Linux (Ubuntu)基礎

Linux ではアプリのアイコンをクリックして起動する以外に、

ターミナル(Terminal，端末)を起動し、それに対してコマンドを打ち込む

ことで起動する（このターミナルソフトは、Windows の「コマンドプロンプト」に相当するが、もっと拡張されている）。

ここでコマンドの名前と意味、それに作業用ディレクトリなど Linux のファイルシステムについて学ぶ。

1）最初のコマンド: ls

まずは ls コマンドを打ち込んでみよう。（ls は L と S の小文字---Linux では大文字と小文字を区別するので注意）

すると、Desktop などの語が表示されたことでしょう。

これは、「現在の(current)ディレクトリにあるファイルやディレクトリが表示された」ものです。つまり ls とは、今いるディレクトリに何があるかを調べるために使うコマンドなのでした。ちなみに、「今いるディレクトリ」とは ls により中身が表示されるディレクトリのことで、正式には「作業用ディレクトリ(working directory)」と言います。

さて、ディレクトリとはなんでしょうか？これは Microsoft Windows の「フォルダ」と同じものです。ただ、Linux では、もっと体系的で統一されたものになっています。

まずシステム全体は / (スラッシュ一個)からなるディレクトリの下にあります。これをルート(root)ディレクトリといいます。あとで見るように、すべてのディレクトリやファイルの「親元」です。その下には、usr や home などのディレクトリがあります。それらは、 /usr とか /home のように表されます。またこれらはその下にいろいろなディレクトリを含みます。例えば /usr というディレクトリの下には、local や bin というディレクトリがあります。これらは、 /usr/local とか /usr/bin という名前で参照できます。ちょっとややこしいですが、/ はディレクトリとディレクトリをつなく記号でもあります。ここで、/usr/local を例にとると、これは /usr というディレクトリの中のディレクトリです。このとき、/usr を親ディレクトリ、もしくは「上」のディレクトリといいます。/usr からみると/usr/local は「子」もしくは「下」のディレクトリになります。このようにディレクトリに親子関係があるシステムをディレクトリ階層といいます。そしてこのディレクトリ階層はしばしば木構造で表されます（普通の木と違って、根(root)が上にあります）:

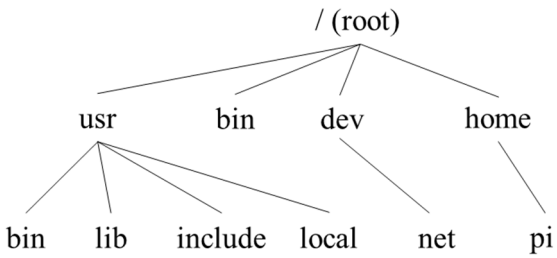

図 1. Linux のディレクトリ階層を「木構造」で表示したもの

さて、それでは、Terminal を起動して、見えているディレクトリはなんでしょうか？

これを調べるには、 pwd というコマンドを使います。pwd とは、 print working directory の頭文字をとったもので、これから意味は明らかですよね。

演習 1 (a) pwd を実行した結果を書け。なお、立ち上げた直後の作業用ディレクトリのことをホームディレクトリ、もしくは単にホーム(home)と言います。

(b) 図 1 に示した木構造ではホームはどれか、マークせよ。

2) ディレクトリを渡り歩く

Linux での仕事は Microsoft Windows とはかなり異なります。まず Terminal を起動し、作業用ディレクトリを適切なものに設定し、その中のファイルを編集し、コンパイル(Microsoft でいうビルドに相当)し、ファイルを呼び出して実行する、というのが標準的な作業です。つまり、「アイコンをクリック」というプログラムの起動方法はあまり使いません(プログラムを作る方としては、Linux のやり方の方が簡単なのです)。

そのためには、まず、編集や実行の対象となるファイルがあるディレクトリを、作業用ディレクトリに設定しなければなりません。そのために使われるのが cd コマンドです。cd は change directory(ディレクトリを変更する)の頭文字です。cd の使い方には大雑把にいって、3通りの方法があります。

(1) 今の作業用ディレクトリの中にあるディレクトリを、新たな作業用ディレクトリに設定する

例えば、今の作業用ディレクトリに src というディレクトリがあったとします。なお、src があるかどうかは ls コマンドで確認できますし、それがディレクトリであるかどうかは ls で表示される色で区別できます---青色がディレクトリです。その src ディレクトリを新たな作業用ディレクトリにするには、

cd src

というコマンドを実行します(最後に改行を忘れないこと)。配布した英文資料では cd ./src のように、「./ 」がつけられていますが、この2つは同じことを意味します(ので、気にしないでください)。このようなディレクトリの指定方法を相対パスといいます。この「相対」とは次に述べる「絶対」の反対語です。そしてこの意味は、src として指定されたディレクトリが(同じ名前のディレクトリやファイルが他所にもあるかもしれませんが)、今の作業用ディレクトリの下にあることを意味しています。

(2) 次は、絶対パスによる指定です。LinuxOS には、/usr/bin というディレクトリがあります。それを作業用ディレクトリに指定するには次のようなコマンドを実行します:

cd /usr/bin

相対パスとの違いは、ディレクトリの指定に / から始まる名称を書いていることです。このように、ルート(/)からはじめて、どのディレクトリの中に指定したディレクトリがあるかという道筋を全部書く(いわば住所を地球、日本国からはじめて、愛知県、名古屋市、昭和区、というように全部書く方式、と思ってください)のが絶対パスです。ちなみにパス(path)とは道、という意味です。

相対パスが今いるところからどうやって道をたどると対象とするディレクトリにたどり着けるか、を書いたものであるのに対し、絶対パスは必ずルートからのたどり方を書いたものです。

ここで、相対パスだと「子」のディレクトリや「孫」のディレクトリを指定するのはできるのはわかりましたね。しかし「親」はどうやって指定したらよいでしょうか?

相対パスで「親」のディレクトリを指定するには「. . 」と書きます。ドットを2つ並べたものです。

(注意:半角文字を常に使ってください)。今、/home/pi が作業用ディレクトリであるとします。そこから相対パス方式で、/usr/bin を作業用ディレクトリにするには、今述べたように

cd ../../usr/bin

と書きます。 図1をみながら考えてみましょう。最初の..で作業用ディレクトリが /home となります。次の .. でルートになります。そこから usr/bin と指定すればよい、というわけです。

言い換えれば相対パスは作業用ディレクトリを基点とした、絶対パスはルート(/)を基点としたファイルやディレクトリの指定の方法というわけです。

(3) 3番目の cd の使い方は、次のようなものです:

cd

この結果については次の演習で確かめることにしましょう。

演習 2 以下をこの順に答えよ(実行せよ)。

(a) 今の作業用ディレクトリを /home/pi とする。そうなっていなかったら、cd コマンドを用いてそうせよ。また、実際に作業用ディレクトリが /home/pi であることを確かめよ(今までのコマンドを用いて)。

(b) ls コマンドにより表示された単語と色を記録せよ。

(c) file * を入力/実行し、その表示と (b) の結果を比較せよ。そこから、ls で表示された色の意味を推定せよ。この結果から推測できるように、file とは指定されたファイルの属性を示すコマンドで、*は、そこにあるすべてのファイルを意味する。

(d) ../../lib を作業用ディレクトリとせよ(本当にそうなっていることを確かめよ)。そのためのコマンドを答えよ。図1にそのディレクトリの位置を書き込め。また、作業用ディレクトリを変更した後、そこにあるディレクトリの個数を答えよ。

(e) /usr/include/X11 を作業用ディレクトリとせよ(本当にそうなっていることを確かめよ)。そのためのコマンドはどのようなものか?

(f) cd コマンドを実行し、その結果、作業用ディレクトリが何になったかを答えよ。

3) 今まで見たように、ディレクトリはとても重要なものです。仕事や種類によってファイルを分類して、何が入っているか、どういう作業に関係しているかが分かるような名前をつけたディレクトリを作ることは、作業効率にとても影響します(これは、コンピュータ上の作業だけではなく、日常生活でも同じです)。今までは、すでにあるディレクトリについて見てきました。ここでは、新しくディレクトリを作ったり、消したり、すでにあるディレクトリのコピーの方法を見ていきます。

演習 3 以下をこの順に答えよ(実行せよ)。

(a) /home/pi を作業用ディレクトリとする。そこにどういうディレクトリやファイルがあるかをメモする。

(b) mkdir testDir を入力、改行する。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。

(c) mv testDir sampleDir を入力、改行する。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。

(d) sampleDir を作業用ディレクトリとせよ。そのためにどのようなコマンドを実行すればよいか？

(e) ls コマンドでどのような表示がされるか、またそれはなぜかを答えよ。

(f) echo "sample" > test1　を入力、改行する。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。
参考:　> は「ファイルのリダイレクション」といい、echo コマンドの結果をファイルに書き込む、という操作を表す。ファイルのリダイレクションには > 以外にも数種類ある。

(g) cp test1 test2 を入力、改行する。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。

(h) cat test2 を入力、改行する。何が表示されたか、またそれはどうしてか考えよ。

(i) rm test1 を入力、改行する。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。

演習 3 を行って予想できたかと思いますが、そこで使われたコマンドの説明を下に書きます。いずれも Linux では基本的なコマンドです。

コマンド名	意味
mkdir 名前	「名前」という新たなディレクトリを作業用ディレクトリの下に作る(make dir)
mv 名１ 名２	「名１」というファイル／ディレクトリの名前を「名２」に変更する(move)
cp 名１ 名２	「名１」というファイルの中身をコピー(copy)したファイルを作り「名２」とする
cat 名前	「名前」というファイルの中身をディスプレイに表示する
rm 名前	「名前」というファイルを削除する(remove)

ここまでは、ファイルとディレクトリの違いをはっきり書いてきませんでした。ここでは、ファイルとはテキストやプログラムなどをさし、ディレクトリとは下にいろいろなファイルやディレクトリを作れる「フォルダ」をさすものとします。基本的にはファイルのコピーや削除という操作コマンドは、ディレクトリのコピーや削除のためのコマンドとしても使えますが、コマンドに対してパラメタ（余分な引数）が必要です。ここでは –r というパラメタを紹介します。この r は再帰的(recursive)を表すもので、簡単にいえば「繰り返し実行」を意味します。このパラメタはディレクトリのコピーと削除などに使えます。

演習 4 演習 3 を行っているものとする。以下をこの順に答えよ（実行せよ）。

(a) 演習 4 で作成した sampleDir を作業用ディレクトリとせよ。

(b) 作業用ディレクトリを、その親のディレクトリに変更せよ。そのためのコマンドはなにか？

(c) cp –r sampleDir testDir を入力、改行せよ。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。

(d) rm sampleDir を入力、改行せよ。どのようなメッセージが出たか？

(e) rm –r sampleDir を入力、改行せよ。どのようなメッセージが出たか？

ここでメッセージが表示されるような場合は、 コントロールキーを押しつつ C を入力し（これはコントロール C と言い、コマンドの強制停止を意味する）、改めて rm –rf sampleDir を入力、改行せよ。作業用ディレクトリにあるディレクトリやファイルにどのような変化があったかを答えよ。

参考: 作業用ディレクトリの中にあるファイルやディレクトリ名は先頭の1，2文字を打ってから tab キーを押すと、その名前を補完してくれる。また、上下矢印キーでは前のコマンドを再利用でき、左右矢印キーを使って、Terminal に打ち込むコマンドの簡単な編集ができる。

Terminal はいくつでも起動できる。

ファイルの編集には gedit を使うと良い（これ以外にも vim などが使える）

Cプログラミングのためには、 gcc がある。これでコンパイルができれば、 a.out という実行ファイルができる。

Python プログラミングのためには、 python と python3 がある。python のバージョンを調べるには
python –V　　または　python3 -V
とすればよい。

白井ゼミの Linux マシンのアカウントは stud （パスワードはゼミの時に知らせる）
白井ゼミのウェブページは(世界中からアクセス可能)

http://lang.sist.chukyo-u.ac.jp/Classes/seminar/index.html