

FreeBSDカーネルコンフィギュレーション

2002年10月

木村健一郎

なぜカーネルコンフィグレーションを行うか？

- 標準のカーネルでは組み込まれていないデバイスや機能を利用する
- 使用していないデバイスをカーネルからはずす
 - 起動の高速化、消費メモリ量削減
- カーネル内部で持っている情報を変更する

使用するファイル

- `/usr/src/sys/i386/conf/MYKERNEL`
 - コンフィグレーションファイル
 - 標準で備わっているもの(GENERIC)をコピーして作る
- `/usr/src/sys/i386/conf/LINT`
 - コンフィグレーションファイルで指定可能なカーネルオプションの詳細

手順

/usr/src/sys/i386/conf に移ってから行う

1. #cp GENERIC MYKERNEL
2. コンフィグレーションファイル(MYKERNEL)を編集
3. #config MYKERNEL
4. #cd ../../compile/MYKERNEL
5. #make depend
6. #make
7. #make install
8. 再起動

手順(2)

最近はこちらのほうがいいかも。

コンフィグファイルを書いたら、/usr/src で作業

1. `#make KERNCONF=MYKERNEL buildkernel`
2. `#make KERNCONF=MYKERNEL installkernel`
3. `#make install`
4. 再起動

コンフィグレーションファイル・必須オプション

- machine i386
 - マシンのアーキテクチャ
- cpu
 - CPUタイプを指定
 - 現在主流のPentiumPro以降のCPUはI686_CPU
- ident
 - カーネルの識別名
- maxusers
 - 最大ユーザ数
 - プロセスの最大数を決定 $20 + 16 * \text{maxusers}$
 - 4.5R以降なら、0にしたらメモリ量から自動的に計算してくれる

その他のオプション(1)

- `makeoptions DEBUG=-g`
 - カーネル再構築の際にgdbのデバッグオプションをつけて構築する
- `options MATH_EMULATE`
 - i386やi486SXなど浮動小数点演算部を持たないCPUの場合カーネルにシミュレートさせる
 - i486DX以降のCPUであればコメントアウトしてもよい
- Ethernet NICのドライバ
 - 必要なものだけ残す

その他のオプション(2)

- options IPFIREWALL
 - パケットフィルタリングのためのコード(IPFW)をカーネルに組み込む
- options IPFIREWALL_VERBOSE
 - syslogdを用いてパケットのログをとるためのコードを有効にする
- options IPFILTER
 - NATを用いるときに必要?
 - 最近ならNATはIPDIVERで

その他のオプション(3)

一部のデバイスは注意。

- SCSI系のデバイス(umassも含む)

- Device scbus を忘れない

- 一部のNIC

- Device miibus を忘れない

- Fxp,dc,rlなど

カーネル周りはバージョンによって頻繁に変わるので、アップデート時には注意。Make worldしたら
かならず/usr/src/UPDATINGを読む。

カーネルオプション

かなりのカーネルオプションは、再構築なしで設定可能

- Sysctlコマンドを使う

- Man sysctl
- 一部の値は起動してしまったら書き換えできない
- /etc/sysctl.confに書く

デバイスドライバ

一部のデバイスドライバは、実行時に組み込むことが可能

- Man kldload

- カーネルモジュールになってるドライバはこれで読み込んで動くものもある(たとえばpcm)

実習

- カーネルの再構築を手順に従って行ってみてください
 - 事前に自分のマシンのデバイスを調べておく(CPU、NIC、SCSIコントローラーの有無、etc…)
 - Dmesg コマンドや/var/run/dmesg.bootを参考に。
- ここも参考にしてください
 - The FreeBSD Project
<http://www.jp.freebsd.org/>