

Debian GNU/Linux で Linux kernel との遊びかた make-kpkg

2006-04-29

矢吹 幸治

yabuki@netfort.gr.jp

アジェンダ

- Linuxカーネルって何?
- カーネルパッチって何?
- Debian GNU/Linux でのカーネルの作り方
- KDBって何?
- 実際やってみよう
 - 手順
 - 実演

Linuxカーネルって何?

- Linuxカーネルは、Linuxそのものです。(狭義のLinux)
 - Linux上全部のプログラムはカーネルの恩恵を受けてます。
 - 主にC言語で書かれています。アセンブラも少々。

Kernel Patchって何?

- 本来の意味は、リーナスのリリースしたカーネルに対する差分を Patchコマンドで他の人にも適用できるようにした diffなどで生成されるテキスト形式
- 上記から転じて、カーネル本体に、(まだ?)採り入れられていない機能を実現するプログラム差分

Debian GNU/Linux での カーネル の作り方(1)

- make-kpkg
 - Debianは、カーネルもパッケージ管理されている。パッケージ管理を楽にしたり、カーネルパッチへの敷居を低くするために、カーネルパッケージ用コマンドがmake-kpkgである。
 - make-kpkg は、もちろんrootで使うことができるが、一般ユーザ権限でもカーネルをコンパイルすることができる。
 - Debian謹製のカーネルソース以外でもパッケージ化できます。注意：--initrdを付けるとき

Debian GNU/Linux での カーネル の作り方(2)

- 細かいオプションなどは、manやWebを見ればわかるので割愛
- make-kpkg と grub の組み合わせは便利。 ---
ここはliloだけどね ;-)
- カーネルを作ることは難しくない。一度正常な手順をやってみれば良い。この発表では、カーネルコンパイルの正常な手順を発表します。 -- VMware Workstation で実験済み

Debian GNU/Linux での カーネル の作り方(3)

- Tipsとしては、menuconfigは、ncursesを使っているなのでlibncurses5-devパッケージを入れておく -- headerは -devパッケージ
- kernel-packageというパッケージは入れておく
- Debianのカーネルパッチは、kernel-patch-ほげほげ という名称になっている。 -- aptitude やapt-cacheで探そう

KDBって何?

- KDB (Built-in Kernel Debugger)はSGIが配布しているGPLなカーネルデバッガです。一台だけで完結します。
 - KDBは下記のようなことができます。
 - CPUの命令ごとの動作, マルチプロセッサにも対応 (ssコマンド)
 - ブレークポイントでの停止 (bpコマンド)
 - メモリおよびレジスタ内容の表示と変更(m?/r?コマンド)
 - 実行しているタスクと他のタスク(プロセス単位)へのスタックのバック・トレース(btコマンド)
 - 逆アセンブル(idコマンド) などなど
 - IBM Developer worksのLinuxカーネル・デバッガ内部 - KDB入門編(http://www-06.ibm.com/jp/developerworks/linux/030912/j_1-kdebug.html)の記事がよさそう

実際にやってみよう

- 今回は、KDB についてをカーネルに入れてみます。
 - consoleでやっているなら、環境に応じて、日本語を表示できるようにしておきましょう。
 - 今回は、rootでの話です。fakerootコマンドを使えば、一般ユーザでも、自分の\$HOME以下でカーネルの作成は可能です。
 - 下記の手順は、Sargeを素で入れた状態からの手順を書いています。

手順(1)

- su -

- 特権へ入る

- aptitude update

- パッケージ管理データベースをインターネット越しに更新

- aptitude install kernel-tree-2.6.8

- カーネルのソースツリーを入手する。kernel-source-2.6.8 は、依存関係で入る。

- aptitude install kernel-patch-kdb

- カーネルパッチであるKDBをインストール

手順(2)

- aptitude install kernel-package
 - make-kpkgコマンドなどのカーネルパッケージの作成に必要なプログラムをインストール
- aptitude install libncurses5-dev
 - menuconfigは、ncursesを使っているのでコンパイルできるように開発用ヘッダーもインストール
- cd /usr/src/
 - Debianの場合、パッケージングされたカーネルソースコードは、/usr/srcに置くことになっている。ちなみに、カーネルパッチは、/usr/src/kernel-patches/以下にある。

手順(3)

- tar jxvf kernel-source-2.6.8.tar.bz2
 - bz2形式で固められているソースコードを展開する
- cd kernel-source-2.6.8
 - 展開したソースコードツリーの中へ入る
- cp /boot/config-2.6.8 .config
 - 現在動いている、カーネルの設定ファイルをコピーする。
- make-kpkg clean
 - カーネルをコンパイルしてできたバイナリーを削除して次のコンパイルに備える。初回は不要

手順(4)

- `make-kpkg --initrd --added-patches kdb --config menuconfig kernel_image kernel_headers modules_image`
 - kernel packageを作る。
 - initrd形式
 - カーネルソースにkdbパッチをあてる。
 - カーネルパッチをあてた後の設定はmenuconfigを使う。
 - コンパイルして作成するパッケージは、
 - (1)カーネル本体のパッケージ,
 - (2)カーネルヘッダーのパッケージ,
 - (3)(追加モジュール(IPW2200,IBM-ACPIなど)があれば)モジュール

手順(5)

- 実行するとpatchがあてられて、menuconfigの画面になるそこでは
 - menuconfig で 'kernel hacking' に入る
 - Kernel debugging を Y にする。
 - "compile the kernel with frame pointers" をy にする。
 - このオプションを Y にしたほうがスタックトレースがとりやすくなる。
 - Built-in Kernel Debugger support を Y にする。
 - KDB Module は M にする。
- コンパイル終了まで待つ

手順(6)

- `cd ..`
 - カーネルパッケージは、ひとつ上のディレクトリに作成されている。
- `dpkg -i kernel-image`
 - できたカーネルをインストールする。
- `vi /boot/grub/menu.lst`
 - Debian Sarge だと、install 時に `grub-update` で更新されているので、kernel option に `kdb=on` のオプションを設定、または起動してから `#echo "1"`
`>/proc/sys/kernel/kdb`
- `reboot`

実際にやってみよう

- <Pause> でデバッガー(kdb)へ移行する。
 - g で実行
 - bp schedule など
- mapファイルは、/boot/System.map-バージョンにあるとので参考にする
- では、楽しんで!