起動と終わり

まず、コンピュータの電源をいれる。
PC は、マザーボードに載っている、ROM(ロム) {(リードオンリーメモリ(Read-Only Memory) - 不揮発性の半導体メモリ及び半導体メモリ以外の読み出し専用媒体。)}
に書きこまれた小さなプログラム BIOS が、コンピュータにつながっている、いろいろなハードウェアの調子をうかがいにチェックします。車で言うところの、試乗前点検をします。

次に BIOS は、IPL(Initial Program Loader)というプログラムを起動して、ハードディスクの先頭にある MBR(Master Boot Record)という領域を読み込む。
MBR の内容からハードディスク内のどの領域(パーティション)から OS を起動するかを選択し、そのパーティションの FreeBSD のシステムの最初(セクタ 0)ブートセクタと呼ばれる領域に記録されたプログラムを呼び出す。
ブートセクタに記録されたプログラムはディスクから OS をメモリに読み込んで起動し、操作可能な状態になる。

---

boot: コンピュータを起動すること。また、人間がコンピュータに電源を投入してから、操作可能な状態になるまでに自動的に行われる一連の処理。

BIOS: Basic Input/Output System (コンピュータに接続されたディスクドライブ、キーボード、ビデオカードなどの周辺機器を制御するプログラム群。これらの機器に対する基本的な入出力手段を OS やアプリケーションソフトに対して提供することからこう呼ばれる。パソコンではマザーボード上に装着されたフラッシュメモリなどの不揮発メモリに記録されている。)

セクタ：円盤(ディスク)状の記憶装置における最小の記録単位。ハードディスクやフロッピーディスクなどのディスクを利用した記憶媒体は、木の年輪のように同心円状の「トラック」に分割される。これをさらに放射状に等分した部分がセクタである

MBR:　パソコンの起動時に最初に読み込まれるハードディスク上の部分(セクタ)。ハードディスクの先頭に置かれ、ハードディスク内に収められたどの OS をどのように起動するかなどの情報が記録されている。パソコンを起動するとまず MBR が読み込まれ、ブートローダと呼ばれるプログラムが動作する。ブートローダはパーティション(ハードディスク内の領域)の位置や大きさを記録したパーティションテーブル読み込み、起動するパーティションのブートセクタと呼ばれる領域を読み込む。ブートセクタにあるプログラムは、そのパーティションに置かれた OS を起動する

ブート

/boot/boot0

FreeBSD のＭＢＲは　/boot/boot0 にある。　これはＭＢＲのコピーで、本物のＭＢＲは　ディスク上の FreeBSD 領域の外に置く必要があります。

boot0 は、512 バイトの大きさで無ければいけないという　制限あります。
FreeBSD の MBR をインストールし、そのうえ複数のＯＳ(オペレーティングシステム)をインストールすると、

```
F1　DOS
F2　Linux
F3　FreeBSD

Default:　F3
```

のような画面がでます。
Windows をインストールすると。今まで在った MBR を Windows の MBR で上書きしてしまうために、既存の MBR を FreeBSD　の MBR で　置き換えたいなら

例えば

fdisk –B -b /boot/boot0 ad0

とコマンドを打てば既存の MBR を FreeBSD　の MBR で　置き換えることができます。
ad0　　は、IDE のハードディスクの　１番目
ad１　　は　IDE のハードディスクの　２番目
ad2　　は、IDE のハードディスクの　３番目
ad(n-1) は　IDE のハードディスクの　ｎ番目
注意することは、n 番目のハードディスクは、　n 番目から１を引いた数字になる。

SCSI は　da0 , da1,　da2　などとなります。

/boot/boot1　・　/boot/boot2

boot1　は boot2　は、ハードディスクの同じ領域上の同一のプログラムの部分部分で
スペースの制約の為に　２つに分割されている。
boot1 と　boot2 は　起動スライス（slice）の起動セクタにある。
起動セクタとは、MBR 上にある　boot0 と同じく　512 バイトの大きさの制限がある。
boo1 は　boot2　を検索し、実行するめ、boot2　のスライスの情報(ディスクラベルに関する情報)
の最低限を持っている。

ディスクラベル：ディスクのコントローラ、ジオメトリ(シリンダ数、ヘッド数、トラックあたりのセクタ数)スライスに関する情報を格納する特殊な領域が確保されています。そのような情報をディスクの「ラベル」と呼びます

boot2 は　ファイルシステム上で　ファイルを見つけ、カーネルやローダを選択するための簡単なインターフェイス(選択肢)を持っている。

カーネル:Kernel とは､階層型に設計されたオペレーティングシステム(OS) の中核となる部分である。システムのリソースを管理し、ハードウェアとソフトウェアコンポーネントのやりとりを管理する。メモリ、CPU、入出力を中心としたハードウェアを抽象化し、ハードウェアとソフトウェアがやり取りできるようにする

ローダ：プログラムを起動するための機能を持ったプログラム

/boot/loader

ローダは 3 段階の起動プロセスの最後です。通常、ファイルシステム上の /boot/loader として存在している。

ローダは、様々なコマンド郡をサポートしたインタプリタにより提供される簡易組み込みコマンド郡を利用する事で、ユーザが利用し易い設定になっている。

インタプリタ：（interpreter）とは、プログラミング言語で書かれたソースコードを逐次解釈しながら実行するソフトウェアである

ローダは初期化のときに、コンソールとディスクの検出を行い、どのディスクから起動しているかを調べる。そして必要な変数を設定してからインタプリンタを起動し、スクリプトコマンドを送ったり、手動でコマンドを入力したりします。

コンソール:　コンピュータを操作するために使う入出力装置のセット。制御卓、操作卓などとも呼ばれる。　ディスプレイなどの表示装置、キーボードなどの入力装置で構成される。

ローダは　次に　/boot/loder.rc　を読み込み、/boot/defaults.conf と、そのマシンにローカルな変数を定義した　boot/loader.conf　を読み込みます。
Loader.rc　は　変数に基づいて､選択されたモジュールとカーネルをロード(プログラムを走らせる)する。

モジュール：機能単位、交換可能な構成部分という意味の英単語。システムへの接合部(インターフェース)が規格化・標準化されていて、容易に追加や削除ができ、ひとまとまりの機能を持った部品のこと。

ローダは　最後に、標準設定で　10 秒のキー入力を待ち　入力がなければ　カーネルを起動する。
入力があれば、簡易コマンド郡が使えるプロンプトが表示され、変数を変えたり、モジュールを
ロードしたり　アンロードしたりして、起動する。

カーネル

FreeBSD のカーネルは、普通　/boot/kernel にある。
/boot/loader に　カーネルを読み込み起動する。
カーネルとは、　OS(オペレーティングシステム)の核となる重要なプログラムで、プログラムとユーザの両方で、ハードウェアへのアクセスを制御しています。
起動して、ハードウェアの検出をしているときに、モニターにメッセージが表示されるが、速くて読めません。そんな時は、ログインして、

dmesg | more

とリターンを押せば

```
Copyright © 1992-2008 The FreeBSD Project.
Copyright © 1979, 1980, 1983, 1986, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California.  All rights reserved.
FreeBSD RELEASE #0: Web Jan 16 04:18:52 UTC 2000
root@dessler.cse.buffalw.ecu:usr



――――――――――――――――――――途中省略――――――――――――――――――――
```

と表示される。

init

FreeBSD を普通に正しく終了（shutdown ）すると、それぞれのディスクに対して、sync と呼ばれるプログラムが実行され、メモリに残っている全てのデータを書き出し、ファイルシステムをアンマウントしてから、各ファイルにクリーンフラグ(メンテナンス不要の証)　を立てます。
途中で、停電などにより電源が切れてしまったりした時には、クリーンフラグが立ちません。
次回の起動のときには、マルチユーザモードになれない事もあります。

カーネルの起動が終ると、init(8)と言う　ユーザプロセスに制御が移る。
/sbin/init にあります。

init　はブート処理の最後に起動され、rc(8)で説明されている自動リブートシーケンスを実行する。
その、処理が終ると、マルチユーザモードになります。
自動リブートシーケンスの実行に失敗すると、init は、スーパユーザが使用するシェルを起動して
シングルユーザモードを実行し、ブートプログラムからのパラメータの指示を受け、一般のデーモンを
起動せずにシングルユーザモードのシェルを起動します。
自動リブートシーケンスの実行に失敗した時は、システムは、メンテナンスのためのモードであるために、
シングルユーザモードのシェルを抜ける為に　(^D を入力して）シングルユーザモードから、マルチユーザモードになります。
これにより、　init は /etc/rc を　ディスクチェック省略をして、実行できます。

fsck　　では,修復できないエラーを検出するとシステムをシングルユーザモードに替えます。
クリーンフラグがたっていなくても　fsck で修復できれば　/etc/fstab のファイルに記述された
ファイルにマウントされる。

自動再起動（automatic reboot）
自動再起動では、システム上で利用できるファイルシステムの一貫性を確認する。
もし、それに問題があり　fsck(8)がその不一致を修復できなければ、管理者に直接対処させるために
init(8)h はシステムをシングルユーザモードへと移行させます。

マルチユーザ：複数のユーザが一つの環境（をあるコンピュータやネットワークを構成するハードウェアやソフトウェアの組み合わせと、それぞれの状態や設定の総体）を共有すること

シングルユーザ：1 台のパソコンを 1 人のユーザーが使用すること。、

f sck：ファイル・システムの検査と修復を行う ファイル

/etc/fstab に記述されているファイルシステムをマウントした後に、init は　次に、　/etc と /etc/defaults　にある、システムの設定スクリプト（rc スクリプト）を読み込み、次に /usr/local/etc/rc.d というディリクトリをチェックする。

スクリプト：(script)機械語への変換作業を省略して簡単に実行できるようにした簡易プログラム。
プログラムはプログラマの書いたソースコードを台本（Script）のように記述するための、簡易的なプログラミング言語である。そのプロセスを自動化して簡単に実行できるようにしたものをスクリプトという。

起動スクリプトを読み込んだら、init　は、コンソール上で　getty プログラムを読み込実行する。
ファイルは /etc/ttys にある。
ｇetty　は　まず、セキュリティーや端末タイプのオプションを制御する。
つぎに、/etc/ttys　に　定義去れている各端末を初期化して、login ファイルを起動します。

login　　logout

起動プロセスが終ると login の文字が　画面左下に　表示されます。

```
login:
```

login:　が表示されたら、ユーザとして、ログイン名を入力して Enter キーを押します。

login: の下に　Password:が表示され　パスワードを入力し Enter キーを押します。
入力したパスワード文字は表示されません。

```
login:　gonbei
Password:
```

ログイン名と　パスワード　が一致すると　画面左下に

```
%
```

プロンプトが表示され、　コマンド入力待ちの状態になります。

ログイン名やパスワードを間違えると 下記の画面がでてきます。

```
Login incorrect
login:
```

この画面に遭遇したら、もう一度　ログイン名とパスワードを入力しなおしましょう。

作業が終ったら、ログアウトを　しておきましょう。
良からぬ、輩が　あなたの情報にいたずらをするかもしれません。

```
exit
```

すると　ログイン画面があらわれます。
これで、一安心ですね。

シャットダウン

FreeBSD での作業が終了し、電源を落とす前にシステムを、停止または再起動するときに
shutdown コマンドを使います。

shutdown　コマンドの使用例

```
% su
Password:
# shutdown –h now
```

システムの停止と再起動は、スーパーユーザだけが、行う事ができます。

まず、su コマンドを実行します。
パスワードをきかれますので
インストールの時に　root　のパスワードを設定しているので、root のパスワードを　入力します。
Enter キーを押して　スーパーユーザーに変身します。

su:すべての権限を持つスーパーユーザー（ルート権限保持者）になるには，「su」コマンドを使う。また，スーパーユーザーになることを「ルート権限を得る」という。コマンド によってはルート権限を得ないと実行できないものもある。
スーパーユーザーになると，プロンプトが「#」に変わる。

次に、
shutdown –h now
と　コマンドを実行した後　システムが停止して、電源をきってもいいと、メッセージが表示されたら電源を切ります。

```
The opearating system has halted.
Please press any key to reboot.
```

再起動の時は
shutdown –h now　の　–h のところを　–r　に変えると　再起動します。

```
% su
Password:
# shutdown –r now
```

7 分後にシステムを再起動する時

```
% su
Password:
# shutdown –r +7
```

2008 年 09 月 12 日 03 時 15 分にシステムを停止、電源を切る。

08　09　12　03　15

```
% su
Password:
# shutdown – p 0809120315
```

shutdown ［オプション］ 時間 ［メッセージ］

オプション

-k　システム停止のメッセージを各プロセスに送り、ユーザにログアウトを促すと同時に以後の新たなログインを禁止する。複数ユーザが利用している時に使う。

-r　システム再起動

-h　システム停止

-p　システム停止　電源をきる。(マシンが自動電源切断でカーネルが電源管理のオプション設定の時)