

UNIXのコマンドとコンパイル・実行

山本昌志*

2005年4月14日

1 UNIXのディレクトリー構造

本授業で使うUNIX(Linux)のディレクトリー構造は、図1のようになっている。これは、木構造と呼ばれる階層構造で、図を反対にすれば、ちょうど木の幹と枝のようになっていることが分かる。木は根(root)が必ずあり、そこから、枝が延び、節で分かれ、末端には葉がる。「/」ルートディレクトリ(記号としては「スラッシュ」と発音)が根に相当し、その他のディレクトリが節に、ファイルが葉に相当する。ディレクトリ(Directory)はWindowsのフォルダに相当する物で、他のファイルあるいはディレクトリを格納できる。ディレクトリを用いる事により、種類別にファイルを整理する事ができるので便利である。

UNIXのディレクトリーやファイルの取り扱いについては、以下の特徴がある。

- UNIXのハードディスク¹のファイル構造は、図1のようにツリー(木)構造と呼ばれる階層構造になっている。それは、ファイルとディレクトリーから構成される。
- ハードディスクなどに記録されたデータのまとまりをファイルと言う。コンピュータが実行することができる命令の集合であるプログラムファイルと、コンピュータの利用者が作成した情報を記録しておくデータファイルがある。
- ファイルを分類・整理するための保管場所をディレクトリー²と言う。関連する複数のファイルをまとめて一つのディレクトリーに入れることにより、効率的に記憶装置を管理することができる。ディレクトリーの中にさらにディレクトリーを作成することもでき、階層構造によって細かい分類を表現することもできる。
- ディレクトリーには、以下のように表現されるものがある。
 - 今、自分が居るディレクトリーを、カレントディレクトリーと言う。³カレントディレクトリーを明示したい場合は、1つのピリオド「.」で表します。
 - カレントディレクトリーの1つ上のディレクトリーを親ディレクトリーと言う。2つのピリオド「..」が、親ディレクトリーを表す。

*国立秋田工業高等専門学校 電気工学科

¹実際はハードディスクに限らず、CD-ROMやフロッピーディスクも、このツリー構造に含まれる。

²Macやwindowsではフォルダーと呼ぶ

³ワーキングディレクトリーと言うこともある。

- カレントディレクトリーの1つ下のディレクトリーをサブディレクトリーと言う。複数存在することが可能なので、それぞれの名前で表すしかない。
- ユーザー各個人が使用 (読み、書き、実行) を許されている最上位のディレクトリーをホームディレクトリーと言う。
- ファイルやフォルダの所在を示すものをパスと言う。ファイルやフォルダのハードディスクでの住所みたいなものである。それは、ディレクトリー名を書き並べることにより表すことができ、その区切りには「/」(スラッシュ)を使います。
- パスの表し方は、2通りある。最上位のルートディレクトリー「/」から表す絶対パスと、カレントディレクトリーから表す相対パスである。例えば、図1の yamamoto をカレントディレクトリーとし、hello.c のパスは、

絶対パス /home/user/work/hello.c

相対パス ../e99999/work/hello.c

となる。

また、図1には、それを扱うためのコマンドも書いてある。コマンドについては、次に勉強する。

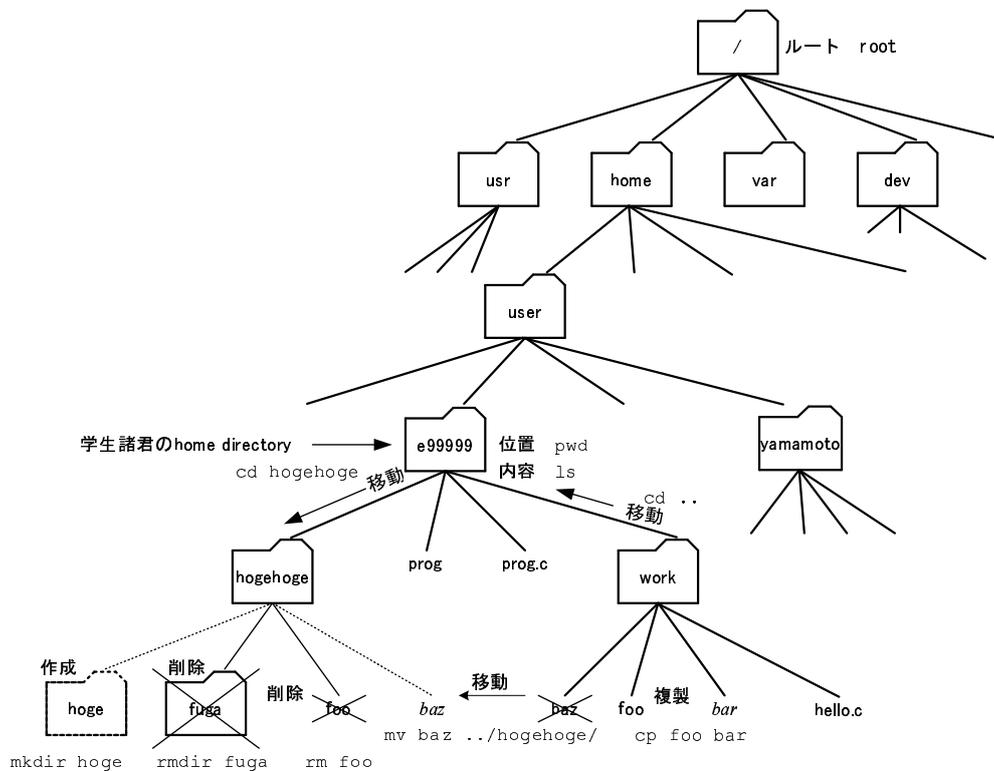


図1: UNIXのファイル構造とそれを扱うコマンド

2 UNIXのコマンド

2.1 使用頻度の高いコマンド

使用頻度の多いUNIXコマンドを以下に示す。これらのコマンドは、すべてターミナルから打ち込み、コンピュータに命令を与える。他に、ファイルマネージャーもあるので、組み合わせて使って欲しい。

表 1: 共振回路の実験に使う機器

コマンド	機能
man	コマンドの動作の説明
pwd	現ディレクトリーの表示
ls	ファイルとディレクトリーの表示
cd	ワーキングディレクトリーの移動
mkdir	ディレクトリーの作成
rmdir	ディレクトリーの削除
cp	ファイルやディレクトリーの複製
mv	名前変更や移動
rm	ファイルの削除
cat	ファイルの表示や連結
less	ファイルの内容を一画面単位で出力
man	コマンドのオンラインマニュアル
↑又は↓	history 以前のコマンドの表示 編集可能
[Ctrl]+c	プロセスの強制終了
[Tab]	補完機能

2.2 コマンドの動作

2.2.1 コマンドの動作を調べる

2.2.2 現ディレクトリーの表示

自分が、今どこにいるか調べるコマンドである。

コマンド	pwd
語源	print working directory
機能	現在のワーキングディレクトリ(カレントディレクトリ)を絶対パス名で表示する。

2.2.3 ファイルとディレクトリーの表示

自分が、今いるディレクトリーの中にあるファイルやディレクトリーを調べるときに使うコマンドである。

コマンド	ls
語源	list
機能	現在のワーキングディレクトリのファイルやディレクトリーの情報を表示する。
形式	ls [-adFgilostux] [<i>filename</i> ...]
オプション	
なし	ファイル名のみ並べて出力する。
-a	.(ドット)で始まる隠しファイルも含めて、すべて出力する。
-l	ファイルの詳細管理情報をロング形式で出力する。
-d	<i>filename</i> がディレクトリの時、その名前のみ表示する。
-F	ファイルの種類を記号で表示する。 < 記号の意味 > 表示なし プレーンなデータファイル / ディレクトリファイル * 実行可能ファイル
-I	inode 番号を表示する。
-R	ディレクトリの階層構造を表示する。
-s	ファイルのサイズをブロック単位で表示する。
-t	最終更新時刻の新しいものから順に表示する。
-u	1 オプションとの併用時、最終更新時刻の代わりに最終アクセス時刻を表示する。
-o	1 オプションと同じだが、グループ名を表示しない。
-x	ファイル名を横に並べて出力する。

2.2.4 カレントディレクトリーの変更

今いるディレクトリーから他のディレクトリーに移るときに使うコマンドである。

コマンド	cd
語源	change directory
機能	<i>directory</i> で指定されたディレクトリに、ワーキングディレクトリを変更する。 <i>directory</i> を指定しない場合は、ログインディレクトリに移動する。
形式	cd <i>directory</i> 指定されたディレクトリーに移動 cd ログインディレクトリーに移動 cd .. 親ディレクトリーに移動

2.2.5 ディレクトリーの作成

ディレクトリーを作成する。ディレクトリー自身を表す「.」と、親ディレクトリーを表す「..」の2つが、自動的に作成される。

コマンド	mkdir
語源	make dir irectory
形式	mkdir <i>directory</i> 指定されたディレクトリーを作成

2.2.6 ディレクトリーの削除

ディレクトリーを削除する。

コマンド	rmdir
語源	r move dir irectory
形式	rmdir <i>directory</i> 指定されたディレクトリーを削除
注意	削除するディレクトリーは空でなければならない。ディレクトリー配下にファイルがある場合、配下のファイルごと削除するには rm コマンドを使う。いずれの場合も mkdir コマンド同様、親ディレクトリーに書き込み権が必要。

2.2.7 ファイルやディレクトリーのコピー

filename1 を *filename2* にコピーする。*directory2* 配下に *directory1* をサブディレクトリーとして、配下のファイルごとコピーする。各 *filename*(サブディレクトリー指定可)を、最後に指定した *directory* 配下にコピーする。

コマンド	cp
語源	c opy
形式	cp [-ip] <i>filename1 filename2</i> cp -r[-ip] <i>directory1 directory2</i> cp -r[-ipr] <i>filename... directory</i>

オプション

- I コピー先ファイルが既存の場合、置き換えを行うかどうか確認してくる。(置き換える場合=y、置き換えない場合=nと入力する)
- p 内容だけでなく、最終修正時刻・アクセス許可もコピーする。
- r ディレクトリー配下のファイルごとコピーする。(ファイルがサブディレクトリーの場合は、その配下のファイルごとコピーする。)

2.2.8

時間が無くて間に合わなかった。web を見てください。

3 プログラムの作成と実行

それでは、実際に、プログラムを作成して、実行させて見ましょう。初心者が最初に作るものとして、最も有名な 'Hello World' というプログラムを作ります。

1. ターミナルの起動

- 端末のアイコンをクリックして、ターミナルを立ち上げる。

2. 作業ディレクトリーの作成と移動

- 「mkdir hello」とタイプして、作業用ディレクトリー hello を作る。
- 作業用ディレクトリーができているか、「ls」コマンドで確認します。すると「hello」が表示される。
- 「cd hello」とタイプして、作業用ディレクトリーに移動する。
- 「pwd」コマンドで、作業用ディレクトリーに移れたことを確認する。

3. エディターの起動

- 「emacs hello_world.c&」とタイプする。
- すると、プログラムのソースを書くウインドウが現れる。

4. プログラムの記述

- エディターのウインドウに以下のプログラムを書きましょう。

```
#include <stdio.h>

int main(){

    printf("Hello World !!\n");

    return 0;
}
```

5. プログラムの保管

- プログラムを書き終わったならば、[file] メニューの [Save(current buffer)] を選択する。あるいは、フロッピーディスクアイコンをクリックして、ソースファイルを保管する。
- ターミナル上で「ls」コマンドを打ち、ソースファイルが保管されていることを確認する。

6. コンパイル

- ターミナル上で「gcc -o aisatsu hello_world.c」と打ち込み、先ほど作成したソースファイルをコンパイルする。
- もし、コンパイルエラーが発生したら、ソースファイルを修正する。

- 実行ファイルができているか、「ls」コマンドで確認する。

7. 実行

- ターミナル上で「./aisatsu」と打ち込み、プログラムを実行させます。
- 'Hello World' と表示されれば、プログラムは動作は完璧です。