**Linuxの基本コマンド**

# ディレクトリ、ファイル操作を中心としたLinux基本コマンドを扱います

本資料においてコマンドの前の $ マークはプロンプト（コマンド入力が待機中であることを意味する記号）を示しますので、入力は不要です

誤った操作などによって、入力が受け付けなくなってしまった（プロンプトが表示されなくなった）場合には、control-Cを押すことで、処理をキャンセルし入力待機状態に戻ることができる場合があります。

“q” を押すことで処理を中断させられる場合もあります。

また、タブキーを素早く２度押すと、利用可能なコマンドやファイル名が補完されます（候補が多すぎて表示しきれない場合 “q” を押すと途中でとめられます）

矢印キーの上下で過去に入力したコマンドの履歴が表示できます。

これらをなるべく活用することで手入力による打ち間違いを避けるのがコツです。

**はじめに**

スパコンログイン直後は、スパコンシステム環境への入り口であるゲートウェイノード(gw)にいます。gw では解析処理等を行うことは想定されておらず、負荷の高い処理を行うと他のユーザにも影響がでることがあります。そのため、スパコンにログインした後に

$ qlogin

コマンドを実行し、計算ノード（インタラクティブノード）に移動します。

($ 記号はプロンプトを示しますので入力不要です。)

qlogin を実行する際に、パスワードを要求されます。スパコンアカウントのパスワードを入力してください（認証鍵を生成するときに指定したパスフレーズとは異なります）。パスワードは”passwd”コマンドを使って変更することが可能です。パスワード入力を回避する方法もあります（後述）。

講習用のファイルを取得するために次のコマンドを実行してください。

$ git clone https://github.com/genome-sci/basic\_course

計算ノードから抜けるには

$ exit

を実行します。さらにgwにおいて exit を実行するとスパコンからログアウトできます。

1. **現在の状況を把握する (pwdとls)**

pwd:  現在自分がいるディレクトリ（フォルダ）を表示します。(print working directory)

$ pwd

ログイン直後には各ユーザーのホームディレクトリ /home/userXXX/ にいるはずです。

ls: 現在のディレクトリ内にあるファイルおよびディレクトリの一覧 (list) を表示します。

$ ls

さきほど取得した練習用のファイルを含んだbasic\_courseというディレクトリが見えるはずです。

$ ls basic\_course

とするとbasic\_courseの中身が表示されます。

不可視ファイル ( . で始まるもの）を含めて表示させます。

$ ls -a

このコマンドによって表示されたディレクトリのうち、

“.” は現在のディレクトリ、”..” は一つ上の階層のディレクトリ（親ディレクトリ）を示します。

また、-a はコマンドラインオプションのひとつです。

linuxのコマンドの多くはコマンドラインオプションを使うことで、既定の動作を変更させることができます。

ファイルの更新日時やファイルサイズも含めて表示させます。

$ ls -a -l

上記は ls -al としても同様の結果が得られます。

また、多くのシステムでは ll が "ls -l” のショートカットとして利用できます。

$ ls --help

多くのコマンドでは -h -help --help などのオプションをつけることでヘルプを表示できます。

1. **ディレクトリの移動 (cd)**

ディレクトリを移動します。

$ cd basic\_course/

一つ上のディレクトリ（もとのディレクトリ）に戻るには、”..”を指定します。

$ cd ..

ホームディレクトリに戻るには

$ cd

または

$ cd ~

とします。”~”はホームディレクトリを示す記号です。

直前にいたディレクトリに戻る時には

$ cd -

を実行します。

1. **Linuxのディレクトリの基本構造**

/ (ルートディレクトリ)

    /bin（おもにLinuxシステムの動作に最低限必要な実行ファイルを格納する）

    /usr（各種アプリケーションと、それに付随するファイルを格納する）

        /usr/local/biotools  各種生命科学用ツールのsingularityイメージが格納されている

        /usr/local/pkg/  NIGスパコンではPythonなどプログラミング言語がバージョンごとに置かれている

        /usr/local/seq/  NIGスパコンではDDBJの塩基配列データやBLASTデータベースが置かれている

    /home  各ユーザーごとのホームディレクトリが格納されている

        /home/userXXX/   userXXXのホームディレクトリ

        /home/userYYY/   userYYYのホームディレクトリ

        /home/userZZZ/   userZZZのホームディレクトリ

UNIX/Linuxではファイルやディレクトリごとに所有者や書き込み・読み込み権限（パーミッション）が設定されており、通常は他のユーザーからはファイルが見られないようになっています。

同じグループに属するユーザーであれば一般的には閲覧可能です。

自分が所有するファイルであれば権限を変更することも可能です（chmodコマンド）

1. **絶対パスと相対パス**

/usr/local/bin ディレクトリに移動します。

$ cd /usr/local/bin

/ （ルートディレクトリ）を起点にした場所の示し方を「絶対パス」と呼びます。

/usr/local/bin から /usr/local/biotoolsディレクトリに移動します。

$ cd ../biotools

現在いる場所を起点にした場所の示し方を「相対パス」と呼びます。

1. **ディレクトリの作成と削除**

練習用ディレクトリに移動します。

$ cd ~/basic\_course/linux\_command

$ pwd

/home/XXXXXXX/basic\_course/linux\_command

ディレクトリの新規作成

mkdir コマンド (make directory)を使い “test"という名称のディレクトリを作ります。

$ mkdir test

ディレクトリを削除します。

$ rmdir test

このコマンドは空のディレクトリに対してしか使用できません。

ディレクトリとその中身のファイルも含めて削除する場合には代わりに

$ rm -r test

コマンドを使用してください

1. **touchコマンドとrmコマンド(ファイル作成と消去)**

touchコマンドを使い空のテキストファイル (a.txt) を作成します

$ touch a.txt

なおテキストファイルであることが目で見てわかりやすいように拡張子 .txt をつけていますが、

Windowsとは異なり、Linuxでは拡張子によってファイルの種類を認識することはありません。

したがって、拡張子は必須ではありません。（touch aでaという名称の空のファイルが作られます）

ファイルの消去

$ rm a.txt

1. **echo コマンド、cat コマンド、リダイレクト**

echoコマンド：文字列を画面に出力します

$ echo Hello

> を使うと画面の出力をテキストファイルに書き込むことができます（リダイレクト）。

$ echo Hello > a.txt

a.txtがすでに存在していた場合、上書きされますので気をつけてください。

>> で追加書き込みできます

$ echo world >> a.txt

他のコマンドの出力結果もファイルに書き込めます。

$ ll > b.txt

catコマンド：ファイルの中身を画面に出力します

$ cat a.txt

catコマンドは本来は複数のファイルを連結（concatenate）するコマンドです。

$ cat a.txt b.txt > c.txt

でa.txtとb.txtを連結し、その結果をc.txtに書き込みます。

1. **ワイルドカード**

.txt で終わるファイルの一覧を表示

$ ls \*.txt

\*は0文字以上の任意の文字列を示します

$ ls ?.txt

?は任意の１文字を示します

$ ls ??.txt

と打った場合には ab.txt や XY.txt にマッチしますが、a.txtにはマッチしません

これまでに作成した .txtファイルを削除します

$ rm \*.txt

(単に rm \* と打つと全てのファイルが消えてしまいますのでご注意ください。

また、-r オプションをつけて rm -r \* を実行すると、カレントディレクトリ以下にあるファイルがすべて消えます。linuxでは一度削除したファイルを復活させることはできませんので、使用する場合には注意してください）

1. **cpコマンド（ファイルのコピー）とmvコマンド（ファイルの移動）**

a.txtをAディレクトリに移動します。

$ mv a.txt test1/

(最後の/はなくても構いません）

b.txtをBディレクトリにコピーします。

$ cp b.txt test2/

(最後の/はなくても構いません）

ファイルを別名で複製することもできます。

b.txtをb2.txtとして複製します

$ cp b.txt b2.txt

mvコマンドはファイル名を変更するときにも使用します。

b.txtをc.txtに名前を変えます

$ mv b.txt c.txt

test2に移動します

$ test2

一つ上のディレクトリにある c.txt をカレントディレクトリに移動します。

$ mv ../c.txt ./

結果的に b.txtとb2.txtとc.txtの両方がカレントディレクトリに存在するはずです。

mvやcpで複数のファイルやディレクトリを同時に移動・コピーすることも可能です。（講義では省略）

1. **headコマンドとtailコマンド, wc**

それぞれファイルの先頭および末端部分を表示します。

$ cd ../test1

$ pwd

/home/XXXXXX/basic\_course/linux\_command/test1

$ head test1-1.gbk

で先頭から10行を抽出して表示します。

行数を指定することも可能です。

$ head -100 test1-1.gbk

末端から100行を抽出し、ファイルに書き込みます。

$ tail -100 test1-2.gff > out.txt

wcで行数を数えて見ます

$ wc test1-1.gbk

表示される数値は順に、行数、単語数、文字数を表します。文字数には改行コードも含まれます。

-l をつけると行数のみ表示します

$ wc -l test1-1.gbk

1. **less コマンド とmoreコマンド**

cat コマンドはファイルの中身を一気に表示させるのに対し、大きなファイルを部分ごとに表示させるのがlessやmoreコマンドです。

lessとmoreは多少の違いはありますが、lessはmoreの拡張版ですので、lessを使うことの方が多いです。

$ less test1-1.gbk

で起動します。

矢印キーの上下左右でスクロール。

fとbで１画面ずつ上下にスクロールします。

h でヘルプが表示できます。

q でlessを終了します。

参考）

lessコマンドは、デフォルトでは画面の右端で自動的に折り返されて表示されます。この設定を無視して画面１行にファイルの内容を１行ずつ表示させる場合には ”-S” オプションを使用します。

$ less -S read.fastq

1. **パイプ**

“ | “ を使うとコマンドの出力を他のコマンドに連結することができます（読み方はパイプ、パイプライン、バーチカルバーなど）

例）ファイルの先頭1000行を抽出した後、その末尾100行を表示させます。

$ head -100 test1-1.gbk | tail -10

結果的に90行目から100行目が抽出されて表示されます。

複数のパイプを連結することも可能です。

例）上記の結果に対して wc を実行します。

$ head -100 test1-1.gbk | tail -10 | wc

1. **ファイル内の検索 （grep）**

ファイルの中から文字列AAAAを含んだ行を表示させます。

$ grep LOCUS test1-1.gbk

パイプと組み合わせて使用することもできます

使用例）

スパコンの実行中ジョブの一覧から待機状態(qw)状態のジョブを表示させる。

$ qstat -u "\*" | grep qw

1. **プログラムの実行（パスを通す）**

$ cd ~/basic\_course/linux\_command/

この中にあるhelloというプログラム（画面に” Hello world!”と表示されるだけのプログラム）を実行する方法を説明します。

実行権限の付与

$ chmod a+x hello

実行方法は3通りあります。

1. 相対パスで実行

$ ./hello

（カレントディレクトリにあるプログラムを実行する場合には明示的に”.”をつける必要があります）

1. 絶対パスで実行

$ /home/XXXXXXX/basic\_course/linux\_command/hello

または

$ ~/basic\_course/linux\_command/hello

(“~”はホームディレクトリを指す記号)

1. パス（PATH）を通す

パスとは、コマンド検索パスを意味します。

あるコマンドを実行するためには、そのコマンドの存在するディレクトリが環境変数 PATH で

指定されている必要があります。

$ printenv PATH

を実行すると現在の設定が確認できます。(echo $PATHでもOK)

exportコマンドを使って現在いるディレクトリ（~/basic\_course/linux\_command）をパスに加えます。

$ export PATH=~/basic\_course/linux\_command:$PATH

これで”hello”コマンドを打つだけでプログラムが実行できるようになりました。このように環境変数PATHを設定し、コマンドを実行できるようにすることを「パスを通す」といいます。

“hello”コマンドの実体がどこにあるかを示すには下記を実行します。

$ which hello

~/basic\_course/linux\_command/hello

export コマンドでPATHの設定に失敗してしまうとコマンドが一切使えなくなってしまうことがありますが、その場合にはターミナルを閉じて再度ログインをすると回復します。

参考1) 次のようにPATHの後側に付け足すこともできます。PATHは先頭に近い方から探索されていくため、前につけた場合とは意味が若干異なります。（同名のファイルがあった場合、先に見つかった方が優先される）

$ export PATH=$PATH:~/basic\_course/linux\_command

参考２）ログアウトすると export コマンドで設定した環境変数の内容は失われてしまいますので、ログインごとに再設定する必要があります。頻繁に使うツールについては、exportコマンドを .bash\_profile ファイル（または .bashrcファイル）の中に記述するとログイン時に自動で設定されるようになります。

**補遺 以下は講習では扱いませんが、比較的よく使うコマンドやtipsを紹介します。**

1. **wget インターネット上からファイルを取得する**

URLを指定してインターネット上のファイルを取得します。

例）DDBJのFTPサーバーからシークエンスデータを取得します。

$ wget ftp://ftp.ddbj.nig.ac.jp//ddbj\_database/dra/fastq/SRA117/SRA117449/SRX456287/SRR1151187\_1.fastq.bz2

1. **ファイルの展開および圧縮**

zip 形式 → 展開 unzip、圧縮 zip

gz 形式 → 展開 gunzip、圧縮 gzip

bz2 形式 → 展開 bzip2、圧縮 bzip2

tar or tar.gz 形式 → 展開および圧縮 tar コマンド

例 tar.gzファイルの展開

$ tar xvfz archive\_file.tar.gz

xは展開（extract）, vは展開の過程を画面に表示（verbose）を表します。

zは対象が gz 形式で圧縮されている場合につけますが、最近はつけなくても展開できます。

fは展開する対象がファイルであることを示します。

1. **scp ローカルマシンとリモートサーバー（スパコン）との間でのファイルの転送を行う**

以下はローカルマシン上で操作する例です。some\_fileというファイルをホームディレクトリにコピーします。

scp some\_file USERNAME@gw.ddbj.nig.ac.jp:~/

スパコンからローカルマシンにコピーすることも可能です。

1. **パスワードの変更**

$ passwd

のあと、現在のパスワードを入力、

続いて新しいパスワードを２回（確認のため）入力します。

1. **vi と emacs**

どちらもターミナル上で動作するテキストエディタです。

本講習ではSFTPクライアントソフトを使ってリモートのファイルを編集しますので、

これらは使用しません。

遺伝研スパコンを含め、一部のLinuxシステムではviが標準のエディタになっているため、

最低限の操作方法は覚えておいた方が良いです。

起動

$ vi

または

$ vi ファイル名

viはコマンドモードと入力モードに分かれています。

入力モードに切り替えるには i をコマンドモードに戻るときは ESC を押します。

コマンドモードにおいて :w で保存、:q で終了します。

1. **JAVA で作られたプログラムを動かすときの注意点**

スパコン上でJAVAで作られたプログラムを動かす際に、

$ java

Error occurred during initialization of VM

Could not allocate metaspace: 1073741824 bytes

というようなエラーが表示されて動作しない場合があります。この場合には下記のように環境変数 \_JAVA\_OPTIONS を指定しておくとエラーが回避できます。

export \_JAVA\_OPTIONS="-Xmx1g -XX:ParallelGCThreads=1"

1gの部分は使用したいプログラムによっては 256m 512m or 2g などにした方が良い場合があります。

export \_JAVA\_OPTIONS="-XX:ParallelGCThreads=1"

だけでも良いかもしれません。

参考）

<https://supcom.hgc.jp/japanese/utili_info/manual/faq.html#2829>

1. **qlogin時にパスワード入力を回避する**

詳細は省きますが、パスワード認証のかわりに鍵認証を行うように設定します。

$ ls ~/.ssh/id\_rsa.pub

を実行してみて、ファイルが存在していないことを確認します。ファイルが存在していなければ、

$ ssh-keygen

を実行します。ファイルの保存場所やパスフレーズを聞かれますが、すべてデフォルトのまま（空欄のまま）エンターキーを押します。

id\_rsa.pubファイルが存在していればこの操作は不要です。

次に、以下のコマンドを実行します。

$ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

以上です。

1. **パーミッションの設定（chmod）**

ls -lでファイルの情報を表示させると、次のように先頭に

rやwからなる10桁の文字列が表示されます。

-rw-r--r--  1 tanizawa  staff  0  3 15 14:45 a.txt

これはファイルやディレクトリのパーミッションを示しています。

最初の１文字目はファイル種別を表しており、

- は通常のファイル、lはシンボリックリンク、dはディレクトリを示しています。

2文字目から4文字目はファイルの所有者に対する権限を表し、

5文字目から7文字目はファイルの所有グループに対する権限を表し、

8文字目から10文字目はその他に対する権限を表しています。

- 権限なし（数値で表すと0）

r 読み取り（数値で表すと4）

w 書き込み（数値で表すと4）

x 実行可能（ディレクトリの場合は中のファイルにアクセス可能）（数値で表すと1）

たとえば、上記の a.txt の場合

ファイルの所有者は読み書き両方ができ(4+2=6)、

同一グループに属するユーザーおよびその他のユーザーは読み込みのみ可能(4)です。

これを数値で表すと 644 となります。

パーミッションを変更するには chmod コマンドを用います。

たとえば、

$ chmod a+x a.txt

とすると、すべてのユーザ (a)に対し実行権限(x)を付与する(+)という意味になり、

パーミッションは以下のようになります。

-rwxr-xr-x   1 tanizawa  staff     0  3 15 14:45 a.txt

また、数値で直接指定することもでき、

$ chmod 751 a.txt

とすると、パーミッションは

-rwxr-x---   1 tanizawa  staff     0  3 15 14:45 a.txt

というようになります。

下記のリンク先を参考にしています。

<https://qiita.com/shisama/items/5f4c4fa768642aad9e06>

<http://inaz2.hatenablog.com/entry/2013/09/01/224223>

1. **新しいアプリケーションをインストールする**

ほとんどのプログラムは管理者権限を持っていなくても自分のホームディレクトリにインストールして使用することが可能です。

ソースコードをダウンロードして自分でコンパイル（＝実行可能なファイルを作成する）する場合もあれば、コンパイル済みの実行ファイル（バイナリファイル）をダウンロードする場合もあります。

コンパイル済みのファイルをダウンロードする場合には、”Linux x86\_64”用のものであればスパコンで動作します。

例）トランスクリプトームマッピングツール HISAT2 のインストール

配布元：<https://ccb.jhu.edu/software/hisat2/index.shtml>

ソースコード：<ftp://ftp.ccb.jhu.edu/pub/infphilo/hisat2/downloads/hisat2-2.1.0-source.zip>

バイナリファイル：<ftp://ftp.ccb.jhu.edu/pub/infphilo/hisat2/downloads/hisat2-2.1.0-Linux_x86_64.zip>

下記はhisat2をコンパイルするときの手順例です。

ソースコード取得

$ wget <ftp://ftp.ccb.jhu.edu/pub/infphilo/hisat2/downloads/hisat2-2.1.0-source.zip>

展開

$ unzip hisat2-2.1.0-source.zip

ディレクトリに移動

$ cd hisat2-2.1.0

インストールの手順を確認する。最初に読むべき重要なファイルは “README" のように大文字で記載されていることが多い。hisat2の場合は “MANUAL” がそれに該当する。

マニュアルに従い、make コマンドでコンパイルを行う

$ make

以上でhisat2およびその関連プログラムがコンパイルできます。

プログラムによっては make の前に “configure”コマンドを実行したり、make後に “make install “を行うものもありますが、基本的には マニュアルに従って行えばそれほど難しくはありません。

1. **プログラミング環境の構築（中級者以上）**

スパコンにはPythonやPerlなどのプログラミング言語がインストールされていますが、異なるバージョンや標準では用意されていないライブラリが必要になることがあります。このような場合に、ホームディレクトリに自分でプログラミング環境を整えておくと便利です。

pyenvやplenvというツールを使うとプログラミング言語のインストールを簡単に行うことができ、また、異なるバージョンを切り替えて使用することも可能になります。インストール方法については pyenv, plenvなど各ツールで検索すると日本語の情報が多く得られますので参考にしてください。