

開発環境としてのLinuxコマンドライン

今回の特集ではLinuxのコマンドラインの使い方について取り上げます。コマンドラインは、黒い画面に文字を入力して、その結果が文字で出力されます。コマンドラインでの操作方法さえ覚えれば、Linuxのほとんどの機能が使いこなせます。

● 最新のGUI開発環境！でも中身は…

最近のソフトウェア開発環境はGUI(Graphical User Interface)による操作が当たり前です。プログラムの入力こそキーボードから行いますが、それ以降のコンパイルやデバッグ時のブレークポイントの設定など、マウスでクリックするだけで完了します。

しかし、マウスのボタンをクリックするだけでブレークポイントの設定…などといっても、実はデバッガの中身まではGUIではありません。これはどういうことか。たとえば最新のGUI開発環境といって誰もが納得するであろう、iPhone向け開発環境のXcode(図1)の画面を見てください。カッコイイ画面で今時の開発環境ですが、よく見ると…

(gdb)

まごうことなきGDB(GNU Debbuger)のプロンプトです。このプロンプトが表示されたら、キーボードからcやpなどのアルファベットのコマンドを入力してコンピュータに指示を出す、これが昔ながらのコマンドラインによるインターフェース：CUI(Character-based User Interface：文字によるユーザ・インターフェース)なのです。このことからわかるように、実はXcodeは内部ではコマンドライン用デバッガを呼び出しているラップ(続にいう「ガワ」)なのです。

ほかにも、組み込み向けツールとしてはルネサス エレクトロ

ニクスの統合開発環境HEWがあります。これもボタン一発でコンパイルできますが、コンパイラ本体は¥Program Files¥Renasas¥Hew¥Tools¥Renasas¥Sh¥9_0_2¥bin¥にある注1コマンドライン向けCコンパイラshc.exeです。

今ではほとんどの開発環境がGUIベースになっているため、コマンドラインから直接コンパイラやデバッガを呼び出すことは少ないのですが、このようなカラクリを知っていると、トラブルが起きたときの対処が容易になります。また、一部にコマンドライン版しか提供されないツールがあった場合、それらを自力でGUIのツールから呼び出せるようにする必要があります(特に変換ツールや転送ツールなどの周辺ツール)。もしくは、それらを手動でコマンドラインから呼び出します。現在ではコマンドラインの知識は「無人島で火をおこす」くらいに^{すた}廃れた知識と思われがちですが、往々にして無人島並みの経験を^{すた}するの^{すた}が組み込みシステム開発だったりします注2。

基本はCUIで、それをGUIから呼び出しているということ意識するとGUI開発環境への理解も深まるでしょう。

● マウスを動かすより速い！

コマンドラインによる操作を習得するポジティブな動機としては、マウスを動かすより操作が速いということがあります。たとえば、ホーム・ディレクトリ(WindowsでいうところのMy Documents)の内容を見る場合、キーボードから手を離してマウスを握り、画面を見ながらフォルダ・アイコンをクリックしてたどっていくことになります。

しかしコマンドラインなら

\$ ls ~ (\$はプロンプト)

で一発です。しかしもっと深いディレクトリの場合、たとえば/usr/local/sbin/のような長い名前を指定するのは面倒そうに思えます。しかしコマンドラインにはたいがい補完機能が備わっています。これは「候補の中から一致するものを自動入力してくれる」機能です。頭文字を何文字か入れてTABキーを押すと、残りの部分を自動で入れてくれます。先の例では、

/u ←ここでTABを押す

/usr/ ←自動的に「sr/」が入力される

注1：バージョンによって場所は異なる。

注2：パソコン向けの開発環境はユーザ数が大変多く、バグも洗い出されているので、開発環境自体のバグを疑う必要はほとんどない。しかし組み込み向けの開発環境はユーザ数が少ない上にアーキテクチャも多種にわたり、しかも新しいアーキテクチャがどんどん登場するので潜在的にバグが多いといえるし、実際、著名なCPU向けコンパイラでもバグがちらほら散見される。



図1 Xcodeの画面

のように少ないキー入力ですぐにディレクトリをたどれます。ディレクトリの名前がうろ覚えでも、頭文字さえ覚えていれば目的のディレクトリにたどりつきます。頭文字すら覚えていない場合でも、TABを押せばすべての候補を列挙してくれます。

現在は、GUIでもキーボードのよさを取り入れてショートカット・キーでたいの動作を行えます。ほとんどの操作をキーボードで行う上級者もいるくらいです。GUIなのにキーボード操作というのも本末転倒という気もしますが…注3。

さらにホーム・ポジションからあまり指を動かさずに操作できることも利点の一つです。カーソルを移動するために右手を浮かせてカーソル・キーまで手を伸ばすのは結構な労力です。1日に何百回もこの操作を行うのは疲れます。そこでUNIXのコマンドラインではたいの操作でControlキーを用いた操作が行えます。Controlキーとp(previous; 1行上), n(next; 1行下), f(forward; 1文字先, 右に), b(backward; 1文字後, 左に)を組み合わせることでほとんど手を動かさずにカーソル操作ができます。もっとも、この操作はControlキーがAの左にあるキーボード(写真1)などでないと使いづらいため、そのようなキーボードを使うか、ControlキーとCapsLockキーを入れ替えるソフトウェアが必要になります。ちょっと気が利いたOSなら、ControlキーとCapsLockキーの入れ替えは標準でサポートしています。

● 覚えることが少ない!

今回の特集ではLinuxのコマンドラインを取り上げます。Linuxとは…については第1章に譲るとして、その元となったUNIXの格言としてKISSの法則: Keep It Simple, Stupid(馬鹿みたいに単純にしておけ)というものがあります。これは複雑なプログラムは作るのが大変でバグが多いし、使う方も覚えるのが大変なので、単純で単機能なプログラムをたくさん作って、それを組み合わせて使うというUNIXの基本思想のことをいっています。

典型的なのがsortコマンドです。これは入力されたテキストをアルファベット順にソートして(並べ替えて)出力します。これをlsやcatコマンドと組み合わせて使えば、

```
$ ls | sort ←現在のディレクトリをソートして出力
```

```
$ cat foo.txt | sort ←foo.txtをソートして出力
```

のように使えます。もしsortコマンドがない場合、lsにソート機能を付けたり、catにソート機能を付けておく必要があるため作る方は大変です。使う方もlsのソート機能を使う方法やcatのソート機能を使う方法を調べる必要があります。しかしsortコマンドの使い方を覚えれば、ソートが必要なあらゆる局面で使えるのです。一つ覚えればさまざまに応用できるのがUNIXのコマンドラインのよいところです。

注3: まあ、キーボード操作のGUIというのも存在するので、必ずしもマウス=GUIというわけではないが。



◀ 写真1
ControlキーがAの左にあるキーボード

▶ 写真2
書籍『SUN システム管理』

下山 智明, 城谷 洋司著, アスキー刊



● 20年前の本が今でも役立つ!

写真2は名著として名高い『SUN システム管理』(下山 智明, 城谷 洋司著, アスキー刊)です。20年前に刊行されたUNIXの書籍ですが、2010年の現在見ても驚くほど現在に通用します。もちろん細かい部分での差異はありますが、UNIXのコマンドラインの使い方はほとんど同じです。ls, cpなどの基本コマンドはもちろん、/etcを見ると今も見かけるファイル名が並んでいます。ファイルの中身だけでなく、コマンドラインからコマンドを入力して行って操作する方法は、今回の特集とあまり変わりません。流行り廃りの激しいコンピュータ界で、今でも通用する20年前の書籍がどれほどあるでしょう?

UNIXは1969年に米国AT&Tで開発されて以来約40年、さまざまな場所で使われてきました。最初は大学のコンピュータ室に鎮座していたUNIXマシンも、近年では組み込み機器で動作するようになり、その活躍範囲を広げています。特にフリー・ソフトウェアとして公開されたLinuxが登場して以来、さまざまな分野で使われるようになりました。サーバのOSやルータなどのネットワーク機器をはじめ、携帯電話、ハード・ディスク・レコーダ、デジタル・サイネージ…さまざまな場所で使われています。

UNIXは豊富なソフトウェア資産や人的資産(UNIXを扱える人が多い)、無料で手に入る実装、移植が容易、今時のハードウェアなら問題なく動作することもあり、これからもずっと使われ続けるでしょう。先のことはわかりませんが、たぶん我々の寿命より長く生き続けると思います。ということは、今覚えたUNIXの知識は死ぬまで使えるでしょう。ただし、UNIXはどんどん進化するので、新しい部分はキャッチアップしていく必要があります。それでも基本部分は一緒です。