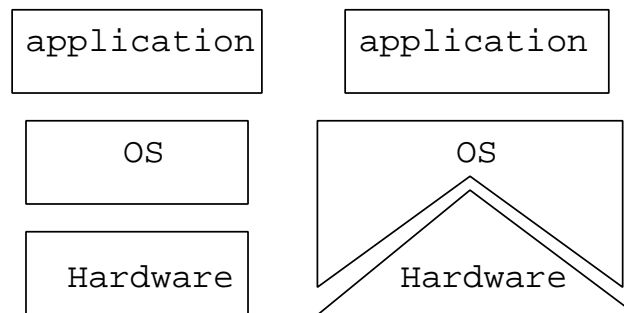


1 Mac OS X と UNIX

1.1 オペレーティングシステム (OS)

オペレーティングシステム Operating System(OS) とは、中央演算装置 CPU、メモリ memory、キーボード keyboard、ディスプレイ display、マウス mouse、プリンタ printer、ネットワーク装置 network device などのコンピュータ資源 computer resource 全体を管理し、利用者に対してコンピュータを操作する際の基本的操作環境を提供するソフトウェアである¹。OS はハードウェアとアプリケーションソフトウェア application software (ワードプロセッサや表計算ソフト、インターネットブラウザなど) との間に介在し、ユーザがコンピュータを用いる際の仲立ちをするソフトウェアといえることができる。また、OS は、多くのアプリケーションソフトから共通して利用される基本的機能を提供するソフトウェアでもある²。

図 1: OS の概念図、ハードウェアの違いを吸収して統一された環境を提供する user user



オペレーティングシステムは基本ソフトウェアとも呼ばれる。パソコンの主な OS には、Microsoft 社の Windows、Apple 社の Mac OS X、UNIX 系 OS(Linux, FreeBSD など) がある。東女のコンピュータシステムの OS は、Mac OS X である。

東京女子大学の情報処理教室で一般利用者が利用できるコンピュータの OS は、Mac OS X(マックオーエステンと発音する) である。

1.2 UNIX

UNIX(ユニックスと発音する) とは OS の一種である。UNIX は、一台のパーソナルユースのコンピュータから巨大なコンピュータシステムまで様々な分野で広く使われている OS である。

UNIX はインテル社 CPU を始めとする様々な CPU 上で動作し、マルチユーザ、マルチタスクの環境を提供する OS である。マルチユーザとは同時に複数の利用者が利用できることを指し、マルチタスクとは同時に複数のプログラムが実行できることを指す。ユーザはコンピュータを利用を開始する際に、そのコンピュータにログイン login するが UNIX では一台のマシンに同時に複数のユーザがログインすることができる。またユーザは同時に複数のプログラムを実行することができる (マルチタスク)。

¹OS が共通であれば、ハードウェア hardware が異なっても基本的操作は共通になる。

²ソフトウェアの開発者は、OS の提供する機能を利用することによって、開発の手間を省くことができ、アプリケーションの操作性を統一することができる。

1.3 Mac OS X と UNIX

Mac OS X は Apple Computer によって作られた OS である。Mac OS の前身である NeXTStep や OpenStep のように、UNIX の一種である FreeBSD をベースにしている。Mac OS X は Quartz と呼ばれる画像表示システムを有しており、そのルックアンドフィール (見栄え) は Aqua と呼ばれている。

Mac OS は、一貫したデザインと操作方法が統一された GUI (Graphical User Interface: マウスを使った入力) によって操作することができ、初心者にも使いやすい man machine interface が提供されている。使いやすさと UNIX の堅固さが共存しているところに他の UNIX 系 OS とは異なる特徴がある。2003 年以降多くの大学で教育研究機関で Mac OS X が導入されている。

Darwin のインターフェースは CUI (Character User Interface: キーボードから文字の入力のみ) であるが、Quartz などの描画エンジンを使って Aqua と呼ばれる独自の美しいユーザーインターフェースを採用し、GUI が可能になっている。

コンピュータ I では GUI 操作の基本を学んだが、コンピュータ IID では CUI 操作の基本を学ぶ。

1.4 ターミナルエミュレータ Terminal emulator

[Finder] → [アプリケーション] → [ユーティリティ]

とするとターミナルエミュレータのアイコン (コンピュータのディスプレイの絵) が表示されるので、ターミナルエミュレータのアイコンを Dock に drag and drop する。

Mac OS X は通常 Aqua の interface によりコンピュータを利用するが、ターミナルエミュレータから操作すると finder などでは操作できない操作をすることができる。

1.5 プロセスの一覧とプロセスの強制終了

Aqua では、コマンドキー、オプションキー、エスケープキーの 3 つを同時に押すと、「アプリケーションの強制終了」の窓が開き表示されたアプリケーションについては強制終了できる。

ターミナルエミュレータでは、

```
$ ps ax
```

とするとプロセスの一覧が表示でき、さらに

```
$ kill そのプロセスの ID
```

とするとそのプログラムが強制終了される。(そのプロセス ID のところには実際には数字が入る)

演習 1.1 1. UNIX の歴史について調べよ。

2. 今自分が操作しているパソコンで幾つのタスクが走っているか。

3. 自宅の Windows では同時に幾つのタスクが走っているか調べる方法を記せ。

4. *UNIX* におけるプロセスとは何か？
5. *UNIX* におけるデーモンとは何か？
6. *CPU* とは何か？
7. *TCP/IP* とは何か？
8. *LDAP* とは何か？
9. *GNU* とは何か？

2 ファイルシステム

UNIX はファイルという概念でコンピュータ資源を管理している。一般にファイルとはコンピュータ上のハードディスク、DVD ディスク、USB のフラッシュメモリ上などの記憶媒体に格納されたひとまとまりの情報をさす言葉として定着してしまった感があるが、ディスプレイ、キーボード、プリンタなどの周辺機器もファイルである。UNIX では、周辺機器が、ファイルシステム上に通常のファイルのような形で提供されるデバイスドライバのインタフェースによって表現されている。このことにより、ソフトウェアが入出力システムコールを通してデバイスドライバを使うことができ、作業が単純化される。すなわちファイルによってプリンタなどの周辺機器への単純なインタフェースを提供されている。

ファイルは名前が付けられ、整理しやすいように目的や性質により関連するファイルを1つのフォルダ (folder= 書類挟み) またはディレクトリ (directory= 住所氏名録、登録簿) と呼ばれる入れ物に入れる。利用者はファイルを利用するとき、その名前だけを指定すればよく、あとは OS が登録簿を参照してファイルが記憶されている記憶媒体を見つけてくれる。

Windows XP や Mac OS X ではフォルダ、UNIX ではディレクトリと呼ばれる。UNIX では、ディレクトリはファイル名の表を格納したファイルである。この表によって、階層構造のファイルシステムが実現される。

2.1 ホームディレクトリ

UNIX では、ユーザごとに1つずつホームディレクトリというそのユーザに専用のディレクトリが割り当てられている。

Mac OS X ではターミナルを起動すると、ユーザーはホームディレクトリにいることになる。Finder を起動するとサイドバーにある家のアイコンの横にログイン名が書いてある。家のアイコンをダブルクリックしたときに表示されるフォルダ (ホームフォルダ) とホームディレクトリは同じものである。

自分のホームディレクトリの中に自由にディレクトリ (子ディレクトリ) を作ったりファイルを管理することができるが、他人のホームディレクトリの中にディレクトリを作ったりファイルをおくことはできない。

自分のホームディレクトリは記号 “~” (チルダ) で表す。

ログイン名 asakawa のホームディレクトリを ~asakawa で表す。ログイン名 asakawa のユーザーにとっては、~ と ~asakawa は同一のディレクトリを指す。³

2.2 カレントディレクトリ

現在作業しているディレクトリをカレントディレクトリまたはワーキングディレクトリという。記号 “.” (ドット) で表わす。

ターミナルエミュレータを立ちあげた直後は、ホームディレクトリがカレントディレクトリである。

複数のターミナルを立ち上げると、ターミナルごとにカレントディレクトリが存在する。

³Windows 95/98/Me は一人の利用者が使うものとの前提 (シングルユーザ用) なので、ホームディレクトリに相当するものはない。Windows XP, Windows Vista はマルチユーザ対応なので、ホームディレクトリに相当するフォルダは (ローカルディスクを C ドライブとすると) C:¥Documents and Settings¥利用者名である。

2.3 親ディレクトリ

ディレクトリの中にディレクトリを含むことができる。あるディレクトリに対しそれを含むディレクトリを親ディレクトリという。親ディレクトリは“..”(ドット2つ)で表わす。

ディレクトリが含むディレクトリを子ディレクトリまたはサブディレクトリという。親ディレクトリはただ1つであるが、子ディレクトリは複数存在してなくてもよい。

2.4 ルートディレクトリ

すべてのディレクトリの「祖先」となる特別なディレクトリが存在する。それを、ルートディレクトリといい、記号“/”(スラッシュまたはスラと読む)で表す。

ディレクトリとサブディレクトリの区切り、ディレクトリとそれに含まれるファイルとの区切りを“/”で表わす。

演習 2.1 ターミナルエミュレータから `ls` コマンドのオプションを `-F` 引数を `/`(スラッシュ)として実行せよ。

ここで、名前の最後に `/`(スラッシュ) がついたものがフォルダ、`@`(アットマーク) のついたものがシンボリックリンク⁴、それら以外はファイルである。

3 基本コマンド

3.1 man

コマンドについて詳しい情報を知りたいときは、`man` コマンドを使う。`man` コマンドはオンラインマニュアルを整形し出力するコマンドである。

```
$ man コマンド名
```

MANual

- スペースキーまたは `f` で次の画面。(forward)
- `b` で前の画面 (backward)
- `q` で終了 (quit)

ASCII コードを表示する `man ascii` も `man` コマンドを使っている。

演習 3.1 `man ls` を実行せよ

3.2 ls

```
$ ls [-オプション] [ファイル]
```

LiSt

`ls` コマンドはファイルまたはディレクトリを引数(ひきすう)にとり、そのファイルまたはディレクトリに関連した情報を表示する。

⁴「必要な情報はここではなく、ファイル×××にある」と張り紙をしているようなもので、指し示している対象の名称を変更しても削除してもそのまま残る。(「入門 Unix for Mac OS X」p.103)

演習 3.2 以下のコマンドを実行せよ。どのような表示が得られるか？また、それぞれのオプションの意味を説明せよ。

1. `ls -R`
2. `ls -l`
3. `ls -lR`
4. `ls -F`
5. `ls -a`
6. `ls -la`
7. `ls -lat`
8. `ls .`
9. `ls ..`
10. `type ls`

演習 3.3 Finder でルートフォルダを開き、ターミナルで表示されるルートディレクトリと比較してみよ。

Applications ディレクトリとファインダで表示される「アプリケーション」フォルダは同じものである。

演習 3.4 Finder でアプリケーションフォルダを開き、ターミナルで表示される Applications ディレクトリと比較してみよ

3.3 echo

```
$ echo [-n] [文字列]
```

echo コマンドは引数を画面に表示するユーティリティプログラムである。

演習 3.5 次のコマンドを実行せよ。

1. `echo ‘‘こんにちは’’`
2. `echo ‘‘こんにちは’’ > hoge.txt`
3. `ls`
4. `cat hoge.txt`
5. `rm hoge.txt`
6. `ls`

3.4 cat

```
$ cat [-オプション] [-] [file ...]
```

conCATinate

cat コマンドはファイルを逐次読み込んで、その内容を表示するユーティリティである。

3.5 rm

```
$ rm [-オプション] file ...
```

ReMove

rm コマンドは引数で与えられたファイルまたはディレクトリを削除するユーティリティである。

指定できるオプションには次のものがある。

- d ディレクトリを削除しようと試みる。
- f 強制的に削除する。
- i 削除する前に確認を求める。
- R ファイルの階層構造を再帰的に削除する。
- r R と同じ。
- v 冗長な出力する。

3.6 cd

cd コマンドはカレントディレクトリを変更する。

```
$ cd [directory]
```

Change Directory

演習 3.6 次のコマンドを実行せよ。

1. cd
2. ls
3. cd ..
4. ls
5. cd
6. ls
7. echo abcdefg > a.txt
8. ls
9. cat a.txt
10. cd ..

11. cat hoge.txt
12. ls
13. cd
14. cat hoge.txt
15. cd /tmp
16. ls
17. cd /
18. ls
19. cd /Applications
20. ls
21. cd /bin
22. ls
23. cd

3.7 pwd

pwd コマンドはカレントディレクトリ (現在のワーキングディレクトリ) を表示する。

```
$ pwd [-PL]
```

Print Working Directory

演習 3.7 次のコマンドを実行せよ。

1. cd
2. pwd
3. ls
4. cd ..
5. pwd
6. ls
7. cd
8. pwd
9. ls
10. cd /tmp
11. pwd

12. cd /
13. pwd
14. ls
15. cd /Applications
16. pwd
17. ls
18. cd /bin
19. pwd
20. ls
21. cd

3.8 pushd, popd, dirs

pushd, popd, および dirs コマンドはセットで使われるコマンドである。pushd はカレントディレクトリを移動するがその際、元いたディレクトリを記憶する。反対に popd は記憶したディレクトリに戻る。dirs コマンドは現在記憶しているディレクトリの変遷履歴を表示する。

演習 3.8 次のコマンドを実行せよ。

1. cd
2. pwd
3. dirs
4. ls
5. pushd ..
6. pwd
7. dirs
8. ls
9. pushd /etc
10. pwd
11. dirs
12. ls
13. pushd /tmp
14. pwd
15. ls

16. pushd /
17. pwd
18. dirs
19. popd
20. dirs
21. pwd
22. popd
23. dirs
24. pwd
25. popd
26. dirs
27. pwd

3.9 cp

cp コマンドはファイルをコピーするユーティリティである。第一引数で与えられたソースファイルを第2引数で与えられたターゲットファイル名としてコピーする。

```
$ cp [-オプション] ソース ターゲット
```

CoPy

演習 3.9 1. ワードで abc.docx というファイル名の文書を作れ。

2. cp abc.docx def.docx を実行せよ。

3. def.docx というファイルができていることを確認せよ。

4. ワードで def.docx というファイルが正常に読み込めることを確認せよ。

3.10 mkdir

mkdir コマンドはディレクトリの作成する。

作成するディレクトリ (のパス) 名を *directory* とすると

```
$ mkdir directory
```

MaKe DIRectory