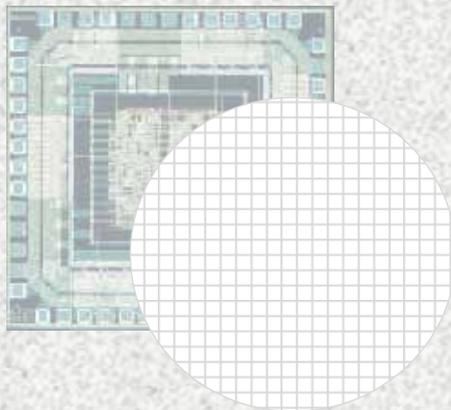


第7章 開発環境



第7章 開発環境

1 . UNIX/LINUX

1 - 1 はじめに

本節ではUNIX/LINUX(以下、UNIX系)OSで使用されるCUI(Character User Interface:文字による入出力インターフェイス)シェルのコマンドの一般ユーザにおける使用方法と簡単なシェルスクリプトを紹介する。なお、ここで紹介するコマンドは以下の環境で動作の確認を行った。

【使用環境】:

- ・ FreeBSD7.0 (BSD系) と Fedora9 (Redhat系)
- ・ tcsh (Ver.6.15.00) と bash (Ver.3.2.25)

なお、UNIX系OS搭載ワークステーション(以下EWS)を共同で使用する一般ユーザを想定している。そのため、管理コマンド(/sbin、/usr/sbin等に格納されているコマンド)は紹介しない。

1 - 2 UNIX系のディレクトリ構成

UNIX系のディレクトリ(Windowsのフォルダに相当)は図1.1のような構造となっている。最上位(図1.1の"/)を根(root)として枝葉が伸びるようにディレクトリとファイルが配置されている為、一般的にルートファイルシステムと呼ばれている。ユーザには/home(OSによっては/user)にユーザ名で作成されたホームディレクトリが用意され、ホームディレクトリ以下で作業を行う。

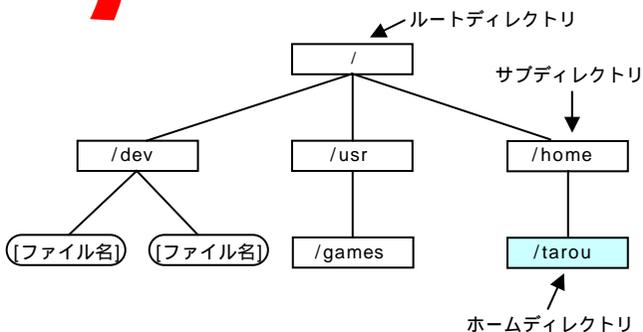


図1.1 UNIX系のディレクトリ構成概念図

1 - 3 シェルについて

シェルとは OS とユーザ間の処理を仲介するインターフェイスプログラムのことを指す。ユーザはキーボードやマウス等の入力装置から、シェルコマンドを発行し、OS に指示を出す。OS は受け取った指示を実行し、その結果はディスプレイやプリンタ等の出力装置に出力される。

Windows に代表される PC のデスクトップ画面も、GUI シェルと捉えることが出来、これもシェルの一種である。また、UNIX 系にも X Window System と呼ばれる GUI 環境が用意されている場合が多い (Gnome や KDE などが有名)。本節は特に言及がない場合、シェルは CUI シェルとして扱う。

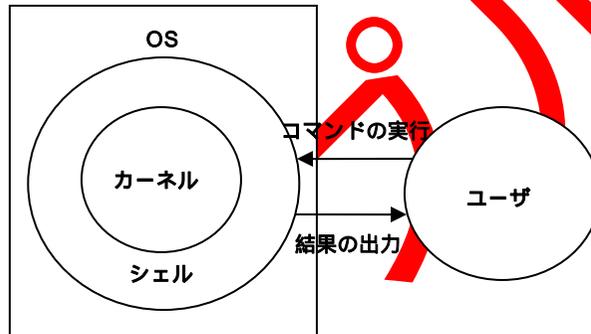


図 1 . 2 カーネルとシェルの概念図

1 - 4 シェルの種類

UNIX 系で使用されるシェルには大きく分けて sh 系と csh 系に分類される。

sh 系 : sh , bash , ksh , zsh

csh 系 : csh , tcsh

新規でユーザを作成した場合、特に指定がなければ、csh または tcsh、もしくは bash が標準のログインシェル (端末エミュレータ起動時に起動されるシェル) として登録される (Linux 系では bash が、BSD 系、Solaris 系では csh もしくは tcsh が採用されている)。なお、シェルはユーザが任意に入れ替え可能である (管理者にしか権限がない場合もある)。

sh 系、csh 系でシェルの操作方法には大きな違いはないが、シェル独自の機能 (コマンドの補完機能、ヒストリ機能)、コマンドがあり、どのシェルがよいかは一概には決めることが出来ない。しか

し、後述のシェルスクリプトを作成する場合、csh 系では関数を作成することが出来ないため、シェルスクリプトを作成するには注意が必要である。

1 - 5 コマンドの確認方法

一口に UNIX 系といっても、そのディレクトリ構成、コマンド構成は OS の系統、インストール時の構成、管理者の方針によって異なる。また、同じコマンドでも OS が異なれば、コマンドの挙動が違うか、サポートしているオプション等がまったく異なる場合がある。

例として、ls コマンドでファイルの種類によって表示色を変更するオプションがあるが、Linux 系では `--color`、BSD 系では `-G` と OS によって異なる。また、cp コマンドでシンボリックリンクファイル（Windows のショートカットに相当）をコピーした場合、Linux 系ではリンクファイルをそのままコピーするのに対して、Solaris だと、リンクファイルをそのままコピーせずに、リンク先のファイルをコピーする等、同一コマンドで挙動が違う場合もある。

コマンドの詳細は“`man [コマンド名]`”で調べることが出来るので、コマンドの使い方が分からない場合や、オプションを調べたい場合はまず `man` を実行することを推奨する。

man ファイルが存在しない場合はシステム管理者に確認する。

1 - 6 設定ファイルについて

`alias` コマンドや `setenv`、`export` コマンドで設定したコマンドの別名や環境変数などは、そのシェルが起動している間だけ有効であり、次回起動時には初期状態に戻っている。

シェル起動時から反映させたい設定がある場合、その設定コマンドをシェル毎の設定ファイル（[ユーザ名]/`.cshrc` や [ユーザ名]/`.bashrc` 等）に記述しておく、シェルは起動時にまず設定ファイルの内容を実行する。設定ファイルはシェルスクリプトとして扱われるので、起動時に実行したい処理を設定ファイル内に記述しておくのも有効である。

6 . 1 .cshrc と .login の役割

2つのファイルには、それぞれ異なる役割がある。`.login` はログインのためのファイル、`.cshrc` はCシェルの初期設定用のファイルである。

- `.login` は、ログインシェルによって、ログイン時だけその内容が実行される
- `.cshrc` は、Cシェルが起動されるたびに実行される（ログイン時だけとは限らない）