
User Mode Linux 活用法

Group TODO

宮本 久仁男

wakatono@todo.gr.jp

講師略歴

- 1991年3月 電気通信大学卒業
- 同4月 某大手SIer入社
- OS構成法の研究／開発、ECシステムの開発、社内技術支援業務を経て、現在は技術調査業務に従事。
- Kernel Panic, Freeze, 不意のRebootの末のファイル破損, マシンのハードウェアリセットとの戦いの末、User Mode Linuxとのつきあいに。

Agenda

- User Mode Linuxとは？
- User Mode Linux 使用法
- User Mode Linux 活用と注意
- 参考文献

User Mode Linuxとは？

User Mode Linux とは？

- ユーザプロセスで動作する Linux Kernel
 - 便宜上、動作させる環境をHostOS、User Mode Linuxを GuestOS と称することあり。
 - ネスト起動も可能
- HostOS の環境としては、カーネル2.4
 - Kernel 2.4.6 以上が必要
- 任意のユーザ権限で動作可能
- HostOSにUniversal TUN/TAPドライバを組み込むことで外部ネットワークも利用可能

開発の現状

- Kernel 2.4 ベース
 - Kernel Sourceに対する Patch で提供
 - 安定して動作
- Kernel 2.5 ベース
 - もともとはKernel Sourceに対するPatchとして提供
 - 一部のオプションを有効にするとコンパイル不可
 - Jffs2 など
 - 現在は Kernel Source に Merge されている
 - しかしコンパイルすらできない...
 - UMLそのもののデバッグになりそう...

利点と欠点

- ホストOSの環境を汚す心配なし
 - UMLinux カーネルとOSイメージがあればOK
 - COWを使えば、OSイメージも影響を受けない
- 仮想的なネットワークをいくらでも割り出せる
 - スイッチデーモンの利用
- 手軽に複数マシン環境を割り出せる
 - Coda や Intermezzo 等も楽に試験可能
- 怪しいカーネルを遠慮なく使える
 - ファイルシステムイメージは壊れるかもしれないが、マシンの再インストールをするよりはラク

利点と欠点(続き)

- 新機能を試したい時に良い
- ハードウェアドライバの作成には不向き
 - すでにHostOSによって(ある程度)仮想化された資源をさわるため
- ソフトウェアドライバの作成／デバッグ向き
- カーネルデーモンのデバッグ／試験向き
 - FreeS/WANの組み込み等で実績あり

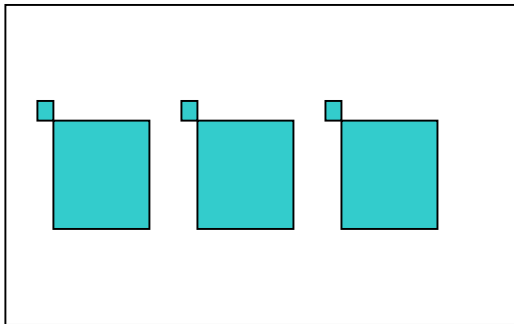
他の仮想OS環境との相違点

- VMWare
 - プロセス1コ
 - 1プロセスの上で、仮想的なPCハードウェアをエミュレート
- User Mode Linux
 - プロセスたくさん
 - 正確にはカーネルスレッド
 - UML上のプロセス1つとHostOS上のカーネルスレッド1つが対応

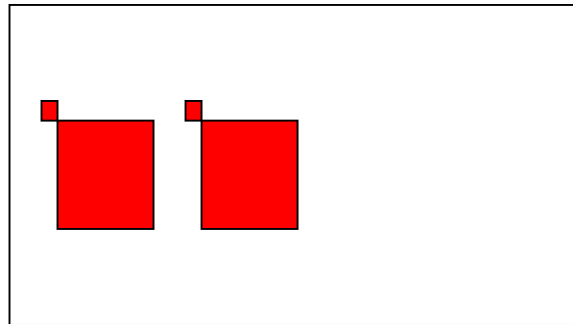
VMWareの場合(1/2)

- 見かけ上は...

Guest OS 1



Guest OS 2



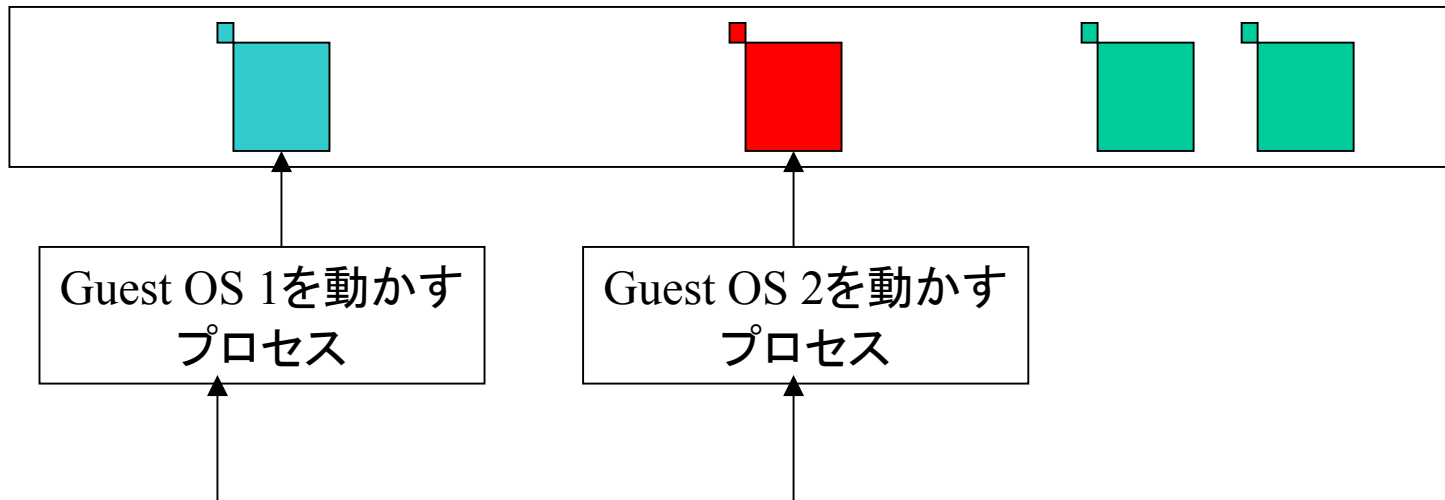
Host OS



VMWareの場合(2/2)

- 実際は...

Host OS

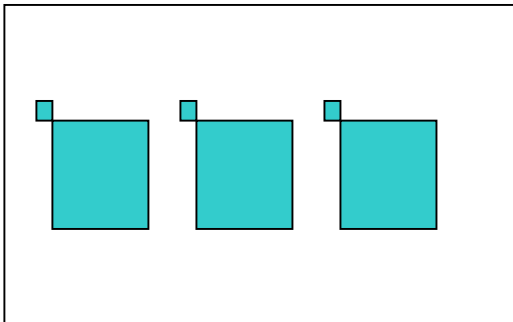


仮想OS環境1つがHost OS上の1つのプロセスに対応

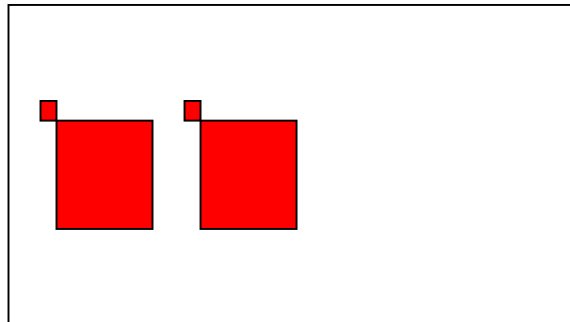
User Mode Linuxの場合(1/2)

- 見かけ上は...

Guest OS 1



Guest OS 2



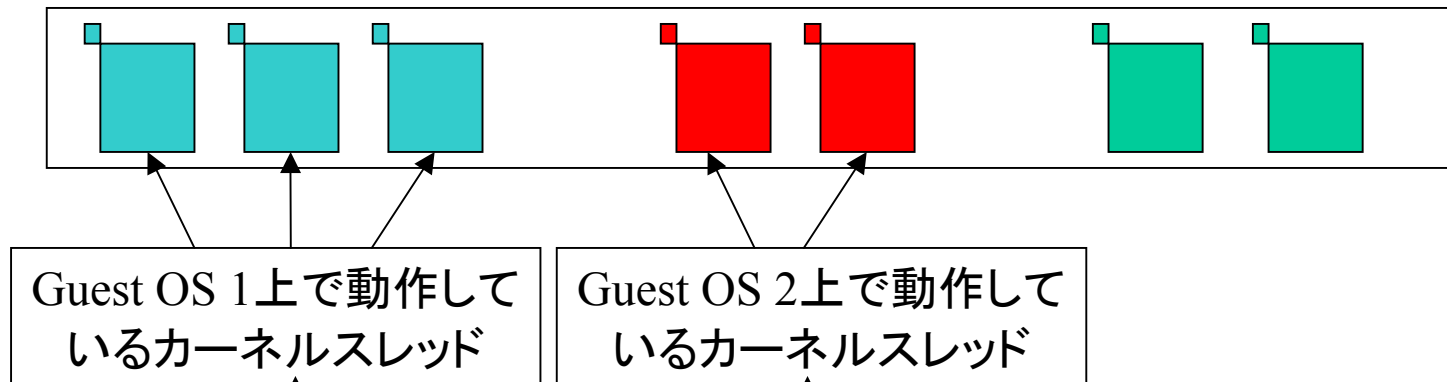
Host OS



User Mode Linuxの場合(2/2)

- 実際は...

Host OS



実際は、すべてHostOS上のカーネルスレッドにマッピング

プロセスの状況(1/2)

- GuestOSでのpsコマンドの結果
 - ごく普通のどこにでも見られるプロセスリスト

```
PID TTY STAT TIME COMMAND
1 ? S 0:00 init
2 ? SW 0:00 [keventd]
3 ? SWN 0:00 [ksoftirqd_CPU0]
4 ? SW 0:00 [kswapd]
5 ? SW 0:00 [bdflush]
6 ? SW 0:00 [kupdated]
7 ? SW 0:00 [mtdblockd]
350 ? S 0:00 syslogd -m 0
355 ? S 0:00 klogd -2
404 ? S 0:00 /usr/sbin/sshd
421 ? S 0:00 xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid
436 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
437 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
438 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
439 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
440 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
441 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
442 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
452 ? S 0:00 crond
453 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
454 ? S 0:00 /usr/sbin/httpd -DHAVE_PROXY -DHAVE_ACCESS -DHAVE_ACT
460 vc/0 S 0:00 login -- root
465 tts/0 S 0:00 /sbin/mingetty serial/0
484 vc/0 S 0:00 -bash
508 vc/0 R 0:00 ps -ax
```

プロセスの状況(2/2)

- HostOS上でのpsコマンドの結果
 - カーネルスレッドがたくさん

```
755 tty1 S 0:25 ./linux [(tracing thread)]
3021 tty3 S 0:00 -bash
5757 tty1 S 0:01 ./linux [(kernel thread)]
5762 tty1 S 0:00 ./linux [(kernel thread)]
5764 tty1 S 0:00 ./linux [(kernel thread)]
5766 tty1 S 0:00 ./linux [(kernel thread)]
5768 tty1 S 0:00 ./linux [(kernel thread)]
5770 tty1 S 0:00 ./linux [(kernel thread)]
5772 tty1 S 0:00 ./linux [(kernel thread)]
5773 tty1 S 0:00 ./linux [(kernel thread)]
5774 tty1 S 0:00 ./linux [(kernel thread)]
5776 tty1 S 0:00 ./linux [init]
6997 tty1 S 0:00 ./linux [syslogd]
7013 tty1 S 0:00 ./linux [klogd]
7187 tty1 S 0:00 ./linux [/usr/sbin/sshd]
7247 tty1 S 0:00 ./linux [xinetd]
7299 tty1 S 0:00 ./linux [/usr/sbin/httpd]
7301 tty1 S 0:00 ./linux [/usr/sbin/httpd]
7303 tty1 S 0:00 ./linux [/usr/sbin/httpd]
7305 tty1 S 0:00 ./linux [/usr/sbin/httpd]
7307 tty1 S 0:00 ./linux [/usr/sbin/httpd]
7309 tty1 S 0:00 ./linux [/usr/sbin/httpd]
7311 tty1 S 0:00 ./linux [/usr/sbin/httpd]
```

User Mode Linux 使用法

User Mode Linuxを使うには？

- カーネルの準備
 - すでにコンパイルされているものを利用
 - 自分でコンパイルして利用
 - こちらの方が安心はできる
- ファイルシステムイメージを準備
 - mkrootfs / UMLBuilder / gBootRoot / rootstrap などを
利用
 - イメージをダウンロードして使用
 - <http://user-mode-linux.sourceforge.net/> からたぐれる

User Mode Linux カーネル生成

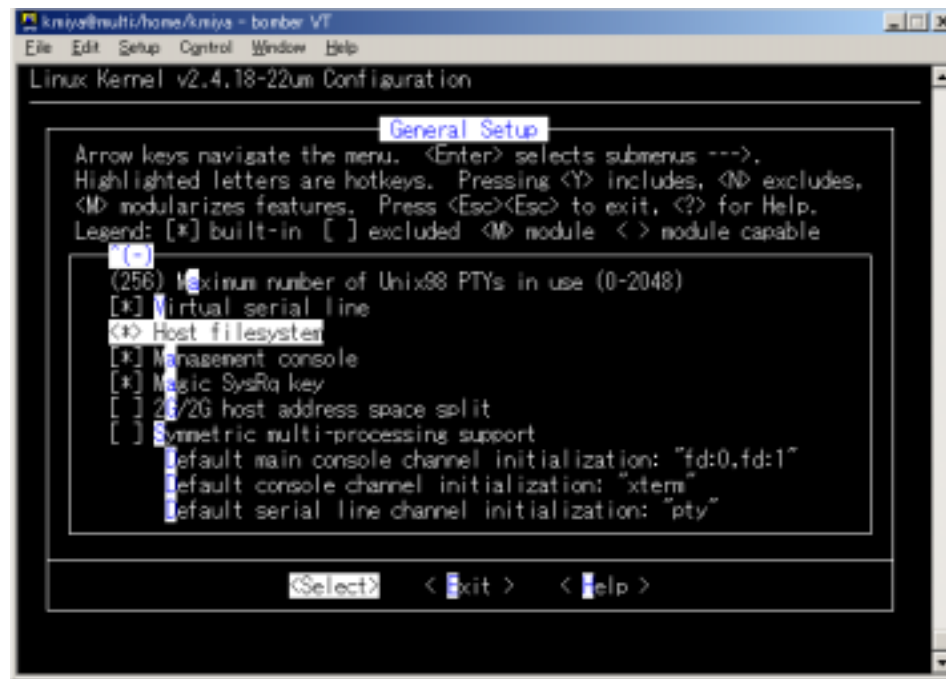
- RPM パッケージを利用
 - User Mode Linux HomePageからダウンロード可能
- カーネルソース + UML パッチ
 - パッチ済み状態のカーネルソースツリーで
 - `make menuconfig ARCH=um`
 - `make dep ARCH=um`
 - `make linux ARCH=um`
 - と次々に実行
 - Kernel 2.4 ではほぼ問題なし
 - Kernel 2.5 ではうまくいく時といかない時が...

Host Filesystem の利用

- Host Filesystem(以下Hostfs) とは何か？
 - Guest OS の環境から、Host OS 上のファイルを参照したりする機能
 - RPMパッケージに入っているカーネルでは、すでにHostfs の機能は有効になっている
 - 最初からカーネルを作成するような時には、あらかじめ当該機能を有効にしてカーネルの中に組み込むなどしてやる必要がある
 - もしくはその逆

Hostfsの設定

- HostFileSystemだからといって、FileSystemの設定ではなく General Setup で行う



Hostfsの挙動

- ファイルのオーナーやグループは、Guest OS 上に存在するものに置き換えられる
- Guest OS が動作しているユーザ権限を超えたアクセスは出来ない

Hostfsの使い方

- Hostfs の機能が有効なカーネルで

```
mount none /mnt/home -t hostfs
```

→Host OS のルートディレクトリが Guest OS 上の /mnt/home にマウントされる

- マウントするHost OS 上のディレクトリを指定する場合は `-o` オプションを利用。

```
mount none /mnt/home -t hostfs -o /home/kmiya
```

Hostfsの使用例(1/2)

- 筆者の環境でのマウント状況と df の状況

```
bash-2.05# mount
```

```
/dev/ubd/0 on / type ext2 (rw)
```

```
/proc on /proc type proc (rw)
```

```
devfs on /dev type devfs (rw)
```

```
devpts on /dev/pts type devpts (rw,mode=0622)
```

```
none on /mnt/home type hostfs (rw,/home/kmiya)
```

```
bash-2.05# df
```

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/ubd/0	175508	148590	17856	90%	/
none	1786760	1602032	93964	95%	/mnt/home

Hostfsの使用例(2/2)

- GuestOSの/mnt/home

```
bash-2.05# ls -l /mnt/home/
total 525848
drwxr-xr-x  5 user  user    4096 Apr 22 04:20 Desktop
drwxrwxr-x 18 user  user    4096 Apr 30 22:59 Python-2.2.1
-rw-rw-r--  1 user  user  6535104 Apr 30 22:37 Python-2.2.1.tgz
drwxrwxr-x  2 user  user    4096 May 16 03:34 auth
drwxrwxr-x 12 user  user    4096 May  1 00:42 cadaver-0.19.0
-rw-rw-r--  1 user  user   463968 Jan  6 16:31 cadaver-0.19.0.tar.gz
-rw-rw-r--  1 user  user    2693 Apr 23 06:08 cert.key
-rw-rw-r--  1 user  user    952 Apr 23 05:11 cert.pem
-rw-rw-r--  1 user  user    56 Apr 30 04:41 cmdline
-rw-----  1 user  user 10833920 May 13 05:44 core
```


User Mode Linux Utilities

- UMLinux で使用可能なツール
 - ツールのみソースコードは、`uml_utilities_*`
- `uml_mconsole`
 - UMLinux がブート時に作成する Unix Domain Socket (以下UDS) ファイルを経由して仮想OSを制御
 - socket ファイル名は、ブート時に表示
 - `mconsole (version 1) initialized on /home/kmiya/.uml/4QV2hZ/mconsole`

User Mode Linux Utilities(2/2)

- `uml_switch`
 - Switch Daemon と呼ばれる
 - Guest OS に対して仮想的なネットワークスイッチを提供
 - Switch mode / Hub mode を選択可能
 - 作成された仮想的なスイッチに対しては、

```
/usr/bin/linux ubd0=./diffs,/home/hoge/rootfs  
eth0=daemon
```

としてやることで、Guest OS が接続される

uml_mconsole使用例

(4QV2hZ) help

OK Commands:

version - Get kernel version

help - Print this message

halt - Halt UML

reboot - Reboot UML

config <dev>=<config> - Add a new device to UML;

same syntax as command line

remove <dev> - Remove a device from the client

sysrq <letter> - Performs the SysRq action controlled by the letter

cad - invoke the Ctl-Alt-Del handler

Additional local mconsole commands:

quit - Quit mconsole

switch <socket-name> - Switch control to the given machine

uml_switch 使用例

- 複数のネットワークを割り出したいときには

```
uml_switch -unix /tmp/net1ctl /tmp/net1.dat  
uml_switch -unix /tmp/net2ctl /tmp/net2.dat
```

```
$ uml_switch -unix /tmp/test /tmp/test2  
uml_switch attached to unix socket '/tmp/test'  
New connection  
Unknown connection for packet, shouldnt happen.  
Addr: fe:fd:00:00:00:00 New port 5  
New connection  
Addr: fe:fd:c0:a8:00:0a New port 6
```

User Mode Linux 活用と注意

発展的な使い方

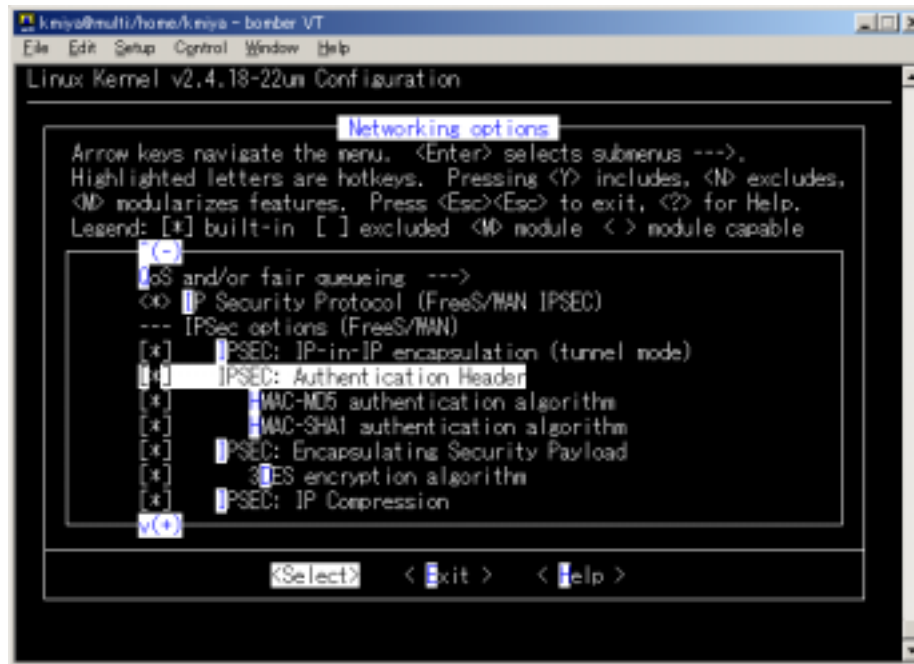
- さまざまなカーネルへの機能追加を実施
 - 基本的には、ハードウェアドライバよりはソフトウェアドライバ／カーネルデーモンのデバッグに向く
 - FreeS/WANの例
 - USAGIの例

FreeS/WANの例(1/3)

- User Mode Linuxパッチの適用
 - /usr/src 配下に展開された Linux カーネルのソースに対して UMLinux のパッチを適用
- FreeS/WAN パッチの適用
 - make menugo が使えない
 - make precheck
 - make verset
 - make insert
 - を順次実行

FreeS/WANの例(2/3)

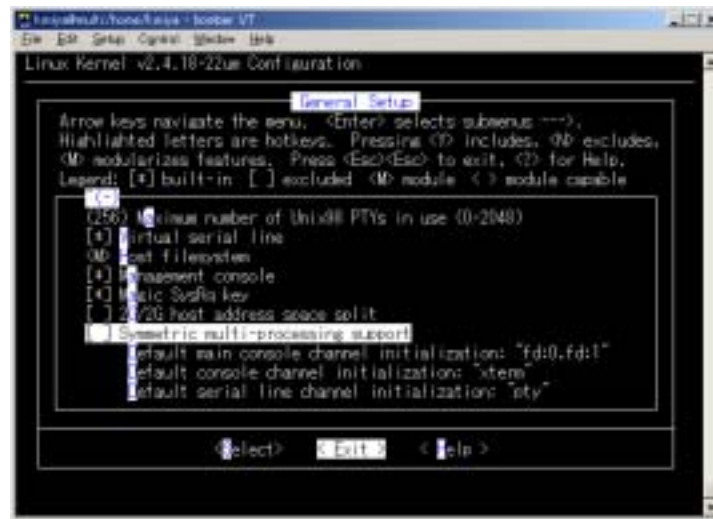
- make menuconfig ARCH=um
 - ARCH=um を指定して、menuconfigを実行



FreeS/WANの例(3/3)

- ハマりの例

- SMP Configuration が有効になっているとコンパイルが正常に通らない



起動メッセージの例

tracing thread pid = 3660

(略)

Kernel command line: ubd0=/home/kmiya/linux/difffs
eth0=tuntap,,192.168.1.100 root=/dev/ubd0

Calibrating delay loop... 413.86 BogoMIPS

Memory: 32244k available

(略)

klips_info:ipsec_init: KLIPS startup, FreeS/WAN IPSec
version: 1.97

(略)

USAGIの例(1/3)

- FreeS/WANと逆の順序でパッチ適用
- USAGI パッチの適用
 - `ftp://ftp.linux-ipv6.org/pub/usagi/snap/split/` から取得されるカーネルパッチを適用する
 - 今回の場合は、`usagi-linux24-s20020429-2.4.18.diff.bz2` という差分ファイルを用いた
- UMLinux パッチの適用
 - メッセージが出てきても気にしない

USAGIの例(2/3)

- Patch時のメッセージの例

patching file Documentation/Configure.help

Hunk #1 succeeded at 14033 (offset 382 lines).

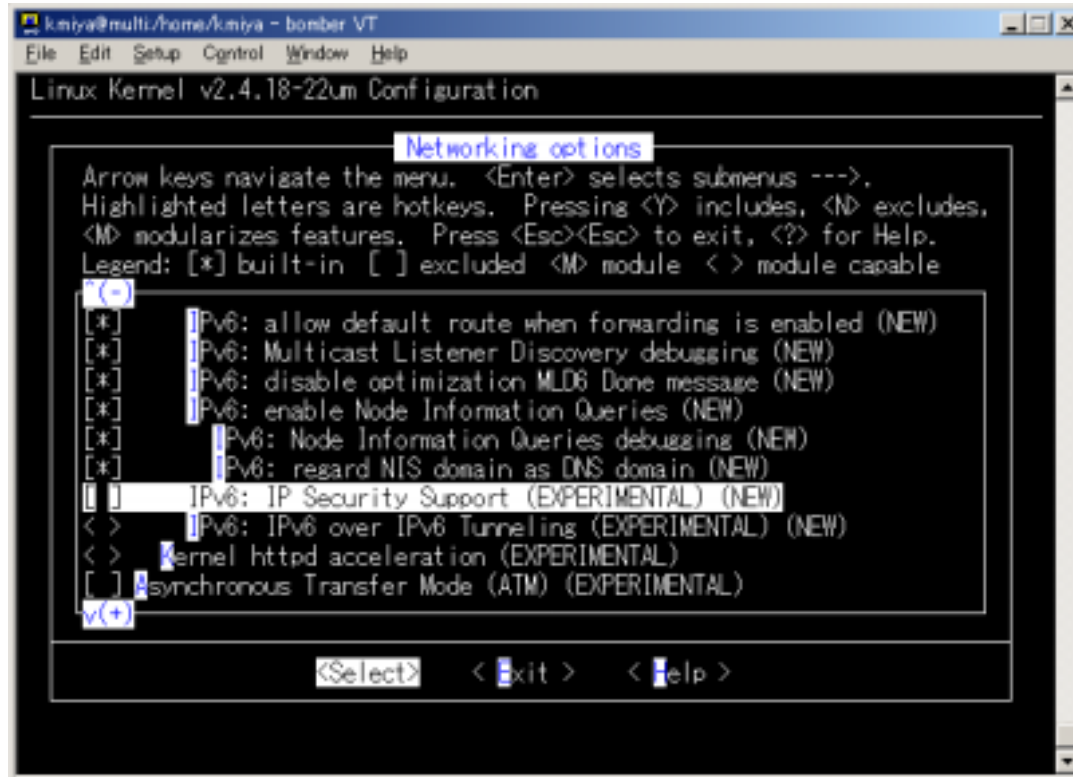
Hunk #3 succeeded at 14240 (offset 382 lines).

patching file MAINTAINERS

Hunk #1 succeeded at 1713 (offset 7 lines).

USAGIの例(3/3)

- make menuconfig ARCH=um



ブートメッセージの例

tracing thread pid = 23731

(略)

Kernel command line:

ubd0=./difffs,/home/kmiya/root_fs_crone
eth0=tuntap,,192.168.1.100 root=/dev/ubd0

(略)

IPv6 v0.8 (usagi-cvs) for NET4.0

(略)

User Mode Linux のセキュリティ

- 気になる部分

- HostOS側のセキュリティ

- Universal TUN/TAP ドライバのアクセス権限

- GuestOS環境がNetworkを使う／使わないで変化

- HostOS側のクラックによるGuestOSの環境破壊

- スレッドを殺されたり、プロセス／スレッドが不必要に増加したり

- GuestOS側のセキュリティ

- Hostfsによるアクセスの制御

- そもそも運用に使うような環境にはHostfsを組み込まない

- GuestOS環境のクラックの危険性

- HostOSは無事でもネットワークが使えてしまうと他がただではすまない

参考文献(1/2)

- The User-Mode-Linux HomePage
 - <http://user-mode-linux.sourceforge.net/>
- User-mode Linux Community Site
 - <http://www.usermodelinux.org/>
- User Mode Linux 紹介
 - <http://www.digitalinfra.co.jp/uml/index.html>
- 仮想OS「User Mode Linux」活用法
 - <http://www.atmarkit.co.jp/flinux/special/uml/uml01.html>

参考文献(2/2)

- 日経Linux 2002年7月号
 - 動向:フリーの仮想OS技術
- Software Design 2002年8月号
 - 仮想Linux UserModeLinuxの世界を覗く
- Linux World 2002年10月号
 - フリーエミュレータ活用術