



ORCA インフラストラクチャとしての Linux システム構築

Debian Project / 日本 Linux 協会

武藤 健志

<kmuto@debian.org>



進行

- ORCA
- Debian GNU/Linux
 - Linux ってなんだ？ Debian ってなんだ？
 - インストール / パッケージ管理 / X Window System / 日本語環境 / 印刷 / DB / メンテナンス / アップグレード / セキュリティ
- 質問は随時受け付けます





本資料に関して

- 本資料は

<http://www.topstudio.co.jp/~kmuto/debian/event/lc2002/>

から取得できます

- 各種パッケージ、ドキュメントも用意



ORCA

- ORCA

- 日本医師会の研究プロジェクト
- オープンシステム
- レセプト
- 医療情報データベース
- 公正な医療政策への提言



ORCA project logo

Copyright © 2002 by ORCA project. All rights reserved.

ORCA project is a project of the Japanese Medical Association (JMA).

ORCA project is a project of the Japanese Medical Association (JMA).

ORCA project is a project of the Japanese Medical Association (JMA).





インフラストラクチャ

- ORCA は OS 上で動作するアプリケーション
- OS に求められる要件
 - 安定性
 - 機能性
 - 透明性



Linux

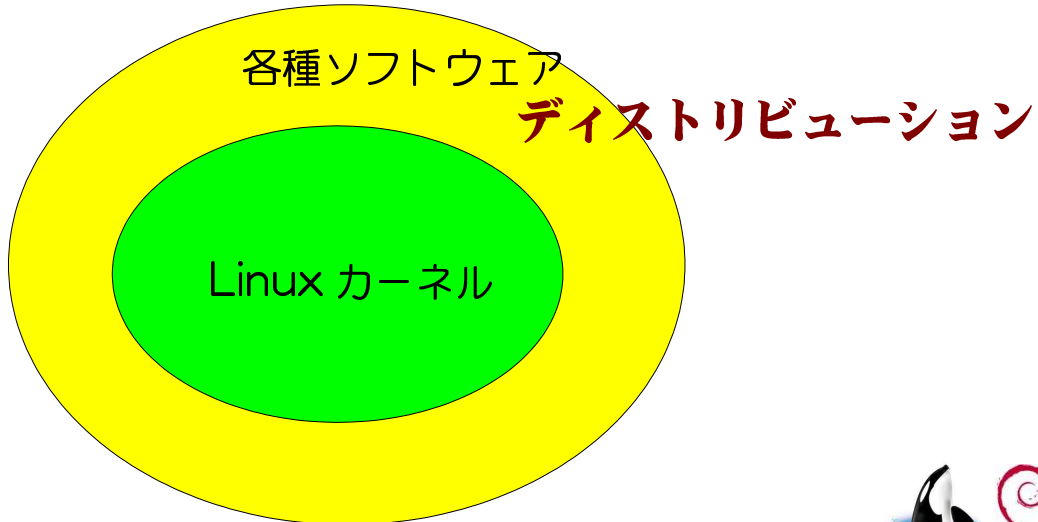
- ORCA のインフラストラクチャに採用
- Linus Torvalds にはじまる世界的な開発





ディストリビューション(1)

- ◎ カーネル + 各種ソフトウェア
 - ◎ カーネルはプロセス処理やI/O、ネットワークなどの低レベル部分を担当



ディストリビューション(2)

- ◎ OEM?
 - ◎ Windows の OEM よりもっと選択範囲が広い





Debian GNU/Linux

- ORCA のインフラディストリビューション
 - サーバーとしてではなくデスクトップ OS
- Vine や Omoikane も ORCA インフラとして整備が進行中

debian



Debian とは (1)

- 完全なフリーソフトウェア (オープンソースソフトウェア) だけで構成された OS
 - GNU/Linux , GNU Hurd , FreeBSD
- Debian Free Software Guidelines
 - Debian Project が考える「フリー」
 - ライセンス* ではない*
 - GNU GPL / BSD / Apache など
 - OpenSource Software の定義

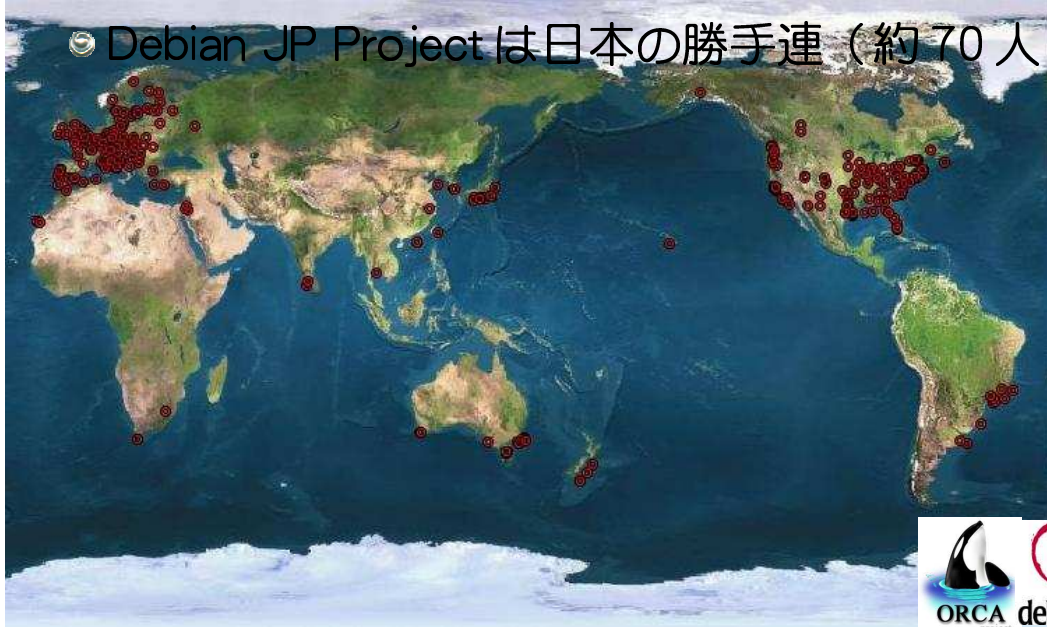




Debian とは (2)

◎ Debian Project

- ◎ 約 1000 人弱 / うち、日本人 33 人
- ◎ Debian JP Project は日本の勝手連 (約 70 人)



Debian とは (3)

◎ コードネーム

- ◎ Debian 2.2 Potato
- ◎ Debian 3.0(stable) Woody
- ◎ Debian x.x(testing) Sarge
- ◎ Debian (unstable) Sid



◎ プールシステム

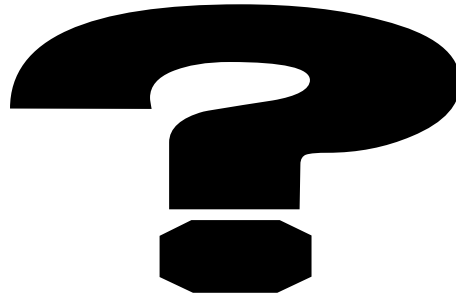
- ◎ いろいろなバージョンのパッケージが「プール」に
- ◎ 最新バージョンは unstable
- ◎ 「安定している」ものは testing





なぜ Debian が選ばれたのか

- 私は ORCA 関係者ではないので、内情はわかりません
- Debian の特長を考えながら推測



安定性

- Linux という安定したカーネル
- セキュリティ意識の高さ
 - セキュリティチーム (debian-security)
 - 各種セキュリティ ML / 団体に参加し、情報収集
 - 既存の安定版のサポート
 - リビジョンアップ
 - potato はそろそろサポート終了
- アップグレードの容易さ
 - APT
 - 再起動不要
 - libc / kernel のみ





機能性

- ◎ さまざまなコンパイル済みパッケージ
 - ◎ potato(contrib,main/non-US) **4073**
 - ◎ Woody(contrib,main/non-US) **8712**
 - ◎ デスクトップ環境、データベース、日本語環境 etc...
 - ◎ rpm とは別の独自パッケージフォーマット deb
 - ・ ただし、格納する情報としては大差なし
- ◎ APT によるパッケージ操作
 - ◎ 依存関係を自動解決
 - ◎ 削除や置き換えが簡単
 - ・ インストールばかりではない



透明性(1)

- ◎ フリーであるということ
 - ◎ 使うのは当然
 - ◎ 配っても 変更しても その変更したのをまた配っても すべてが自由
 - ◎ 勝手に売って商売にしても無料で配布しても
 - ◎ 「目玉の数が多ければバグなど物の数ではない」 by Eric S. Raymond
- ◎ フリーではないものは non-free として
 - ◎ non-free に強依存するパッケージは contrib





透明性 (2)

- いろいろディストロはありますが…
 - 企業の意向に左右されない
 - その企業がもしつぶれたら？
 - 独占的な方向にシフトしたら？
 - 開発者が多い
 - プロジェクトのマンパワーやモチベーションが尽きて崩壊ということはなさそう
 - メンテナ制度
 - パッケージについて興味と技術のある人がメンテナンス



Debian は難しい？

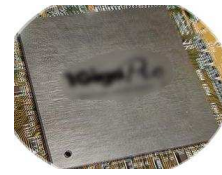
- 管理方法が RedHat 系と違う
 - FHS や LSB はあるが
- 参考書が少ない
- 対応を謳う商用ソフトウェアがほぼ皆無
- でも慣れればきっと！！
- Debian GNU/Linux の主にハマるところ
 - Potato はもう古い
 - Woody 版 ORCA も今月登場予定
- Woody で役立つ情報を中心に





インストールのポイント(1)

- ハードウェア選定
 - Intel アーキテクチャ PC
 - CPU
 - メモリ
 - ハードディスク
 - NIC
 - ビデオカード



インストールのポイント(2)

- デバイスの調査
 - NIC やビデオカード
 - SCSI カード





インストールのポイント(3)

- CD-ROM / DVD-ROM
 - フルサイズ
 - jigdo (<http://www.debian.org/CD/jigdo-cd/>) **8 枚 !**
 - ミニサイズ
 - ネットワーク (<http://www.debian.org/CD/netinst/>)
- FD
 - 環境により枚数変動
 - 最低 2 最悪 (2+6+20) = **28!**
- Loadlin
 - MS-DOS / Windows95, 98
- TFTP



インストールのポイント(4)

- フレーバー
 - Potato: vanilla, udma66, idepci, compact
 - Woody: vanilla, idepci, compact, bf2.4
 - CD1: isolinux(idepci) CD2:vanilla CD3:compact CD4: idepci CD5:bf2.4
- インストールさえ済ませれば、あとから置き換え可能
- USB Floppy だけが命綱の場合
 - USB Floppy 対応の Woody インストール
<http://www.debian.or.jp/~kmuto/usb/>





インストールのポイント(5)

- ◎ パーティショニング
 - ◎ ブート (/boot) 領域
 - ◎ スワップ
 - 後でファイルとして追加もできる
 - ◎ ホーム
 - ◎ 変動 (/var) 領域
 - ORCA では特に重要
- ◎ 壊れたら終わりなのであまり分けても…
 - バックアップのリカバリに楽？



インストールのポイント(6)

- ◎ ファイルシステム
 - ◎ bf2.4 で選択
- ◎ EXT2
- ◎ EXT3
 - ◎ EXT2+
ジャーナリング
 - tune2fs -j /dev/...
- ◎ ReiserFS
 - ◎ B-Tree
 - ◎ ジャーナリング
 - mkreiserfs /dev/...

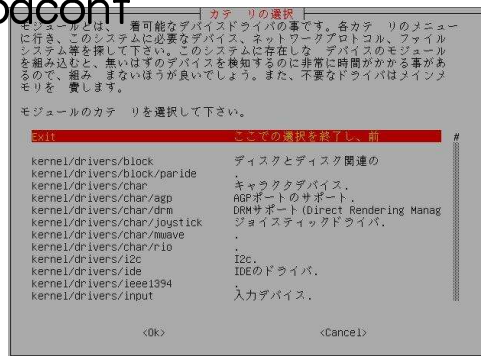




インストールのポイント(7)

● デバイスドライバ(モジュール)の組み込み

- あまり多く選ばない
 - NIC
 - parport / parport_pc
 - bf2.4 では必要なら apm
- あとから modconf



インストールのポイント(8)

● ネットワークの設定

- DHCP / 静的 IP
- インストーラのバグ
 - PCMCIA カードで静的 IP が設定されない
 - あとで /etc/network/interfaces を直す
 - <http://www.debian.or.jp/~kmuto/usb/> のイメージでは修正





インストールのポイント(9)

● パッケージの選択

● Task 選定

- 各パッケージがタスク情報を持つシステムに
- Woody の Task システムはイマイチ？

● dselect の悪夢

● お約束

● Woody のおまじない

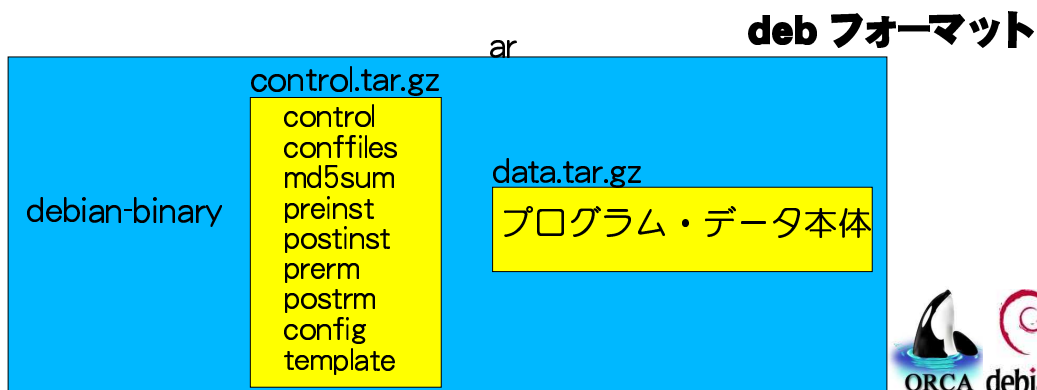
`dpkg-reconfigure -plow locales`



パッケージ管理(1)

● deb フォーマット

- rpm とあまり変わらない
- tar+gzip, メタ情報 を ar でアーカイブしたもの
- 依存 (depends)、推奨 (recommends)、提案 (suggests)、衝突 (conflicts)、置換 (replaces)、能提供 (provides)





パッケージ管理 (2)

● apt-get

● インストールまたは削除するツール

● apt-get [-f] install package

- インストール
- -f は修正インストール

● apt-get [--purge] remove package

- 削除（設定は保存）
- --purge は設定を含めて完全削除

● apt-get upgrade

- 他パッケージを削除しない範囲で更新

● apt-get dist-upgrade

- 他パッケージの削除を許可して更新



パッケージ管理 (3)

● /etc/apt/sources.list

● 「apt-line」

● deb http://... woody main contrib non-free

- 通常の apt-line
- http / ftp / file

● deb cdrom:[...]/... woody main contrib non-free

- CD-ROM / DVD-ROM

● deb http://... directory/

- 単一ディレクトリ

● deb-src

- ソースコードパッケージ

● apt-setup





パッケージ管理 (4)

● 典型的な sources.list

- deb http://ftp.jp.debian.org/debian woody main contrib non-free
- deb http://ftp.jp.debian.org/debian-jp woody-jp main contrib non-free
- deb http://ftp.jp.debian.org/debian-non-US woody/non-US main contrib non-free
- deb http://ftp.jp.debian.org/debian woody-proposed-updates main contrib non-free
- deb http://ftp.jp.debian.org/debian-non-US woody-proposed-updates/non-US main contrib non-free
- deb http://security.debian.org/ woody/updates main contrib non-free

● + ORCA 関係



パッケージ管理 (5)

● apt-cache

● パッケージの情報を検索・表示

● apt-cache show package

- パッケージ情報表示

● apt-cache search “keyword”

- キーワードにマッチするパッケージを検索

● auto-apt

● auto-apt search “filename”

- ファイルがどのパッケージに格納されているか検索



 aptitude

🌀 dselect に代わる操作インターフェイス

 dpkg-reconfigure debconf

🌀 フロントエンドの選択

- Dialog / Readline / Gnome /Editor / Noninteractive

- dpkg-reconfigure [-p low/medium/high/critical]
package

- debconf
- base-config
- locales





パッケージ管理 (8)

- update-alternatives
 - 同じ機能を持つパッケージの中でのデフォルト
 - update-alternatives --display name
 - update-alternatives --config name
- 主な alternative
 - editor
 - デフォルトのエディタ
 - pager
 - デフォルトのページャ (man など)
 - /etc/alternatives



パッケージ管理 (9)

- dpkg
 - rpm コマンドに相当するもの

操作内容	Dpkg	Rpm
パッケージの説明を表示	dpkg -S <u>package</u>	rpm -qi <u>package</u>
パッケージ内のファイル一覧	dpkg -L <u>package</u>	rpm -ql <u>package</u>
パッケージファイルの説明表示	dpkg -I <u>file</u>	rpm -qpi <u>file</u>
パッケージファイル内のファイル一覧	dpkg -c <u>file</u>	rpm -qpl <u>file</u>
インストールされたパッケージの一覧	dpkg -l <u>pattern</u>	rpm -qa
ファイルが所属するパッケージの検索	dpkg -s <u>file</u>	rpm -qf <u>file</u>
パッケージのインストール	dpkg -i <u>file</u>	rpm -i <u>file</u>
パッケージの削除	dpkg -r <u>package</u>	rpm -e <u>package</u>
パッケージの完全削除	dpkg -P <u>package</u>	該当ナシ





パッケージ管理 (10)

- dpkg を使う場面
 - 現在インストールされているパッケージの一覧
 - `dpkg -l`
 - これからインストールするパッケージの内容
 - `dpkg -c package-file`
 - カーネルやモジュールのインストール
 - `dpkg -i kernel-package-file`
 - 途中でインストールが中断してしまった
 - `apt-get -f install` でダメ
 - `dpkg --configure package`
 - `dpkg --configure -a`



X Window System の設定 (1)

- XFree86
 - Potato = 3.3.6 / Woody = 4.1.0
 - 非公式 4.2.1
 - `deb http://people.debian.org/~branden/ sid/i386/`
- デバイス確認
 - `lspci`
 - Intel i830, ATI Radeon 7500/8500, Matrox G550, NVIDIA nForce などは XFree86 4.2 ベース
 - i810/815 はカーネルでの対応も必要
 - bf2.4 と vanilla 以外のフレーバーはダメ
 - i845G
 - Woody では現状利用できない





X Window System の設定 (2)

● 基本設定

● debconf 設定

- dpkg-reconfigure -plow xserver-xfree86
- vesa ドライバまたはネイティブドライバ
- kernel framebuffer device は No
- keyboard model は jp106
- layout は jp

● 周波数、解像度

- Simple か Medium にして適当でも OK

● Intel 統合グラフィックチップ



X Window System の設定 (3)

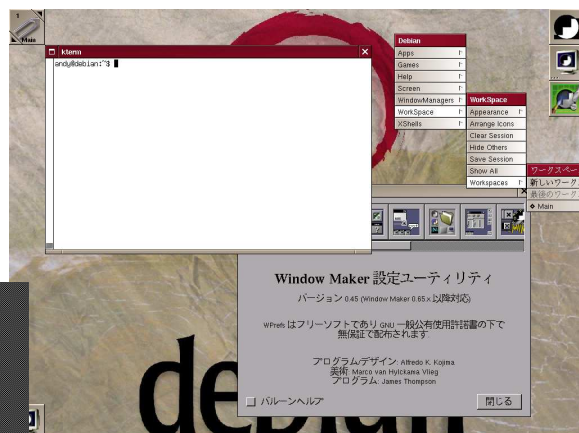
● ウィンドウマネージャの選択

● x-window-manager

● WindowMaker

- wsetfont japanese
で日本語化

● Sawfish





X Window System の設定 (4)

● TrueType フォントの設定

● xfs-xtt または XFree86 4.0 の FontPath

- freetype / xtt

● Xft

- X から freetype ライブラリを使ってダイレクトにフォントにアクセス
- アンチエイリアスなどの機能を高速に実現
- 今後の XFree86 の方向性

● ttf-kochi-mincho-naga10, ttf-kochi-gothic-naga10

● /etc/X11/XftConfig の設定

- dir /usr/share/fonts/truetype
dir /usr/share/fonts/truetype/kochi
- xftcache



X Window System の設定 (5)

● 統合環境

● インターフェイスを統一

● x-session-manager





X Window System の設定 (6)

GNOME

- 今までの ORCA のデスクトップとしておなじみ



- gnome-session, gnome-control-center, nautilus
- ~/.gnome*, ~/.gconf*

Nautilus

- gmc に代わるファイルマネージャ
- x-ttcidfontconf, kochi フォントパッケージ
- ln -s /var/lib/defoma/x-ttcidfontconf.d/dirs/TrueType ~/.nautilus/fonts



X Window System の設定 (7)

KDE

- 統一が GNOME より取れている

- kde kde-i18n-ja

- 日本語にあたっての初期設定

- Preferences→Personalization→Country & Language→jp / ja / jisx0208.1983-0

- 再ログイン

- アンチエイリアス

- libfreetype6 の入れ換え
- Preferences→Look & Feel→Fonts→antialias
- 再ログイン





X Window System の設定 (8)

● ディスプレイマネージャ

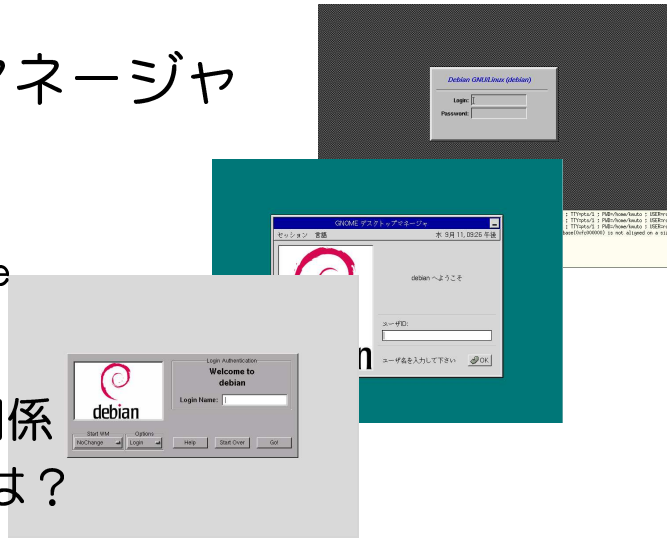
● GUI ログイン

- xdm, gdm, wdm
- dpkg-reconfigure xdm/wdm/gdm

● Debian では runlevel とは無関係

● 統合環境の選択は？

- .xsession
- update-alternatives --config x-session-manager



日本語環境 (1)

● Woody での日本語関係の変化

● ロケール名

- ja_JP.ujis → ja_JP.eucJP

● locale-ja パッケージ廃止

- GNU libc に統合





日本語環境 (2)

● Canna + kinput2

● 入力スタイルの設定

- default, 1.1, 1.2, jdaemon, just, Lan5, matsu, skk, tut, unix, vje, wx2+ (libcanna1g)
/usr/share/canna/ からコピーされる
- Egg/Kinput2 (kinput2-common)

● 辞書

- と思ってたのですが



日本語環境 (3)

● language-env

● かつて user-ja だったもの

● set-language-env

- doc-linux-text-ja, mutt-ja はない

● root では実行しない





日本語環境 (4)

● ATOK X

- JustSystem の商用ソフトウェア
- atokx パッケージ

● 私は ATOK X を持っていないので…

1. atokx パッケージを入れる

2. .xsession の XIM の部分

- XIM=kinput2 → XIM=htt
XIM2=-canna → XIM2=
- export XMODIFIERS の下に
/usr/lib/im/locale/ja/atokserver/atokx_client



日本語環境 (5)

● PDF

- xpdf

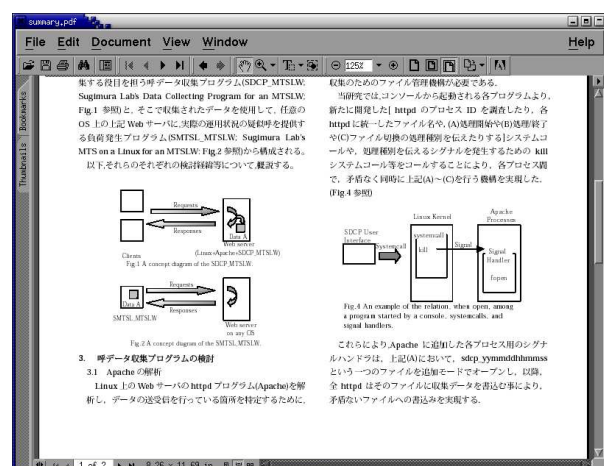
● Acrobat Reader

- セキュリティ問題
- 非公式パッケージ

- acroread, acroread-asianfont

● フォントと印刷に関係するいくつかの問題

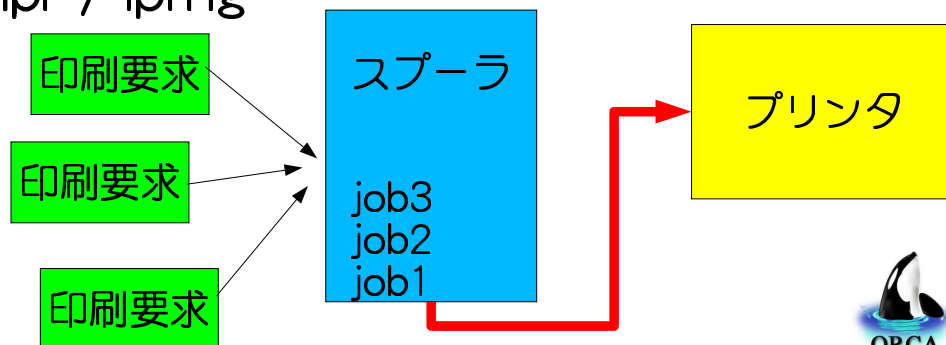
- <http://www.matsusaka-u.ac.jp/~okumura/linux/acroread.html>





印刷の整備 (1)

- Postscript
 - Adobe 社のページ記述言語
 - プリンタ側にインタプリタが必要
- スプーラ
 - 印刷ジョブの円滑な管理
- lpr / lprng



印刷の整備 (2)

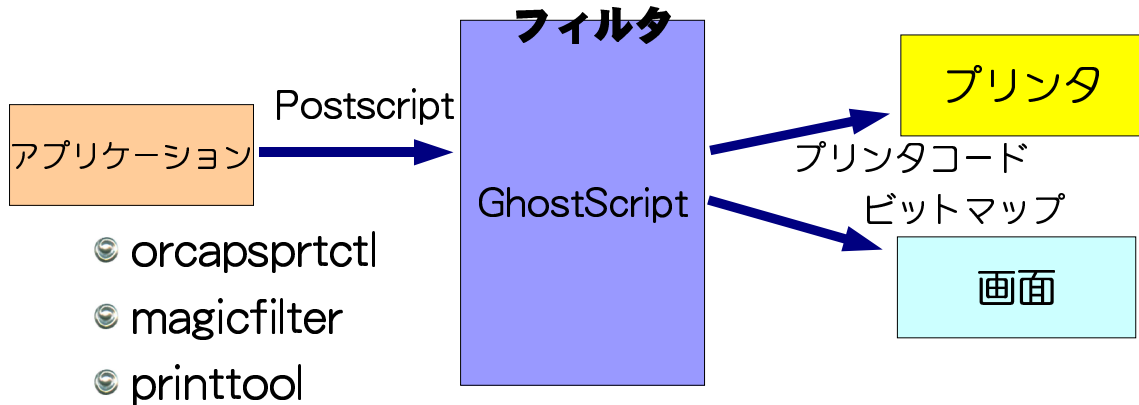
- Ghostscript
 - Postscript からの出力コンバータ
 - Potato→Woody で変化
 - lips 問題
- Debian Font Manager (defoma)
 - dfontmgr で TrueType/CID を管理
- インストールと設定
 - `apt-get install gs gsfonts gs-cjk-resource cmap-adobe-japan1 cmap-adobe-japan2 defoma psfontmgr ttf-kochi-gothic-naga10 ttf-kochi-mincho-naga10`





印刷の整備 (3)

◎ フィルタ



印刷の整備 (4)

◎ magicfilter パッケージの設定

- ◎ magicfilterconfig はイマイチ
- ◎ /etc/magicfilter/ にある各種フィルタファイル
 - *-filter: ファイルの先頭に情報
- ◎ ちょうどよいものがない場合には「プリンタ名-filter」でコピー
 - gs 設定の -sDEVICE=(デバイス名)と -r(DPI)
 - テキスト印刷設定 (a2ps-perl-ja パッケージ)

default pipe /usr/bin/enscript -q ...



default pipe /usr/bin/a2psj





印刷の整備 (5)

● 設定したフィルタの使用

● /etc/printcap

- .common:

- :sd=/var/spool/lpd/%P

- :sh:mx=0:mc=0

- # Printer on Parallel Port

- lp | psprinter

- :tc=.common

- :lp=/dev/lp0

- :filter=/etc/magicfilter/psfoo-filter

● リモートプリンタなら

- :lp= プリンタ名 @ ホスト名



データベースの設定 (1)

● PostgreSQL

● Potato 時代の問題

- apt-get --purge remove postgresql; apt-get install postgresql

● Woody 時代の問題

● ディレクトリ構造

● /etc/postgresql/

- 設定

● /usr/lib/postgresql/bin/

- 実行プログラム

● /var/lib/postgresql/

- データ





データベースの設定 (2)

- PostgreSQL の操作
 - 通常はパッケージで対処
 - ユーザーアカウントの作成、削除
 - postgres ユーザー
 - createuser / dropuser username
 - データベースの作成、削除
 - createdb dbname / dropdb dbname
 - バックアップ
 - pg_dump dbname > bakfile
- psql インターフェイス



管理のポイント(1)

- root ログインはしない
 - 誤操作が大ダメージ
 - ついつい日本語環境を構築したくなる
 - root 環境は最悪の状況でも動作するようにする

Login: ~~root~~





管理のポイント(2)

- su
 - 原則禁止！
- sudo
 - `sudo [-u loginname] command...`
 - root のパスワードを教えなくてよい
 - コマンドの実行可能範囲を設定できる
 - ログ (`/var/log/auth.log`) が取れる
 - これからどういう操作をするのかが明確になる
- visudo
 - `/etc/sudoers`
 - `kmuto ALL=(ALL) ALL`



管理のポイント(3)

- make install は極力しない
 - `/usr/local/` は許可されているが
 - ハマリの原因になる
 - `apt-cache search` などで探してみよう
 - いろいろな apt-line
 - <http://www.internatif.org/bortzmeyer/debian/apt-sources/>





管理のポイント(4)

- APT の PIN 機能
 - apt-get install package=version
 - release 名での指定
 - apt-line で woody のほかに sid も入れておく
 - /etc/apt/preferences
Package: *
Pin: release a=stable
Pin-Priority: 1000
 - apt-get install package/release
 - stable, unstable, proposed-updates, ...



管理のポイント(5)

- deb 化する
 - Sid にあるならリビルドで済むことも
 - apt-get -b source package
 - Debian / JP Project や ORCA に頼んでみる？
- マニュアル
 - Debian メンテナ入門
 - <http://www.debian.or.jp/~mhatta/docs/maint-guide-ja/index.ja.html>
 - Debian パッケージの作り方
 - <http://www.topstudio.co.jp/~kmuto/debian/event/lc2000f/>





管理のポイント(6)

● ユーザー管理

● useradd

- ユーザーの追加
- グループに追加 (useradd -G group loginname)

● userdel

- ユーザーの削除

● グループ設定

- adm, floppy, cdrom, dialout, audio
- chown cdrom /dev/hdc (CD-ROM デバイス)



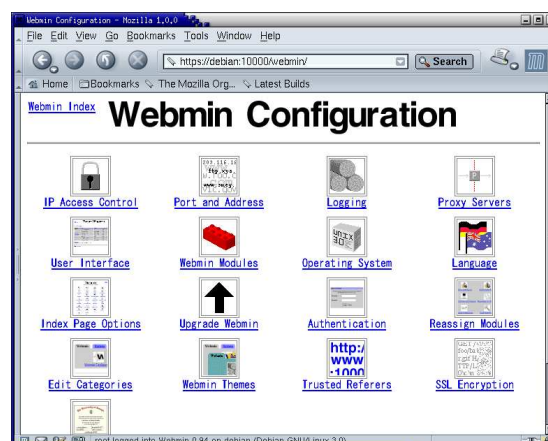
管理のポイント(7)

● webmin

● Web ブラウザベースの管理

- <https://localhost:10000/>

● あまりお勧めはしません





リモート管理(1)

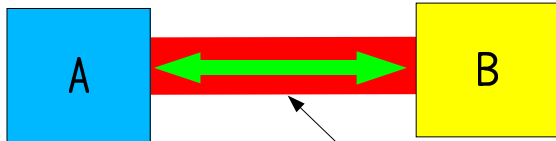
SSH

version 1 / version 2

- /etc/ssh/sshd_config
Protocol 2,1
HostKey /etc/ssh/ssh_host_key
- ssh-keygen -f "/etc/ssh/ssh_host_key" -N '' -t rsa1

SSHのクライアント

- ssh hostname / slogin hostname
- TeraTerm Pro+TTSSH プラグイン
<http://www.zip.com.au/~roca/ttssh.html>
- Putty + PuttyJP / パッチ
<http://pure.fan.gr.jp/simm/ssh/putty.html>



リモート管理(2)

X Window System のウィンドウ

- /etc/ssh/sshd_config
X11Forwarding yes
- ssh -X

Astec-X

- <http://www.astec.co.jp/products/ASTECX/astecx.html>

VNC

vncserver / xvncviewer

- <http://www.uk.research.att.com/vnc/>





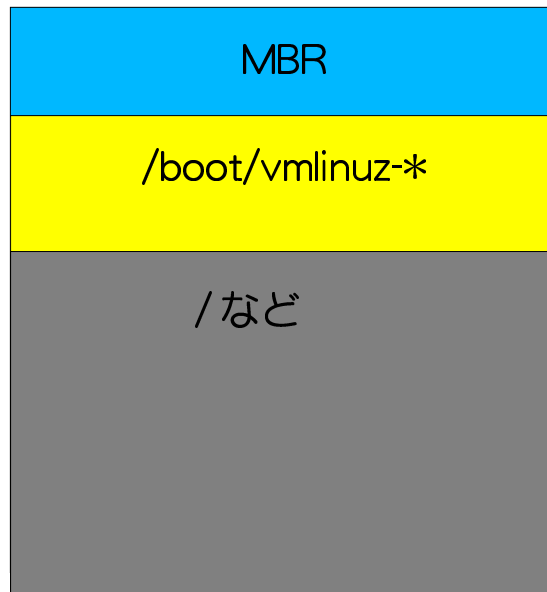
カーネルのコンパイル (1)

- カーネルの場所

- /boot/vmlinuz-*
- /vmlinuz リンク

- MBR

- ブートローダ



カーネルのコンパイル (2)

- コンパイル済みカーネルパッケージ

- 2.4.18 については CPU に合わせたパッケージが
 - kernel-image-2.4.18-386/586tsc/686/686-smp/k6/k7/bf2.4

- initrd

- initrd-tools

- ブートローダの設定 (lilo)

- /etc/lilo.conf

- image=/vmlinuz
initrd=/initrd.img
- image=/boot/vmlinuz-*
- initrd=/boot/initrd.img-*

- /sbin/lilo を忘れずに





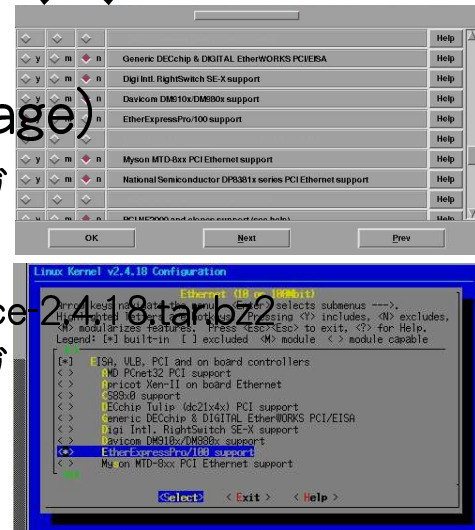
カーネルのコンパイル (3)

- なぜカーネルを再構築するのか
 - パフォーマンス
 - ハードウェア障害



カーネルのコンパイル (4)

- make-kpkg (kernel-package)
 - カーネルソースパッケージ
 - kernel-source-2.4.18
 - tar jxf /usr/src/kernel-source-2.4.18.tar.bz2
 - カーネルパッチパッケージ
 - kernel-patch-usagi
- xconfig / menuconfig
 - tk8.3-dev / libncurses5-dev
- PATCH_THE_KERNEL=YES make-kpkg --config xconfig [--added-patches patch, ...] configure
- .config の流用





カーネルのコンパイル (5)

- ◎ 最適化のポイント
 - ◎ インストーラのカーネル設定から不要なものを除いていく
 - ◎ CPU
 - 適切なものに。SMP
 - ◎ IDE
 - Intel / VIA 82cxxx
 - ◎ SCSI
 - ATAPI CD-RのためのSCSIエミュレーションも含む
 - ◎ AGP サポート
 - Intel 統合チップセット



カーネルのコンパイル (6)

- ◎ fakeroot とコンパイル
 - ◎ fakeroot make-kpkg --revision rev kernel_image
 - ◎ revision は「ホスト名 + 作った番号」など
 - orca+1
 - ◎ CONCURRENCY_LEVEL
 - 並列化
- ◎ インストール
 - dpkg -i kernel-image-2.4.18_orca+1_i386.deb
- ◎ 最初からやり直し
 - ◎ fakeroot make-kpkg clean





カーネルのコンパイル (7)

● モジュールパッケージ

● lm-sensors-source

- 温度やファンの監視

● pcmcia-source

- カーネル 2.4 ではカーネルに PCMCIA 機能がある

● e100-source

- Intel NIC の Intel によるドライバ

● コンパイル

- /usr/src/ モジュール tar.gz を展開
- MODULE_LOC に module ソースディレクトリを指定
- カーネルソースディレクトリで
fakeroot make-kpkg modules_image



ブートローダー grub (1)

● lilo の問題点

● いちいち /sbin/lilo

- grub なら設定ファイルの修正だけ

● 間違った設定を起動時に直せない

- grub ならシェルインターフェイスで修正

● grub フロッピー

● grub-floppy /dev/fd0

- root(HDno, PARTno)
- kernel /boot/vmlinuz-... root=/dev/... ro
- initrd /boot/initrd.img-...
- boot
- chainloader +1 (最初のセクタ)





ブートローダー grub (2)

● ハードディスク

- grub-install [--root-directory=/boot] /dev/hda
 - バグで2回必要？
- update-grub
 - /boot/grub/menu.lst
 - kopt, groot
- /etc/kernel-img.conf
 - do_bootloader = no
 - postinst_hook = /sbin/update-grub
 - postrm_hook = /sbin/update-grub
- apt-get remove lilo



セキュリティ(1)

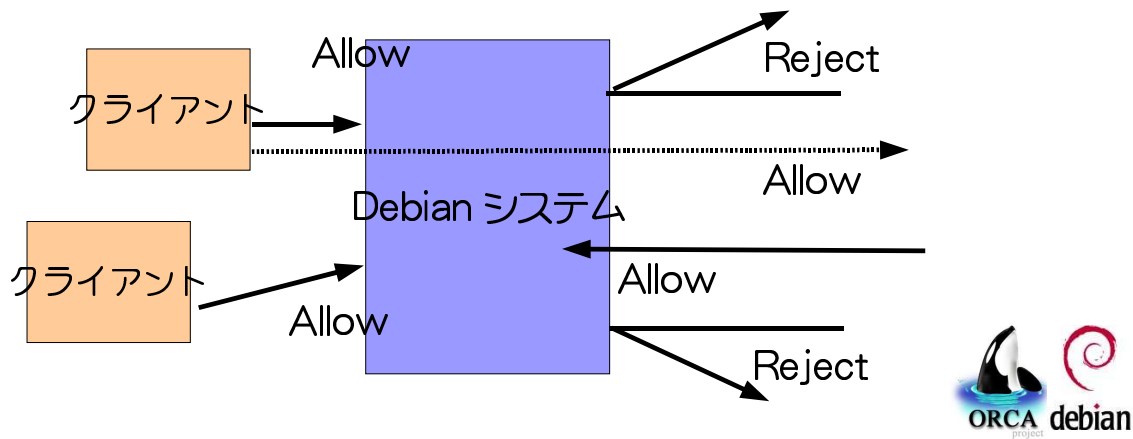
- 機密性と信用
- カーネルレベルのセキュリティ、アプリケーションレベルのセキュリティ
- 「まず全部拒否、それから許可」
 - 逆は泥縄
- 「複雑にしない」
 - わからないものは管理できない
- 「無駄なユーザーを作らない、無駄なアプリケーションをインストールしない」
 - 増えるほど潜在的危険が高まる





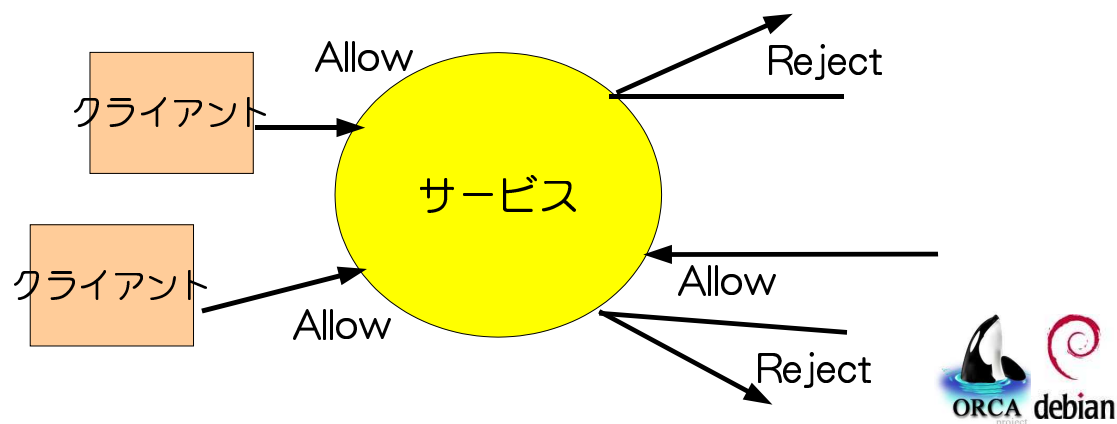
セキュリティ(2)

- ipchains / iptables
 - カーネルレベルのセキュリティ
 - ポートフィルタリング
 - ホストやネットワークを特定してフィルタリング



セキュリティ(3)

- TCP wrappers
 - アプリケーションレベルのセキュリティ
 - ホスト、ネットワーク、ユーザーを特定してフィルタリング
 - /etc/hosts.allow, /etc/hosts.deny





セキュリティ(4)

- ログのチェック
 - /var/log/syslog
 - /var/log/auth.log
- ログ監視ツール
 - log2mail
 - swatch



セキュリティ(5)

- 専用ルーター
 - システムを直結するというリスクの回避
 - セキュリティ機能を装備
 - シンプル
 - 発熱や障害耐性





セキュリティ(6)

- セキュリティ侵害はネットワーク越しだけではない
- 内部犯行への注意
 - 内部 LAN の防備は手薄
 - 盗聴、盗難、破壊工作
- 地道な教育



セキュリティ(7)

- ノート PC の危険性
 - 簡単に盗難される
 - サスペンド機能の危険
 - スクリーンセーバー
 - BIOS パスワード
- 無線 LAN
 - 接続可能なカードの MAC アドレス指定は基本
 - ESSID や WEP は万能ではない
 - 通信の暗号化
 - ssh -X
 - VPN





セキュリティ(8)

- ◎ バックアップからの機密流出
- ◎ GPG (gnupg) を使った暗号化
 - ◎ 鍵の作成
 - `gpg --gen-key`
 - DSA/ElGamal, 2048 ビット
 - ユーザー名, メールアドレス
 - パスフレーズ
 - ◎ 暗号化の実行
 - `gpg --encrypt -r 'orca@orca.jp' backup.tar.gz`
 - `backup.tar.gz.gpg`
 - ◎ 復元
 - `gpg backup.tar.gz.gpg`



Potato からのアップグレード(1)

- ◎ `apt-get dist-upgrade / dselect`
 - ◎ `dselect` を Debian Project では推奨
- ◎ 実験を行った限りでは
 - ◎ そう大きく差は出ない
 - ◎ `dselect` のほうがいろいろ引っぱってくる
- ◎ 実際のところ
 - ◎ 再インストールしたほうがよいことも
 - 特定目的なので重要データは散らばっていない





Potato からのアップグレード(2)

◎ 注意点

◎ バックアップ

- /etc, /var, /home

◎ postgresql, gnome-bin の障害

- ORCA パッケージのバグ
- ORCA で対策中

◎ locales

- dpkg-reconfigure -plow locales

◎ ネットワーク設定

- **auto lo**
iface lo inet loopback
- **auto eth0**
iface eth0 inet ...



まとめ

◎ これが完成形ではない

- ◎ Debian も ORCA も未完成な部分が多い

◎ 声をあげていこう

- ◎ バグ報告
- ◎ 助言サポート
- ◎ ドキュメント

◎ 一緒に改良していこう





参考

● Debian-users / Debian-devel

- <http://www.debian.or.jp/MailingList.html>

● Debian JP Project が開設している ML

● Linux-users

- <http://www.linux.or.jp/community/ml/linux-users/>

● Japan Linux Users Group による大規模な ML

● ORCA-users / ORCA-dev / ORCA-tech

- <http://www.orca.med.or.jp/ml/>

● ORCA が開設しているメーリングリスト

