知能情報工学演習 I

柳本 豪一

平成 19 年 4 月 19 日

1 Linux

ここでは、これから授業で使っていく Linux についての説明と、よく使われる用語について説明していきます。

1.1 Linux と OS

Linux とはフィンランドのヘルシンキ大学の当時大学院生であった Linus Torvalds 氏によって開発された OS です。

OS とは入出力機能 (キーボードからの入力・画面出力など)、ディスクやメモリの管理、多くのソフトウェアから利用される共通した基本的な機能を提供する**コンピュータ全体を管理するソフトウェア**です。OS が提供する機能を利用することによって、ソフトウェアの開発者は開発の手間を省くことができます。さらに、ハードウェア (コンピュータ自体) の違いは OS で吸収されるため、特定の OS に対して開発されたソフトウェアは、同じ OS が動作する異なるハードウェア上でも動作することが保証されます。

代表的な OS としては、Microsoft 社の **Windows** シリーズ、Apple 社の **Mac OS**、フリーソフトである **Linux** や **FreeBSD** などの OS があります。Linux や FreeBSD は **UNIX 系の OS** と呼ばれ、この授業では RedHat Linux を通して、UNIX の使い方を勉強していきます。

Linux にはディストリビューションと呼ばれる様々な種類のものが提供されています。Linux 自体は基本的な機能のみを提供するソフト (カーネルと呼ばれる) を指しますが、単独では使いにくいために様々なソフトウェアが一緒にまとめられています。このように必要なソフトウェアをまとめて配布されているものをディストリビューションと呼びます。RedHat Linux はそのようなディストリビューションの一つであり、他にも FedraCore、VineLinux などがあります。

1.2 Windows と Linux

おそらく一度は今までに使用したことがあると思われる Windows と比較しながら、簡単に Linux での用語を説明していきます。

まずは Windows の画面を見てみましょう。図 1 に Windows の画面の一例を載せます。Windows では、**デスクトップをメタファー**にした画面を持っています。このため、それぞれの名称がデスクトップにあるものの名前が付いています。例えば、**ファイル**、**フォルダ**、**マイコンピュータ**などがそうです。このように、グラフィックを用いてコンピュータ内の状況を表し、マウスにより操作を行う仕組みのことを**グラフィカルユーザインターフェイス**と呼びます。

図 1: Windows の画面

一方、Linux でも **X Window System** を用いてグラフィカルユーザインターフェイスを提供しています。しかし、この授業では X Window System 自体の使い方ではなく、キーボードからコマンドを入力することで、Linux を利用する**キャラクターベースのインターフェイス**について勉強していきます。X Window System については、Windows と操作とよく似ているので、特に困ることはないと思います。

Linux では、Windows と少し呼び方が異なるものがあります。例えば、Windows の「フォルダ」に対応するものを、Linux では「**ディレクトリ**」と呼びます。またこれに対応して、Windows で「フォルダを開く」ことを、Linux では「**ディレクトリに移動する**」といいます。このように Windows と Linux では操作の呼び方が異なっているので、注意をしてください。

1.3 基本的なディレクトリ

Linux では、いくつか特別な名前を与えられているディレクトリがあります。

まず最初は**ホームディレクトリ**です。ホームディレクトリは、ログインしたときに最初にいる

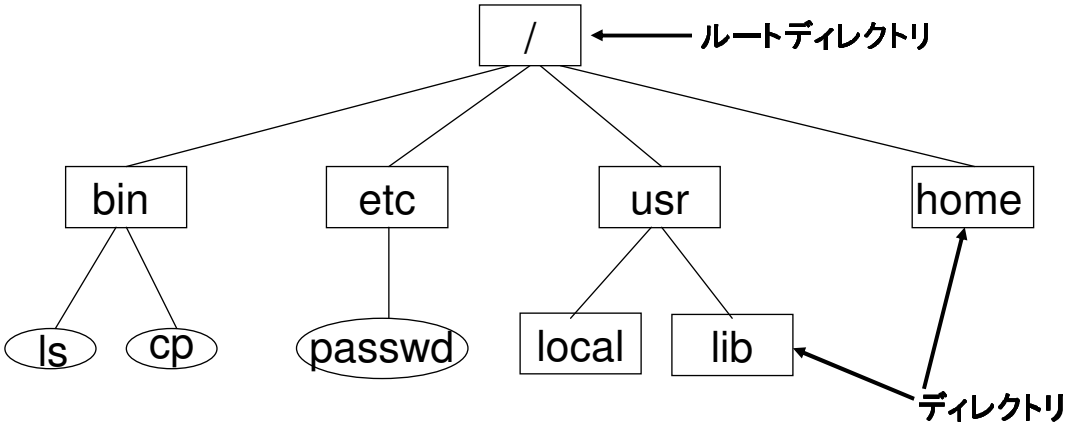

図 2: ツリー構造

ディレクトリになります。Linux では、複数の利用者がコンピュータを問題なく同時に利用できるように、ホームディレクトリは各自異なったディレクトリとなっています。Linux では、「~」でホームディレクトリを表します。

自分のホームディレクトリを確認するため、ターミナルを起動して「pwd」と入力してください。表示された文字列が自分のホームディレクトリです。隣の人と比べて、自分のホームディレクトリと異なっていることを確認してください。

次は「**ルートディレクトリ**」です。Linux では、ファイルは**ツリー構造**によって管理されています。ツリー構造とは、図 2 に示すような構成を表します。ツリー構造では、必ず始まりを示すディレクトリがあり、それをルートディレクトリと呼びます。Linux では「/」で表します。

ここで先ほど表示されたホームディレクトリを見直してみましょう。例えば、「/home/nt/thy00057/unix」と表示された場合、先頭の「/」はルートディレクトリを表し、他の「/」はディレクトリの区切りを表します。つまり、ルートディレクトリ下の、home ディレクトリ下の、nt ディレクトリ下の、thy00057 ディレクトリ下の、unix ディレクトリがホームディレクトリになっていることを表します。

それでは、ルートディレクトリに移動してみましょう。ターミナルで「cd /」と入力します。問題なくプロンプト (利用者に入力をうながすシステムからの表示される記号) が戻ってきたら、「pwd」と入力します。画面上に「/」と表示されれば、ルートディレクトリに移動できています。「ls」と入力するとルートディレクトリにある全てのファイル (ディレクトリも含みます) が表示されます。

最後は**カレントディレクトリ**です。カレントディレクトリとは、現在自分がいるディレクトリを表します。Linux では、「.」で表されます。

後で説明しますが、ディレクトリを移動するコマンドは「cd」です。今は、ルートディレクトリにいるので、いろんなディレクトリに移動してみましょう。カレントディレクトリを表示するためには、「pwd」を利用します。ディレクトリを移動した後、「pwd」を実行すると表示されるディレクトリ名が変化していることが分かると思います。

2　基本的なファイル操作

Linux でファイルを操作するため、基本となるいくつかのコマンドを紹介します。これらのコマンドはすべてターミナル上から実行することになります。

2.1　man

man はコマンドの使い方を表示するコマンドです。使い方がわからないコマンドがあれば、「man コマンド名」と入力すると説明が表示されます。この説明には、使い方や、オプションの設定の仕方などさまざまな情報が載っているので、新しいコマンドに出会ったときには man で確認するようにしましょう。man の説明も登録されているので、「man man」と入力してみましょう。

man の簡単な使い方です。

- スペースキー
  次のページを表示します。
- q
  表示を終了します。

2.2　pwd

pwd は、カレントディレクトリを表示するコマンドです。Linux を使っていると、他のディレクトリに移動し、今いるディレクトリがわからなくなることがあります。このとき、pwd でカレントディレクトリを確認します。

pwd の詳しい使い方を知るため、man を使って pwd の説明を表示しましょう。

2.3　ls

ls は、ディレクトリの中身を表示するコマンドです。Linux では、ディレクトリもファイルとして扱われるので、ls によりディレクトリに含まれているディレクトリ (**子ディレクトリ**と呼ばれます）も表示されます。ls にはさまざまなオプションが用意されており、これを使いこなすことでさまざまなファイルの情報を表示することができます。たとえば「-l」オプションは、「ls -l」と入力することで、ファイルの所有者や大きさなどが表示されます。他にも「-a」オプションを付けることで、「ls」では表示されなかった「.」で始まるファイルも表示されます。「.」で始まるファイルは設定ファイルであることが多いので、削除や修正をする時には注意してください。オプションは組み合わせることもでき、「-l」と「-a」を組み合わせるときには、「-la」または「-al」とします。

では、カレントディレクトリのファイルを表示してみてください。オプションを付けて、ファイルを表示してみてください。man を使って、ls の説明を読んでみましょう。

2.4　cat

cat は、ファイルの中身を表示するコマンドです。使い方は「cat ファイル名」です。

それでは、ホームディレクトリに「.cshrc」があることを確認して、中身を表示させましょう。man を使って、cat の説明を読んでみましょう。

2.5 mkdir

mkdir は、ディレクトリを作成するコマンドです。Windows でフォルダを作成したように、mkdir を用いてディレクトリを作ることができます。使い方は、「mkdir 新しいディレクトリ名」となります。

それでは、「test」というディレクトリをホームディレクトリに作成し、無事作成できたことを確認しましょう。また、man で mkdir の使い方を調べましょう。

2.6 cd

cd は、ディレクトリを移動するコマンドです。使い方は「cd ディレクトリ名」です。

cd の引数であるディレクトリ名ですが、Linux では 2 通りの指定の仕方があります。**絶対パス名**と**相対パス名**の 2 つです。絶対パス名は、pwd で表示されるようなルートディレクトリから表示される形式です。絶対パス名で表すことで、ディレクトリの場所を一意に決定することができます。一方、相対パス名とは、カレントディレクトリを中心にしてディレクトリを指定する方法です。つまり、カレントディレクトリと移動するディレクトリの相対的な位置を記述します。

たとえば、mkdir で作成した「test」というディレクトリに移動してみましょう。絶対パス名では pwd で表示されたディレクトリ名に「/test」を追加したディレクトリ名を入力することで移動ができます。一方、相対パス名では、「cd test」と入力するだけで移動できます。相対パス名では、カレントディレクトリを中心に移動するので、入力する文字数が少なくすむ場合があります。

ディレクトリ test に移動したので、元のホームディレクトリに戻るにはどうすればよいでしょうか。絶対パス名では、ホームディレクトリで pwd を実行した時に表示されたディレクトリ名を入力することになります。一方、相対パス名では、カレントディレクトリの一つ上のディレクトリ（**親ディレクトリ**）を表現するのに「..」が用意されています。したがって、ディレクトリ test の親ディレクトリに戻るには、「cd ..」と入力することになります。このような特殊な表現としては、「.」と「~」があります。「.」はカレントディレクトリを表し、「~」はホームディレクトリを表します。特に「~ユーザ名」と入力するとそのユーザのホームディレクトリに移動します。また、よく使われる cd の使い方として、引数を持たずに実行する「cd」があります。これは、ホームディレクトリに戻る方法として利用されます。

それでは、cd を使っていろいろな場所に移動してみましょう。たとえば、友人のホームディレクトリやルートディレクトリなどいろいろあります。移動できる場所がわからないときは、自分で調べてみましょう。また、man を使って cd の使い方を調べましょう。最後に、自分のホームディレクトリに戻りましょう。

2.7 cp

cp は、ファイルをコピーするコマンドです。Linux では、ディレクトリもファイルなので、cp でコピーすることができます。使い方は、「cp コピーするファイル名 コピー先」です。

cp は、指定されたファイルをコピー先にコピーします。ここで、コピー先が (1) 存在しないファイル名、(2) すでにあるファイル名、(3) ディレクトリ名の場合が考えられます。まず、存在しないファイル名の場合は、その名前のファイルが新しく作成されて内容がコピーされます。すでにあるファイルの場合には、すでにあるファイルに内容が上書きされてしまいます。したがって、もとのファイルの内容がなくなってしまうので、注意が必要です。最後に、ディレクトリ名の場合です

が、そのディレクトリにファイルがコピーされます。ファイルの名前は、指定されたファイルと同じ名前です。

ディレクトリをコピーするには、「-r」オプションを付ける必要があります。使い方は「cp -r ディレクトリ名 コピー先」となります。

それでは、ディレクトリ「`~thy00057/ecsis1/1/practice`」を自分のホームディレクトリにコピーしてみましょう。次に、自分のホームディレクトリの practice に移動し、「test1」と言うファイルを「test5」という名前のファイルにコピーしてみましょう。最後に、man で cp について調べてみましょう。

2.8 mv

mv は、ファイルを移動するコマンドです。cp に似た動作をしますが、元のファイルがなくなってしまう点が異なっています。使い方は、「mv 移動するファイル名 移動先」です。

基本的には cp と同じ動作をするので、cp の動作を参考にしてください。mv は、ファイル名を変えるときによく利用されます。そのときには、「mv 元のファイル名 新しいファイル名」と書きます。

それでは、test2 を test6 に名前を変えてみましょう。man で mv の使い方を調べてみましょう。

2.9 rm

rm は、ファイルを削除するコマンドです。使い方は「rm ファイル名」です。

rm はファイルを削除するコマンドなので、注意してください。Windows では、一時的にゴミ箱に削除したファイルが残っていたり、ツールを使うことで復活させることができます。しかし、Linux では、一度削除したファイルを復活させることができません。したがって、rm を使うときには本当に消しても構わないのか注意するようにしてください。「基本コマンド」の授業で説明する、「*」（任意の文字列を表す）や「?」（任意の一文字を表す）と組み合わせて使うことが多いのですが、不注意で必要なファイルを消してしまうことがよくあります。

それでは、practice ディレクトリに移動して、rm を用いて cp で作成したファイル「test5」を削除してみましょう。practice にある他のファイルも消してみましょう。最後に、man で rm について調べてみましょう。

2.10 rmdir

rmdir は、ディレクトリを削除するコマンドです。ファイルを削除する rm のディレクトリ版となります。使い方は、「rmdir ディレクトリ名」です。

rmdir はディレクトリにファイルや子ディレクトリが存在する場合、そのディレクトリを削除することができません。したがって、rmdir でディレクトリを削除する前に、ディレクトリに含まれるすべてのファイルや子ディレクトリを削除しておく必要があります。少し面倒くさい気がしますが、必要なファイルを消してしまわないためにも、慣れるようにしてください。

それでは、ホームディレクトリに作成したディレクトリ「test」を削除してみましょう。次に、自分の好き名前のディレクトリを作成して、そのディレクトリを削除してみましょう。最後に、man で rmdir について調べてみましょう。

2.11 more

more は、ファイルの中身を表示するコマンドです。cat とは異なり、1 ページごとに表示されます。使い方は、「more ファイル名」です。

more の使い方の簡単な説明です。

- スペースキー
  次のページを表示します。

- q
  表示を終了します。

more は長い表示結果を分割して見ることができるので、単独で使うより他のコマンドと一緒に使うことが多いです。これについては、次に説明します。

それでは、ディレクトリ practice にあるファイル test1 の中身を見てみましょう。man で more について調べてみましょう。

3 リダイレクションとパイプ

ここでは、Linux を便利に使うポイントである、リダイレクションとパイプについて説明を行います。リダイレクションを使うことによって、プログラムの実行結果をファイルに保存することが簡単にできます。また、パイプを利用して、複数のコマンドを組み合わせることができます。パイプにより、簡単な処理を行うコマンドを組み合わせて複雑な処理を行うことが Linux の使い方のポイントです。

3.1 リダイレクション

リダイレクションとは、プログラムの入出力をキーボードや画面以外のファイルに指定することです。通常、入力はキーボードから、出力は画面となっています。しかし、ファイルを入力として使いたい場合や、実行結果をファイルに出力したいことがあります。このときに、プログラムを変更せずに実現できる仕組みがリダイレクションです。

まず、出力のリダイレクションですが、これはコマンドの後に「> ファイル名」を付けることで実現できます。たとえば「cal」と入力すると、画面に今月のカレンダーが表示されます。ここで、「cal > cal.dat」と入力すると、カレントディレクトリに cal.dat というファイルが作成され、その中身が今月のカレンダーとなっています。

「>」は上書きで結果を書き出すので、同じファイルが存在するときには、前の内容が消えてしまいます。リダイレクションでは、追加の書き込み「>>」が用意されており、これを用いるともとの内容の後に追加して書き込まれます。

それでは、「cal > test.dat」「ls > test.dat」と「cal >> test2.dat」「ls >> test2.dat」を実行して、違いを確認してください。

入力をファイルとする場合には引数として指定できることが多いので、入力のリダイレクションを利用することは少ないかもしれません。入力のリダイレクションは「< ファイル名」で指定されます。たとえば、ホームディレクトリで「cat < .cshrc」とすると、.cshrc の内容が表示されます。これは、「cat .cshrc」と同じ動作です。

3.2 パイプ

パイプとは、Linux のコマンドを組み合わせて使うときに利用するもので、前のコマンドの結果を次のコマンドの入力として利用することができます。このパイプのおかげで、単純な機能しか持たないコマンドを駆使して、複雑な処理を行うことが可能となります。いくつか例を挙げます。

まず、コマンドの出力が長くて一画面に収まらない場合には、more と組み合わせることで一画面ずつ表示させることができます。「ls -l /etc」と入力すると、ディレクトリ etc にあるファイルの一覧が表示されます。しかし、etc には大量のファイルが存在しているため、すべてのファイルを一画面で表示することができず、画面から消えてしまいます。ここで、「ls -l /etc | more」と入力すると、一画面ずつ表示されます。

他の例として、ディレクトリ/etc にあるファイルを名前の順に並べることを考えます。Linux では、ソートするコマンドとして、「sort」が用意されています（sort については、またあとの授業で説明をします)。使い方は、「sort ファイル名」です。ここで、ソートしたいのはディレクトリ/etc にあるファイルなので、ファイルの一覧を入手するために、「ls」を使います。これだけのコマンドが分かれば、ls の実行結果を sort で並べ替えれば良いので、「ls /etc | sort」とします。実行してみると、きちんと並び替えられていることが分かります。この場合も、ファイルが多すぎて画面に収まりきらないので、一画面ずつ見るには、「ls /etc | sort | more」とすれば大丈夫です。

このように Linux のコマンドはたいていパイプにより組み合わせることができます。希望の処理にあったコマンドがないときは、パイプでコマンドを組み合わせて実現することが必要です。

それでは、つぎの問題を考えてみましょう。まず、ディレクトリ/etc にあるファイルはいくつか数えてみましょう。ファイルの行数、単語数、文字数を数えるコマンドとして「wc」があります（wc については、またあとの授業で説明をします)。これを使って自動的にファイルの数を数えてみましょう。