout)
- 標準エラー出力 (stderr)

```
1 ls > myfile # lsコマンドの出力をmyfileに入れる
2 grep ATGC file1 > myfile # file1のATGCを含む行をmyfileに入れる
```

5.2 パイプ

パイプはあるコマンドの標準出力へのデータを、別のコマンドの標準入力データとして渡す機能。

```
1 prg1 | prg2 # prg1の出力をprg2に送り込む
2 prg1 | prg2 | prg3 # prg1の出力をprg2に送り込み、prg2の出力をprg3に送り込む
```

```
1 ls | less # lsの結果をlessで確認
2 ls | wc -l # ファイルの総数をカウント
```


6 vi エディタ

vi は二つのモードがあります。

1. コマンドモード
2. テキスト入力モード

```
1 vi file1
2 vi -R file1
3 vi
```

ファイル操作と終了操作

:e filename	指定した名前のファイルを開く
:w filename	指定した名前でファイルを保存する
:w	ファイルを保存する
:q	vi を終了する
:q!	ファイルを保存せずに vi を強制終了する
:wq	ファイルを保存して vi を終了する
ZZ	同上


テキスト入力モードへの移行コマンド

i	現在文字位置の前から入力開始
a	現在文字位置の次から入力開始
I	現在行の先頭から入力開始
A	現在行の末尾から入力開始
o	現在行の次に空白行を設けて入力開始
O	現在行の前に空白行を設けて入力開始

カーソル移動操作

←→↑↓	一文字移動
h	左
j	下
k	上
l	右
0 ^	カーソルを行先頭へ移動
\$	行末へ
gg	先頭行へ
G	最終行へ
Ngg NG	n 行へ

テキストのカット&ペースト

x 	カーソル位置の 1 文字を削除
X	カーソル位置の左の 1 文字を削除
dd	カーソルのある行を削除
d\$	カーソル位置から行末までを削除
d^	先頭からカーソル位置までを削除
y<カーソル移動指示>	カーソル移動指定位置までコピー
yy	カーソルのある行をコピー
p	現在位置の後ろにペーストする
P	現在文字の前にペーストする

参考文献

- [1] 林晴比古 “新 Linux/UNIX 入門 第 3 版”