

**2006年度  
オープンソースソフトウェア基盤整備事業**  
「Linux ディストリビューション仕様記述様式に関する調査」

**自動判別ツール(プロトタイプ)**

**外部仕様書**

2007年10月

**独立行政法人 情報処理推進機構**

Version 1.0

本文書は、IPA「2006年度オープンソースソフトウェア基盤整備事業(Linux ディストリビューション仕様記述様式に関する調査)」に係わる委託業務で作成された。

本書の一部、または全部を著作権所有者の許諾なしに、商用目的のために複製、配布することはできません。

Turbolinux、ターボリナックスの名称およびロゴはターボリナックス株式会社の商標または登録商標です。Linux は Linus Torvalds 氏の米国および他の国における商標です。UNIX は The Open Group の米国および他の国における登録商標です。Red Hat、および RPM は Red Hat, Inc. の米国およびその他の国における登録商標です。X Window System は X Consortium, Inc. の商標です。Microsoft、MS-DOS、Windows は米国 Microsoft Corporation の米国及びその他の国における登録商標です。

その他、記載された会社名およびロゴ、製品名などは該当する会社の商標または登録商標です。

# 目次

1. 自動判別ツール（プロトタイプ）の概要.....	1
1.1. 背景・目的.....	1
1.1.1. 背景.....	1
1.1.2. 目的.....	1
1.2. 自動判別ツールとデータベースについて.....	1
1.3. プロトタイプに必要な機能.....	2
1.3.1. ハードウェア自動判別機能.....	2
1.3.2. アプリケーション提示機能.....	3
1.3.3. Linux ディストリビューション情報の自動更新.....	3
1.4. 対象とするハードウェア機器.....	3
1.5. 対象とするディストリビューション.....	4
2. 自動判別ツール（プロトタイプ）の機能.....	5
2.1. システム機能概要.....	5
2.2. Hardware 機器のサポート情報自動判別.....	5
2.2.1. ハードウェア機器の検出機能.....	5
2.2.2. 通信機能.....	5
2.2.3. 画面表示機能.....	5
2.2.4. 動作イメージ.....	6
2.3. データベース・メンテナンス機能.....	6
2.3.1. データベース自動投入データ生成機能.....	6
2.3.2. データベース更新/追加機能.....	7
2.3.3. データベース検索機能.....	7
2.3.4. 動作イメージ.....	7
3. ハードウェア検出機能詳細.....	8
3.1. ブート機能.....	8
3.1.1. 利用する Linux Kernel.....	8
3.1.2. ブート機器（デバイス）.....	8
3.1.3. ブートの仕組み.....	8
3.1.4. ルートツリー.....	9
3.2. 画面表示機能.....	9
3.2.1. 利用する X Window システム.....	9
3.2.2. Window マネージャ.....	9
3.3. ハードウェア機器検出ソフトウェア.....	10
3.3.1. ハードウェア機器の検出.....	10
3.3.2. ハードウェア機器の検出データ構造.....	10

3.4. 通信機能.....	11
3.4.1. Ethernet 通信.....	11
3.4.2. サーバー側との通信プロトコル.....	12
3.4.3. サーバーとのデータ通信.....	12
3.5. ユーザインタフェース機能.....	12
3.5.1. 利用するインタフェース.....	12
3.5.2. その他ユーザへのインタフェース.....	12
4. データベース機能.....	13
4.1. データベース機能.....	13
4.1.1. Linux Kernel テーブル (モジュール) .....	13
4.1.2. Linux Kernel テーブル (vmlinuz) .....	14
4.1.3. X Window テーブル.....	14
4.1.4. デバイス名情報テーブル.....	14
4.1.5. アプリケーションソフトウェアテーブル.....	14
4.1.6. 検出結果テーブル.....	15
4.2. データベースインタフェース機能.....	15
4.2.1. データ自動投入.....	16
4.2.2. データ手動投入.....	16
4.2.3. データ追加.....	16
4.2.4. データ削除.....	16
4.2.5. データ検索.....	16
付録 A. ツリー構造.....	18
A.1. ルートツリー構造.....	18
A.2. ISO イメージファイルの作成.....	30
付録 B. サーバー側への問い合わせに関して.....	31
B.1. ローカルでの収集情報の転送.....	31
付録 C. データ ベースへの自動投入に関して.....	32
C.1. Linux のデバイス情報.....	32
C.2. Linux Kernel モジュール情報.....	32
C.3. X Window システムモジュール情報の取得.....	32
C.4. 該当パッケージからの情報取得.....	32
C.5. アプリケーションソフトウェアの情報.....	33
C.5.1. Debian 系 Linux ディストリビューションのパッケージ情報取得.....	33
C.5.2. RPM 系 Linux ディストリビューションのパッケージ情報取得.....	33
C.5.3. データベースの投入データ生成 (Linux Kernel 編).....	34
C.5.4. vmlinux ファイルから Linux Kernel のバイナリ情報を取得する場合.....	36
C.5.5. コンパイルにて vmlinux を生成し、Linux Kernel のバイナリ情報を取得する場合.....	42
C.6. データベースの投入データ生成 (X Window システム編) .....	43
C.6.1. Binary ファイル (xxx_drv.so) から情報を入手する場合.....	44

C.6.2.各ディストリビューションが配布しているテキスト情報から入手する場合.....	55
C.6.3.各ディストリビューションが配布しているソースコードから情報を入手する場合.....	55
C.6.4.データベースの投入データ生成(パッケージ情報編).....	56
C.6.5.Debian 系パッケージの場合.....	56
C.6.6.RPM 系パッケージの場合.....	63
C.7. データベースへの自動投入.....	65



# 1. 自動判別ツール(プロトタイプ)の概要

## 1.1. 背景・目的

### 1.1.1. 背景

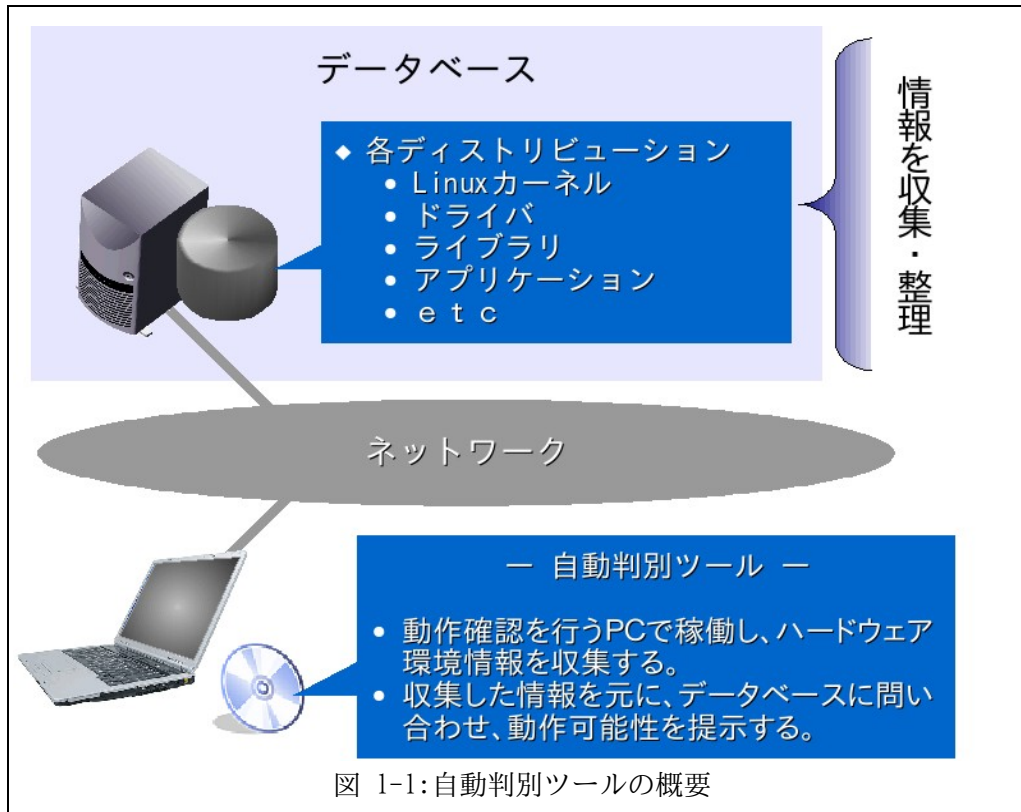
OSS の導入が進まない原因のひとつとして、Linux ディストリビューション毎の細かな差異、Linux カーネルバージョンの差やデバイスドライバの対応状況による動作環境の違いといった Linux の多様性が挙げられている。これらに関してはその情報を得ることも簡単ではなく、また Linux が動作するハードウェアも多種多様であることから、実際にそのハードウェアでどのディストリビューションが利用できるのか確認することが困難な状況がある。

### 1.1.2. 目的

本事業では、Linux ディストリビューションの存在や特徴を共有する方法を確立し、多くの人がそれを容易に知り得る状態を作り出すことを目的として、Linux ディストリビューションの仕様を記述する様式を検討する。さらにその様式を利用して利用者が用意したハードウェアに対して利用可能な Linux ディストリビューションについて選択肢と導入方法を提示したうえでさらに利用者がその中から好みのものを容易に選択できる環境の実現について検討するために、Linux ディストリビューションの利用可能性を自動的に判断できるような仕組みを検証するためのプロトタイプ(自動判別ツール)を開発する。

## 1.2. 自動判別ツールとデータベースについて

自動判別ツールとは、利用可能性を判断する対象のハードウェア環境で実行することで、自動的に利用可能な Linux ディストリビューションの選択肢を、ツールの利用者に提示ものである。



自動判別ツールで使用する情報をデータベース化し、本ツール以外からも広く利用できるようにして、継続的にサービスを維持管理できるようにする。

### 1.3. プロトタイプに必要な機能

#### 1.3.1. ハードウェア自動判別機能

- ◆ 本ツールは、CD 等の可搬可能なメディアやデバイスから起動する GNU/Linux に同梱され、容易に利用することが可能である。
- ◆ ネットワークに対応したデータベースに接続するために、本ツールが同梱される GNU/Linux に、簡単な操作でネットワーク設定を行うことができる機能を実装する。
- ◆ 利用可能性を判断する必要があるハードウェア環境で、自動的にハードウェア情報を収集する機能を実装する。
- ◆ 自動収集対象ハードウェアは、PCI や USB、ファイヤーワイヤー等に接続された、ソフトウェアから検出することが可能な機器を対象とする。
- ◆ 収集したハードウェア情報を、一覧もしくは個別にディスプレイに表示する機能を有する。
- ◆ 収集したハードウェア情報を元にデータベースに問い合わせを行い、利用することが可能

であろうディストリビューション情報を一覧もしくは個別にディスプレイに表示する機能を実装する。

- ◆ 利用可能性を判断する必要があるハードウェア環境で、簡単な利用試験を自動的に行い、本ツールが同梱される GNU/Linux で使用することが可能かどうか判断することができる機能を実装する。これにより、自動判別結果データの信頼性を上げることが期待できる。
- ◆ 利用試験結果を、一覧もしくは個別にディスプレイに表示する機能を有する。
- ◆ ハードウェア自動判別結果を、USB メモリやネットワークに対応したデータベースに格納することができる機能を実装する。これにより、データベースに格納されるデータの精度を上げることが期待できる。

### 1.3.2. アプリケーション提示機能

- ◆ 自動判別ツール利用者との簡単な対話機能を実装し、ユーザの要求を満たすであろう各ディストリビューションで稼動可能なアプリケーションについて一覧もしくは個別にディスプレイに表示する機能を実装する。
- ◆ 対話内容や提示するアプリケーションについての情報をデータベース化し、実装するツールが有するロジックと情報を分離し、提示可能な内容について鮮度を保つように実装する。
- ◆ アプリケーション提示結果を、USB メモリやネットワークに対応したデータベースに格納することができる機能を実装する。

### 1.3.3. Linux ディストリビューション情報の自動更新

Linux ディストリビューションに収録されているドライバやアプリケーション情報の自動更新を行う。これは、自動判別機能で利用するデータベースに保存する情報の鮮度を簡易に保つために必要な機能である。

自動更新のタイミングは、Linux ディストリビューションのソフトウェアがバージョンアップされた場合や、新たな Linux ディストリビューションをサポートする場合にデータベースの更新を行うようにする。自動更新を適用できない情報も考えられるため、手動にて更新する事も考慮する。

## 1.4. 対象とするハードウェア機器

ハードウェアの自動判別を行う場合、対象とするハードウェア機器は、次の表に示すものとする。

表 1-1:対象とするハードウェア機器の一覧

ハードウェア機器名	説明
Mother Board	PC を構成するための電子回路基盤
CPU	中心的な回路(中央処理装置,中央演算処理装置)
Memory (DIMM)	情報の記録を行う装置(主記憶装置)
Storage Controller	補助記憶装置(ハードディスク)を制御する装置
Video	PC で映像を信号として出力または入力するための電子基板
Network(Wired, Wireless)	PC で通信を行うための装置
Capture	映像や画像を取り込み保存する装置
Sound	PC において音声の入出力機能を持つ装置
PCMCIA Card	PC 用小型カード型インターフェース、およびその規格による拡張カード
Printer*1	印刷装置
Scanner*1	PC に画像や文書などをデジタル静止画像情報化するための機器

\*1. USB 接続のみを対象とする。

## 1.5. 対象とするディストリビューション

本調査で対象とする Linux ディストリビューションは、次の表に示すものとする。

表 1-2:対象とする Linux ディストリビューション

Linux ディストリビューション	説明
Turbolinux FUJI	パッケージ管理に RPM を使用している商用ディストリビューション  <a href="http://www.turbolinux.co.jp/">http://www.turbolinux.co.jp/</a>
KNOPPIX 5.1.1	ドイツの Klaus Knopper が Debian パッケージを元に開発しており、日本語版は独立行政法人産業技術総合研究所が日本語化をはじめとする、日本の国情にあわせた様々な機能を追加して配布を行っているディストリビューション  <a href="http://unit.aist.go.jp/itri/knoppix/">http://unit.aist.go.jp/itri/knoppix/</a>
Ubuntu 7.04	Debian GNU/Linux をベースにしたマーク・シャトルワースが創業した Canonical Ltd. から支援を受けているディストリビューション  <a href="http://www.ubuntulinux.jp/">http://www.ubuntulinux.jp/</a>
Fedora 7	RPM 系ディストリビューションのひとつで、Red Hat が後援するディストリビューション

	<a href="http://fedoraproject.org/">http://fedoraproject.org/</a>
OpenSUSE 10.2	RPM 系ディストリビューションのひとつで、Novell が支援するディストリビューション  <a href="http://ja.opensuse.org/">http://ja.opensuse.org/</a>

## 2. 自動判別ツール(プロトタイプ)の機能

### 2.1. システム機能概要

システムイメージを次に示す。各機能に関する概要は、次項に示す。

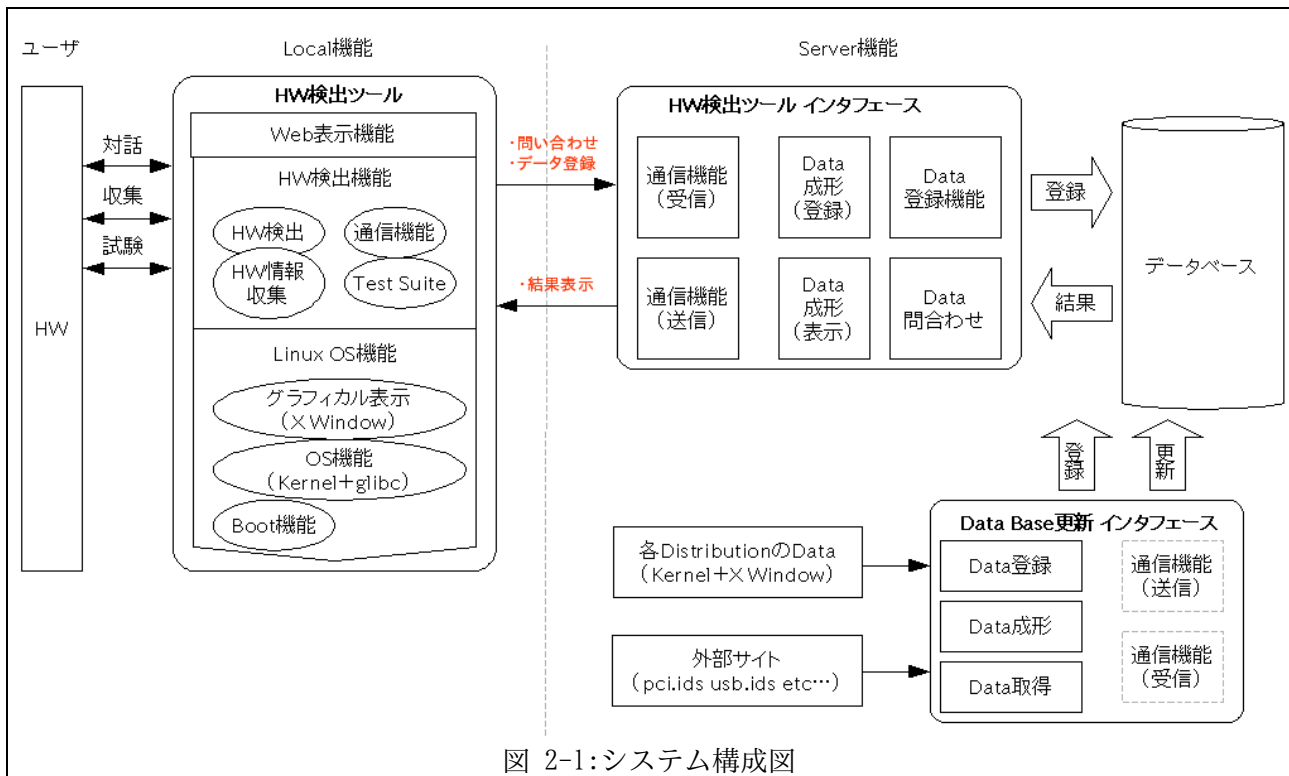


図 2-1:システム構成図

### 2.2. Hardware 機器のサポート情報自動判別

下記の各機能は、PCにCD-ROMを挿入することで利用できるLocal側の機能となる。

#### 2.2.1. ハードウェア機器の検出機能

PCに実装されているハードウェア機器を自動的に検出して、検出した結果を保存する機能を提供する。

#### 2.2.2. 通信機能

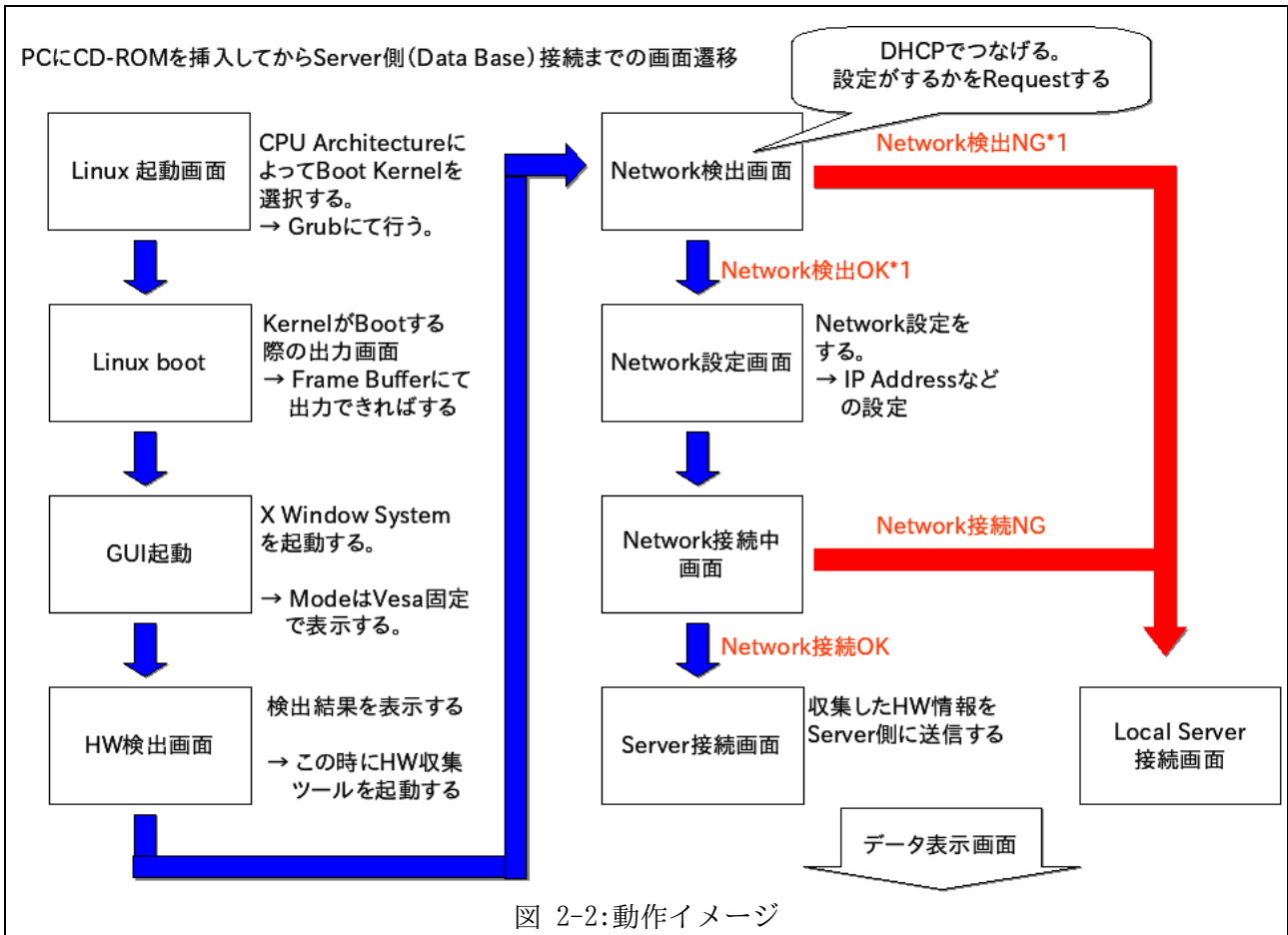
ハードウェア機器の検出機能にて検出された結果をサーバー側へ送信し、サーバー側からの検索結果を受信する機能を提供する。

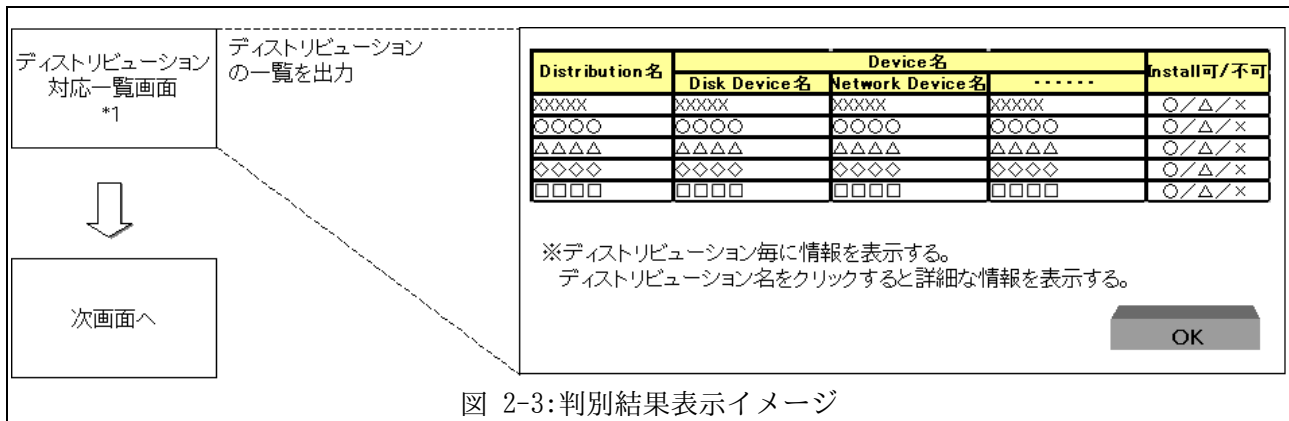
### 2.2.3. 画面表示機能

ハードウェア機器の検出やサーバー側との通信時に利用するユーザーインターフェースを GUI (Graphical User Interface)として提供するために、Window System を利用する。

### 2.2.4. 動作イメージ

各機能は、次の図に示すような動作となる。





## 2.3. データベース・メンテナンス機能

下記の各機能は、サーバー側に実装される機能である。

### 2.3.1. データベース自動投入データ生成機能

各 Linux ディストリビューションからハードウェア対応情報に必要となるデータを収集して、データベースに自動投入するためのデータ成形を提供する。

### 2.3.2. データベース更新/追加機能

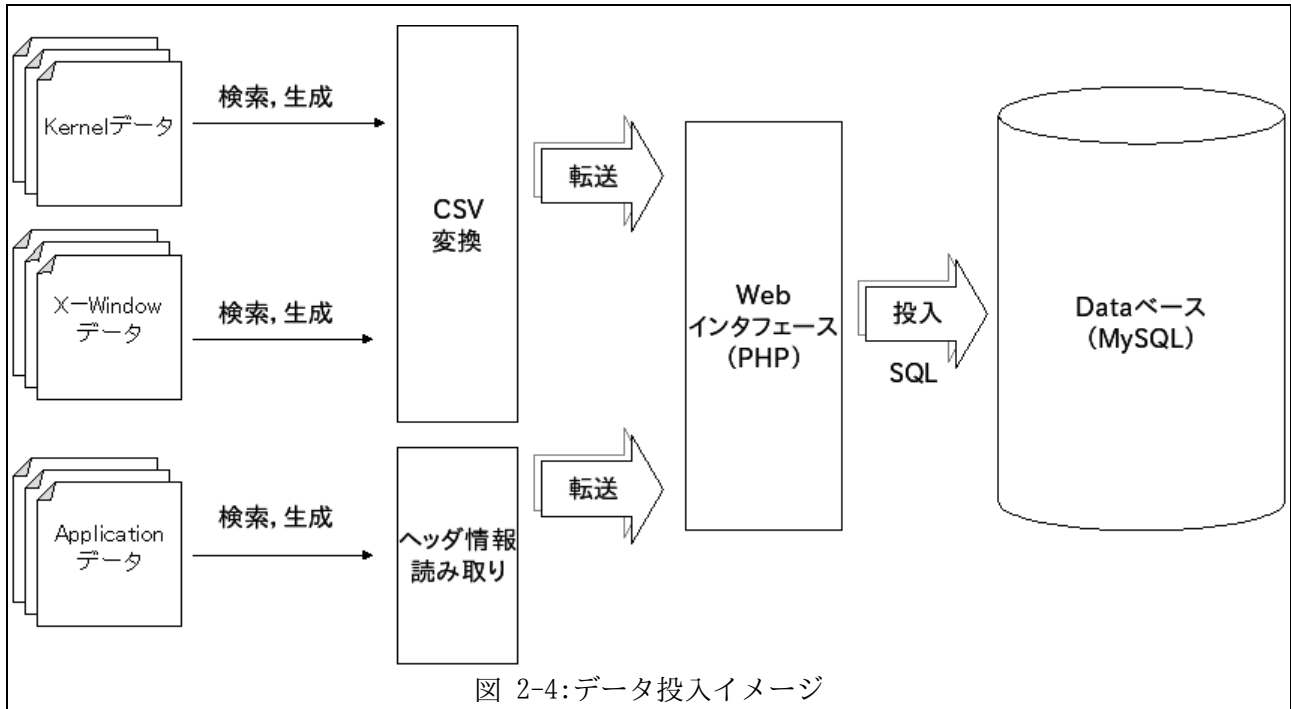
自動投入データ生成機能からのデータの投入や、必要な情報を手動にてデータベースに投入する機能を提供する。

### 2.3.3. データベース検索機能

投入されているデータのキーワード検索や、ローカル側の自動判別ツールからのデータを自動検索する機能を提供する。

### 2.3.4. 動作イメージ

各機能は、次の図に示すような動作となる。



## 3. ハードウェア検出機能詳細

### 3.1. ブート機能

#### 3.1.1. 利用する Linux Kernel

Kernel のバージョンは、利用する PC システムのハードウェア・アーキテクチャに依存する。Kernel のバージョンを決定するには、対象とするマザーボードのアーキテクチャを策定する必要がある。また、本事業に関連する Help PC の要件では、古い PC をリユースできる事を判別できることも必須となる。

対象となるマザーボードのアーキテクチャとリユース対象とする PC スペックを策定するため、目安として、次の表に記載されたスペックの PC にて立ち上がる事を前提に Kernel バージョンを決定する事とする。また、最新の PC システムのアーキテクチャにも対応することも求められる。

表 3-1:自動判別対象 PC の最低スペック

サポート情報	マザーボード	CPU	メモリ	ネットワーク機器
最低スペック	440BX 以上	Pentium II 相当	128M バイト以上	NE2000 互換

#### 3.1.2. ブート機器 (デバイス)

確実に PC システムにてブートする事が必要となる。この場合には、PC システムにてブートするハードウェア機器の選択が重要となる。

PC システムのブートを許容しているハードウェア機器としては、CD-ROM (R/RW (DVD-ROM 含む) 機器と、ディスク機器 (ハードディスク, Flash メモリ) がある。接続形態としては、内部接続 (IDE), 外部接続 (USB/IEEE) がある。この形態より、PC システムで最もブートが確実に行えるのは、「内部接続・CD-ROM/ハードディスク」になると考えられる。今回の要件では、ブートが確実に行える方法として、「内部接続・CD-ROM」を選択し、CD-ROM より確実にブートして、メモリ上にて動作するような機能とする。

#### 3.1.3. ブートの仕組み

CD イメージでの提供とするため、isolinux の仕組みを利用して PC を Boot Up する。今回は、ディストリビューションとして Turbolinux を利用し、CD-ROM を作成する。



図 3-1:起動画面

### 3.1.4. ルートツリー

本事業のプロトタイプ要件では、次の表に示す機能が必要となる。この要件を満たすためにルートツリー(/)を用意し、各種ソフトウェアが利用できるようにする。(ルートツリーの詳細は、付録を参照のこと。)

表 3-2:プロトタイプで利用する主なソフトウェア

機能	利用ソフトウェア	備考
Ethernet 通信	Kernel Driver	
IP アドレス設定	ネットワーク設定	
Web ブラウザ	Fire Fox	
Data Base 機能	My SQL	5.0.37
Data Base Interface	PHP	5.2.1
日本語入力	SCIM	
Web サーバ	Apache	2.0.51

## 3.2. 画面表示機能

### 3.2.1. 利用する X Window システム

最新機器に対応するには、最新版の X Window システムが必要となるが、今回の要件としては、X Window を介して Firefox にてブラウジングができることが目的のため、最新版は適用せずに、安定している X Window システムを利用することとする。

### 3.2.2. Window マネージャ

起動時間の短縮を図るため、軽量な Window マネージャを利用することとし、「Xfce」という Window マネージャを利用する。



図 3-2: デスクトップ画面

## 3.3. ハードウェア機器検出ソフトウェア

### 3.3.1. ハードウェア機器の検出

Kernel には、Kernel が起動した際に、PC システムのハードウェア機器を検出し、そのハードウェア機器に合った Kernel ドライバをロードする機能がある。

Kernel が検出した内容をデータ化して、Supported/Unsupported の状態を判別できることが必要となる。プロトタイプでは、「lshw」というソフトウェアを利用し、ハードウェア機器の情報や Kernel が検出した結果などをテキストデータに変換する。

参照 URL : [HardwareLiSter - ezIX](http://ezix.org/project/wiki/HardwareLiSter)

<http://ezix.org/project/wiki/HardwareLiSter>

「lshw」の詳細に関しては、上記 URL にドキュメントされている。本仕様書では、詳細情報に関する事柄は言及しないので、上記 URL での配布ドキュメントを参照することとする。

### 3.3.2. ハードウェア機器の検出データ構造

「lshw」では、PCシステムのH/W機器情報や、Kernelが検出した情報をいろいろな形にて表示することができる。プロトタイプでは、サーバー側のデータベースを利用して、Linux Kernelの情報を検索する事が求められる。データベース側にて検索を行い易いように、検出したデータを構造化するため、XML構造のデータを利用することとする。

表 3-3:検出データ XML 構造

利用されるタグ名	タグの説明	備考
<node>	検出した Hardware データ 始まり	
<description>      </description>	機器種別	
<product>            </product>	ハードウェア機器名	
<vendor>              </vendor>	ハードウェアベンダー名	
<physid>              </physid>	物理 ID	☆
<businfo>             </businfo>	PCI バス番号	
<logicalname>        </logicalname>	Kernel 上のデバイス名	Logical な名前がない場合は ない。 ☆
<dev>                 </dev>	/dev 配下の番号	/dev を利用しないものは表示 しない。 ※
<version>             </version>	Hardware 機器の Version	☆
<serial>              </serial>	シリアル番号	機器により内容が異なる ※
<size                  units= >            </size>	サイズ	機器により内容が異なる ※
<capacity             units= >            </capacity>	容量	機器により内容が異なる ※
<width                 units= >            </width>	Bus の bit 幅	
<clock                 units= >            </clock>	PCI バスのクロック	
<configuration> <setting id=     /> </configuration>	Kernel, Device の設定値 設定値	
<capabilities> <capability     id= >            </capability> </capabilities>	ハードウェアがサポートして いる機能 Driver 名など	Kernel 上の Driver 名も表 示される。 ※
<resources> <resource       type=                /> </resources>	PCI 上のリソース情報	
<pcibus_info> <pci             id=                 /> </pcibus_info>	PCI バスの ID 情報	Vendor ID,Device ID など。
<usb_info> <usb            id=                 /> </usb_info>	USB バスの ID 情報	Vendor ID,Product ID など。
</node>	検出した Hardware データ の終わり	

\* Hardware 機器によって値が変化する。

☆ タグの用途として、内容が不明であり、調査中のもの。

## 3.4. 通信機能

### 3.4.1. Ethernet 通信

IP のプロトコルを利用し、Ethernet デバイスを介した通信を行う事が必要となる。このため、Ethernet デバイスの検出/設定を行い、IP アドレスが動的/静的に設定できることとする。また、IP レベルでは、DNS/ルートルーティングデバイス、アドレスの設定も必要となるため、設定するためのユーザインタフェースを提供し、各種のネットワーク設定ができるようにする。

### 3.4.2. サーバー側との通信プロトコル

Ethernet での通信を介して、TCP による通信をサーバー側と実施する。この際に利用するプロトコルとして、HTTP プロトコルを利用して、サーバー側との通信をすることとする。サーバー側との通信データ形式は、全てテキスト形式で通信する。

### 3.4.3. サーバーとのデータ通信

ハードウェア機器の検出したデータをサーバー側に送信し、サーバー側にて受信したデータを利用して、データベースに登録されているかを検索する。検索結果は、Local 側にて受信を行い表示することが必要となる。

この機能を実現するため、「curl」というソフトウェアを利用して、サーバー側とのデータ通信を実現する。(詳細は付録を参照)

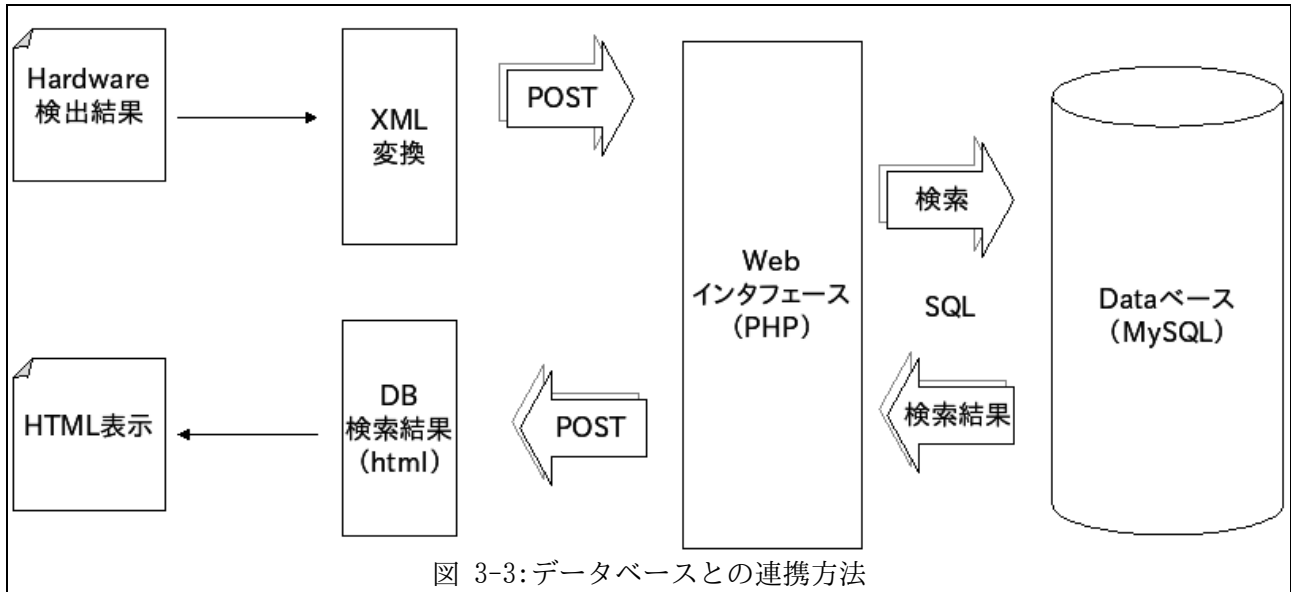


図 3-3: データベースとの連携方法

## 3.5. ユーザインタフェース機能

### 3.5.1. 利用するインタフェース

ユーザとの対話を実現するインタフェースとしては、Web ベースのインタフェースを利用する。Web 表示を実現するため、Web ブラウザを利用する。Linux で一般的に利用されている Firefox を利用し、Web ベースのインタフェースを利用する。

### 3.5.2. その他ユーザへのインタフェース

サーバー側への問い合わせ、ハードウェアの検出時などの待ち状態ができた場合に、プログレスバーを表示することによって、待ち合わせをユーザーに知らせる。

## 4. データベース機能

### 4.1. データベース機能

データベース機能を実現するために、MySQLというデータベースソフトウェアを利用する。このソフトウェアは、サーバー側に配置するが、Local 側がサーバー側と Ethernet 通信できなかった場合には、Local 側のみで利用できるようにする必要があるため、Local 側にも配置する。

MySQLにてデータベースに持つテーブルの種類は、下記の項に示す。その他で利用が必要なテーブルがあった場合には、必要に応じてデータの追加をする。

(データベース構造についての詳細な情報は、データベース仕様書を参照)

#### 4.1.1. Linux Kernel テーブル(モジュール)

Linux Kernel で利用できるハードウェア機器の情報を一覧化したものをテーブル化する。このテーブルは、Linux Kernel 上でモジュール化(.ko)したものが対象となる。一覧化したテーブルの他に、個々のデバイスドライバの情報もテーブル化する。

##### (1) PCI バスの情報

/lib/modules/‘kernel ver’/に配置されている PCI バスのマップ情報を利用し、データを生成する。

pci module 名	vendor id	device id	sub vendor id	sub device id	class id	class_mask	driver_data
-----------------	--------------	--------------	------------------	------------------	----------	------------	-------------

##### (2) USB バスの情報

/lib/modules/‘kernel ver’/に配置されている USB バスのマップ情報を利用し、データを生成する。

USB module 名	Match flag	Vendor id	Products id	bcbDevice_ lo	bcdDevice_ hi	bDeviceCla ss
bDeviceSub Class	bDeviceProt ocol	bInterfaceCl ass	bInterfaceSub Class	bInterfaceProt ocol	Driver_in fo	

##### (3) ドライバ情報

/lib/modules/‘kernel ver’/kernel/drivers/に配置されている\*.ko ファイルより各種ドライバの情報を収集し、データを生成する。

Author	Description	License	Version	Vermagic	Depends	Alias
--------	-------------	---------	---------	----------	---------	-------

alias 情報は、次に示すような構造がある。

バス種別	意味								
PCI	vendor id	device id	sub vendor id	sub device id	base class	sub class	interface		
USB	vendor id	product id	device	device class	device sub class	device protocol	interface class	interface sub class	interface protocol
PCMCIA	manfactor id	card id	function id	function	device no	product id1	product id2	product id3	product id4
IEEE1394	vendor id	model id	specifier id	verison					
Serial	type	protocol	id	extra					
PNP	id								

図 4-1:alias 情報

alias 行は、ハードウェア機器が接続されているバス種別によって意味が異なる。

## 4.1.2. Linux Kernel テーブル (vmlinuz)

Linux Kernel では、モジュール化(.ko)を行わず、ブートする Kernel イメージに含まれている情報もある。この情報を vmlinuz より抜き出すことで、対応情報を生成する。

## 4.1.3. X Window テーブル

X Window System が持っているハードウェア機器の情報を一覧化したものをテーブル化する。また、モニタ情報もテーブル化する。

## 4.1.4. デバイス名情報テーブル

ハードウェア機器の情報を一覧化した際に、一般的な名前との関連付けが必要となる。このため、ハードウェア機器の一般的な命名規則をテーブル化する。この情報は、Linux にて一般的利用されている“pci.ids”、“usb.ids”などハードウェア機器の ID がまとまっているものを利用し、テーブル化する。

### (1)PCI ID のテーブル

Vendor 名	Device 名	Vendor ID	Device ID	Sub Vendor ID	Sub Device ID
Device 種別	Sub Device 種別	Interface 種別	Class ID	Sub Class ID	Interface

## (2)USB ID のテーブル

Vendor 名	Product 名	Vendor ID	Product ID	Interface
Device 種別	Sub Device 種別	Interface 種別	Class ID	Sub Class ID
				Interface

## 4.1.5. アプリケーションソフトウェアテーブル

アプリケーションソフトウェアの情報は、ディストリビューション毎に取得できる内容が異なる。このため、統一したフォーマットを定義し、このフォーマットをデータベースに登録する。次にデータベースに登録する際のフォーマットを示す。

	DataBase上での名前	CSVのKEY
共通ヘッダ	パッケージ名	NAME
	パッケージカテゴリ	Group
	パッケージサイズ	Size
	メンテナンス情報	Packager
	アーキテクチャ	Architecture
	ソースパッケージ名	Source RPM
	バージョン	Version
	シグネチャキー情報	Signature
固有ヘッダ	パッケージの説明	Description
	Distribution_info1	それぞれの名前
	Distribution_info2	
	Distribution_info3	
	Distribution_info4	
	Distribution_info5	
	Distribution_info6	
Distribution_info7		
共通ヘッダ	インストールディレクトリ情報	Install list
	依存情報	Require list

図 4-2: データベースへの登録フォーマット

	Knoppix	Ubuntu	Fedora	OpenSUSE	Turbolinux	
共通ヘッダ	Package	Package	Name	Name	Name	
	Section	Section	Group	Group	Group	
	Install-Size	Install-size	Size	Size	Size	
	Maintainer	Maintainer	Packager	Packager	Packager	
	Architecture	Architecture	Architecture	Architecture	Architecture	
	Source	Source	Source RPM	Source RPM	Source RPM	
	Version	Version	Version	Release	Version	Release
	Signature	Signature	MD5sum SHA1 SHA256	Signature	Signature	Signature
	Description	Description	Description	Description	Description	Description
固有ヘッダ	Provides	Provides	URL	URL	URL	
	Priority	Priority	Summary	Summary	Summary	
	Depends	Conflicts	Build Data	Vendor	Vendor	
	Conflicts	Bugs	Build Host	Build Data	Build Data	
	Status	Origin	Task	Build Host	Build Host	
	Replaces	Original-Maintainer	License	License	License	
	License	Filename	Size	Relocations	Relocations	
	Relocations	Relocations	Relocations	Relocations	Relocations	
共通ヘッダ	Install Directory	Install Directory	Install Directory	Install Directory	Install Directory	
	Requires list	Requires list	Requires list	Requires list	Requires list	

図 4-3:各ディストリビューションから得られる情報

## 4.1.6. 検出結果テーブル

Local 側で収集したハードウェアの検出データをテーブル化する。これは Local 側から送信されてくる XML データを登録する。

## 4.2. データベースインタフェース機能

データベースへのデータの投入については、自動的に投入する形態と手動で投入する形態を用いる。投入されたデータは、規則に従い検索できるようにする。データベースのメンテナンス性を考え、追加／削除もできることとする。

### 4.2.1. データ自動投入

Kernel 情報,X Window System 情報,アプリケーション情報の収集を自動的に行い、CSV データの形式でデータを投入する。データの自動投入にあたっては、(1),(2)に示すような情報収集の機能が必要となる。※現段階では、(1),(2)とする。

#### (1)ソフトウェア情報収集

前項にて示した各テーブルを作成するための基情報を作成する。Kernel,X Window System が持っている情報を収集して、CSV 形式のデータを生成する。

#### (2)データ投入

(1)にて生成された CSV データをデータベースのテーブルに投入する。

(詳細な手順は、APPENDEX C を参照のこと。)

### 4.2.2. データ手動投入

自動的に投入されたデータに付帯情報を追加するために用いる。自動投入された情報を検索し、必要となる情報を入力するために、レコード／カラムの追加を行うこととする。

### 4.2.3. データ追加

自動投入データにて生成されたデータを基に、データベースを検索して差分抽出して、差分データをデータベースに投入するかを選択できることとする。

### 4.2.4. データ削除

ソフトウェアのバージョンアップに伴い、不要となったデータを削除する。削除にあたっては、前項のデータ追加時に検索したデータから、削除が選択できることとする。

### 4.2.5. データ検索

投入された情報を規則に従い検索できるようにする。検索にあたっての規則は、(1),(2)に示す規則とする。※現段階では、(1),(2)とする。

#### (1) Local 側からのデータの検索

前章で解説したハードウェア情報の収集機能より送信されてくるデータから、データベースに投入されたデータを検索する。検索した結果として、データベースに登録／未登録の情報を結果として表示できるようにする。

#### (2) Web 上でのデータ検索

一般的な検索サイトと同様に、文字列での検索ができるような機能とする。検索にあたっての規則としては、「PCI ID」、「USB ID」、「デバイス名」にての検索が可能なものとする。

## 付録 A. ツリー構造

### A.1. ルートツリー構造

本プロトタイプで利用した Turbolinux の CD-ROM を作成する場合のルートツリー構造を次に示す。

```
root-tree/
|-- bin
|-- boot
|-- dev
|   |-- amdtp
|   |-- ataraid
|   |-- carmel
|   |-- cciss
|   |-- cpu
|       |-- 0
|       |-- 1
|       |-- 10
|       |-- 11
|       |-- 12
|       |-- 13
|       |-- 14
|       |-- 15
|       |-- 2
|       |-- 3
|       |-- 4
|       |-- 5
|       |-- 6
|       |-- 7
|       |-- 8
|       |-- 9
|   |-- dri
|   |-- dvl394
|   |-- i2o
|   |-- ida
|   |-- input
|   |-- pts
|   |-- raw
|   |-- raw1394
|   |-- rd
|   |-- shm
|   |-- snd
|   |-- usb
|   |-- vc
|   |-- video
|   |-- videol394
|-- etc
|   |-- X11
|       |-- app-defaults
|       |-- applnk
|       |-- dm
|           |-- Sessions
|       |-- fs
|       |-- gdm
|           |-- Sessions
|       |-- lbxproxy
|       |-- proxymngr
|       |-- rstart
|           |-- commands
|               |-- x -> x11r6
|               |-- x11 -> x11r6
|               |-- x11r6
|   |-- contexts
|   |-- serverconfig
|   |-- starthere
|   |-- sysconfig
|   |-- twm
|   |-- wmconfig
|   |-- wmsession.d
|   |-- xdm
|       |-- pixmaps
|   |-- xinit
|       |-- xinitrc.d
|       |-- xinput.d
|   |-- xkb
|       |-- compat
|       |-- compiled -> ../../../../var/lib/xkb
|       |-- geometry
|           |-- digital
|           |-- ibm
|           |-- sgi
|       |-- keycodes
|           |-- digital
|           |-- sgi
|       |-- keymap
|           |-- digital
|           |-- sgi
|           |-- sun
|       |-- rules
|       |-- semantics
|       |-- symbols
|           |-- digital
|           |-- fujitsu
|           |-- hp
|           |-- macintosh
|           |-- nec
|           |-- pc
|           |-- sgi
|           |-- sony
|           |-- sun
|           |-- xfree68
|       |-- types
|   |-- xserver
|   |-- xsm
|-- acpi
|   |-- actions
|   |-- events
|-- alternatives
|-- bonobo-activation
|-- cron.daily
|-- cron.hourly
|-- cron.monthly
|-- cron.quarter-daily
|-- cron.weekly
|-- default
|-- dev.d
|   |-- default
|   |-- dxr3
```

```

|-- dhcpd
|-- dynaplug
  |-- template
|-- fonts
  |-- conf.d
|-- gconf
  |-- 2
  |-- gconf.xml.defaults
    |-- desktop
      |-- gnome
        |-- accessibility
          |-- keyboard
          |-- startup
        |-- applications
          |-- browser
          |-- component_viewer
          |-- help_viewer
          |-- terminal
          |-- window_manager
        |-- background
        |-- file_views
        |-- interface
        |-- peripherals
          |-- keyboard
          |-- mouse
        |-- sound
        |-- thumbnailers
        |-- typing_break
        |-- url-handlers
          |-- aim
          |-- callto
          |-- ghelp
          |-- h323
          |-- http
          |-- https
          |-- info
          |-- mailto
          |-- man
          |-- trash
      |-- schemas
        |-- desktop
          |-- gnome
            |-- accessibility
              |-- keyboard
              |-- startup
            |-- applications
              |-- browser
              |-- component_viewer
              |-- help_viewer
              |-- terminal
              |-- window_manager
            |-- background
            |-- file-views
            |-- interface
            |-- peripherals
              |-- keyboard
              |-- mouse
            |-- sound
            |-- thumbnailers
            |-- typing_break
            |-- url-handlers
              |-- aim
              |-- callto
              |-- ghelp
              |-- h323
              |-- http
              |-- https
              |-- info
              |-- mailto
              |-- man

```

```

|-- trash
  |-- system
    |-- http_proxy
    |-- proxy
  |-- system
    |-- http_proxy
    |-- proxy
  |-- gconf.xml.mandatory
  |-- schemas
-- ggw
  |-- kbd
-- gnome-vfs-2.0
  |-- modules
-- gtk
-- gtk-2.0
  |-- i386-pc-linux-gnu
-- heritage.d
-- hotplugd
-- icalicense
-- iim
-- init.d -> rc.d/init.d
-- iproute2
-- logrotate.d
-- makedev.d
-- master
  |-- etc
    |-- X11
    |-- xinit
  |-- pcmcia
  |-- sysconfig
  |-- udev
    |-- rules.d
  |-- xdg
    |-- xfce4
      |-- mcs_settings
      |-- panel
      |-- xfwm4
      |-- xfce4-session
  |-- usr
    |-- share
      |-- pixmap
      |-- themes
        |-- Default
        |-- balou
-- murasaki
-- opt
-- pam.d
-- pango
  |-- i686-pc-linux-gnu
-- pcmcia
  |-- cis
-- ppp
  |-- peers
  |-- rp-pppoe-gui
-- profile.d
-- racoon
  |-- certs
-- rc.d
  |-- init.d
  |-- rc0.d
  |-- rc1.d
  |-- rc2.d
  |-- rc3.d
  |-- rc4.d
  |-- rc5.d
  |-- rc6.d
  |-- rc7.d
  |-- reprex.d
  |-- reinit.d
-- rc0.d -> rc.d/rc0.d

```

```

|-- rc1.d -> rc.d/rc1.d
|-- rc2.d -&; rc.d/rc2.d
|-- rc3.d -> rc.d/rc3.d
|-- rc4.d -> rc.d/rc4.d
|-- rc5.d -> rc.d/rc5.d
|-- rc6.d -> rc.d/rc6.d
|-- rc7.d -> rc.d/rc7.d
|-- rpm
|   |-- athlon-linux
|   |-- i686-linux
|-- scim
|-- security
|   |-- console.apps
|-- skel
|-- sound
|   |-- events
|-- ssh
|-- sysconfig
|   |-- apm-scripts
|   |-- cbq
|   |-- console
|   |-- distribution
|   |-- network-scripts
|-- udev
|   |-- agents.d
|   |   |-- usb
|   |-- conf.d
|   |-- rules.d
|   |-- scripts
|-- xdg
|   |-- xfce4
|   |   |-- desktop
|   |   |-- mcs_settings
|   |   |-- panel
|   |   |-- xfce4-session
|   |-- xinetd.d
-- home
  |-- turbo
-- initrd
-- lib
|-- firmware
-- modules
|   |-- 2.6.19 -> 2.6.19-5.2
|   |-- 2.6.19-5.2
|       |-- build
|       |-- kernel
|           |-- arch
|           |   |-- i386
|           |   |-- kernel
|           |   |-- cpu
|           |       |-- cpufreq
|           |       |-- mcheck
|           |-- crypto
|           |-- drivers
|               |-- acpi
|               |-- ata
|               |-- base
|               |-- block
|               |-- bluetooth
|               |-- cdrom
|               |-- char
|                   |-- agp
|                   |-- drm
|                   |-- ipmi
|                   |-- tpm
|               |-- connector
|               |-- cpufreq
|               |-- firmware
|               |-- ide
|               |-- legacy

```

```

  |-- pci
  |-- ieee1394
  |-- input
  |   |-- gameport
  |   |-- joystick
  |   |   |-- iforce
  |   |-- misc
  |   |-- mouse
  |   |-- serio
-- md
-- media
  |-- video
  |   |-- cpia2
  |   |-- et61x251
  |   |-- pwc
  |   |-- sn9c102
  |   |-- usbvideo
  |   |-- zc0301
-- message
  |-- fusion
-- misc
-- mmc
-- net
|   |-- bonding
|   |-- chelsio
|   |-- e1000
|   |-- ixgb
|   |-- myri10ge
|   |-- sk98lin
|   |-- tulip
|   |-- wireless
|       |-- bcm43xx
|       |-- hostap
|       |-- prism54
|       |-- zd1211rw
-- parport
-- pcmcia
-- scsi
|   |-- aacraid
|   |-- aic7xxx
|   |-- aic94xx
|   |-- arcmsr
|   |-- libsas
|   |-- lpfc
|   |-- megaraid
|   |-- pcmcia
|   |-- qla2xxx
|   |-- qla4xxx
|   |-- sym53c8xx_2
-- serial
-- usb
|   |-- class
|   |-- input
|   |-- net
|   |-- storage
  |-- video
  |   |-- backlight
-- fs
|   |-- aufs
|   |-- autofs
|   |-- autofs4
|   |-- cifs
|   |-- configfs
|   |-- cramfs
|   |-- exportfs
|   |-- ext4
|   |-- fat
|   |-- fuse
|   |-- isofs
|   |-- jbd2

```

```

|-- lockd
|-- minix
|-- msdos
|-- nfs
|-- nfs_common
|-- nfsd
|-- nls
|-- ntfs
|-- reiserfs
|-- romfs
|-- smbfs
|-- squashfs
|-- udf
|-- ufs
|-- unionfs
|-- vfat
|-- lib
|-- zlib_deflate
|-- zlib_inflate
|-- net
|-- 8021q
|-- bluetooth
|-- bnep
|-- hidp
|-- rfcomm
|-- ieee80211
|-- softmac
|-- ipv4
|-- ipv6
|-- key
|-- sunrpc
|-- auth_gss
|-- xfrm
|-- sound
|-- core
|-- oss
|-- seq
|-- oss
|-- drivers
|-- mpu401
|-- pci
|-- ac97
|-- hda
|-- usb
|-- misc
|-- nptl
|-- security
|-- pam_filter
|-- tls -> nptl
-- lost+found
-- media -> mnt
-- mnt
|-- camera
|-- cdrom
|-- floppy
|-- player
-- opt
-- proc
-- root
-- sbin
-- selinux
-- srv
-- sys
-- tmp
-- usr
|-- X11R6
|-- bin
|-- doc
|-- include
|-- X11

```

```

|-- bitmaps
|-- pixmap
|-- lib
|-- X11
|-- app-defaults -> /etc/X11/app-
defaults
|-- doc
|-- PDF
|-- PostScript
|-- html
|-- etc
|-- fonts
|-- 75dpi
|-- CID
|-- Speedo
|-- TTF
|-- Type1
|-- encodings
|-- large
|-- extra
|-- local
|-- misc
|-- util
|-- fs -> ../../../../etc/X11/fs
|-- getconfig
|-- icons
|-- default
|-- handhelds
|-- cursors
|-- redglass
|-- cursors
|-- whiteglass
|-- cursors
|-- ja_JP -> ja_JP.eucJP
|-- ja_JP.UTF-8
|-- app-defaults
|-- ja_JP.eucJP
|-- app-defaults
|-- ko_KR -> ko_KR.eucKR
|-- ko_KR.eucKR
|-- app-defaults
|-- lbxproxy -> /etc/X11/lbxproxy
-- locale
|-- C
|-- armSCII-8
|-- common
|-- el_GR.UTF-8
|-- en_US.UTF-8
|-- georgian-academy
|-- georgian-ps
|-- ibm-cp1133
|-- iscii-dev
|-- isiri-3342
|-- iso8859-1
|-- iso8859-10
|-- iso8859-11
|-- iso8859-13
|-- iso8859-14
|-- iso8859-15
|-- iso8859-2
|-- iso8859-3
|-- iso8859-4
|-- iso8859-5
|-- iso8859-6
|-- iso8859-7
|-- iso8859-8
|-- iso8859-9
|-- iso8859-9e
|-- ja
|-- ja.JIS

```

```

-- ja_SJIS
-- ja_JP.UTF-8
-- ko
-- ko_KR.UTF-8
-- koi8-c
-- koi8-r
-- koi8-u
-- lib
  |-- common
-- microsoft-cpl251
-- microsoft-cpl255
-- microsoft-cpl256
-- mulelao-1
-- nokhchi-1
-- pt_BR.UTF-8
-- tatar-cyr
-- th_TH
-- th_TH.UTF-8
-- tscii-0
-- vi_VN.tcvn
-- vi_VN.viscii
-- zh_CN
-- zh_CN.UTF-8
-- zh_CN.gb18030
-- zh_CN.gbk
-- zh_HK.UTF-8
-- zh_HK.big5
-- zh_HK.big5hkscs
-- zh_TW
-- zh_TW.UTF-8
  |-- zh_TW.big5
-- proxymngr -> /etc/X11/proxymngr
-- rstart -> /etc/X11/rstart
-- twm -> ../../../../etc/X11/twm
-- x11perfcomp
-- xdm -> ../../../../etc/X11/xdm
-- xinit
-> ../../../../etc/X11/xinit
  |-- xkb -> /etc/X11/xkb
  |-- xserver -> /etc/X11/xserver
  |-- xsm -> ../../../../etc/X11/xsm
  |-- zh_CN
  |   |-- app-defaults
  -- zh_CN.GB18030 -> zh_CN
  -- zh_CN.GB2312 -> zh_CN
  -- zh_CN.GBK -> zh_CN
  -- zh_TW
  |   |-- app-defaults
  |-- zh_TW.Big5 -> zh_TW
  |-- modules
  |   |-- dri
  |   |-- drivers
  |   |-- extensions
  |   |-- fonts
  |   |-- input
  |   |-- internal
  |   |-- linux
  |-- multimedia
-- man
  |-- man1
  |-- man4
  |-- man5
  |-- man7
  |-- share
-- bin
  |-- X11 -> ../X11R6/bin
-- etc
-- games
-- include
-- java

```

```

-- kerberos
  |-- share
  |-- gnats
-- lib
  |-- GConf
  |   |-- 2
  -- ICAClient
  |   |-- keystore
  |   |-- cacerts
-- X11 -> ../X11R6/lib/X11
-- bonobo
  |-- monikers
  |-- servers
-- engines
-- expect5.43
-- firefox
  |-- chrome
  |   |-- icons
  |   |-- default
  |   |-- overlayinfo
  -- components
  |-- defaults
  |   |-- autoconfig
  |   |-- pref
  |   |-- profile
  |   |-- chrome
  |   |-- profile.ja
  |   |-- chrome
  -- extensions
  |   |-- inspector@mozilla.org
  |   |-- chrome
  |   |-- components
  |   |-- defaults
  |   |-- preferences
  |-- {972ce4c6-7e08-4474-
a285-3208198ce6fd}
  |-- gprefs
  |-- icons
  |-- init.d
  |-- plugins
  |-- res
  |   |-- dtd
  |   |-- entityTables
  |   |-- fonts
  |   |-- html
  |-- searchplugins
-- games
-- gcc
  |-- i386-pc-linux
  |-- 3.4.3
-- gcc-lib
-- gconv
-- gettext
-- gnome-vfs-2.0
  |-- modules
-- grub
-- gtk-2.0
  |-- 2.4.0
  |   |-- engines
  |   |-- filesystems
  |   |-- immodules
  |   |-- loaders
  |-- immodules
  |-- modules
-- hotplug
  |-- firmware
-- im
  |-- csconv
  |-- compoundtext
  |-- icuconv

```

```

-- leif
-- lib
-- share
  -- icu
  |  -- 1.4.0
  -- iiim
  -- locale
  |  -- ja
  |  |  -- LC_MESSAGES
  -- ko
  |  |  -- LC_MESSAGES
  -- zh_CN
  |  |  -- LC_MESSAGES
  -- zh_HK
  |  |  -- LC_MESSAGES
  -- zh_TW
  |  |  -- LC_MESSAGES
-- kbd
  -- consolefonts
  -- consoletrans
  -- keymaps
  |  -- amiga
  |  -- atari
  |  -- i386
  |  |  -- azerty
  |  |  -- dvorak
  |  |  -- fgJlod
  |  |  -- include
  |  |  -- qwerty
  |  |  -- qwertz
  |  -- include
  |  -- mac
  |  -- sun
  -- keytables -> keymaps/i386/qwerty
  -- unidata
  -- videomodes
-- libglade
  -- 2.0
-- libuser
-- locale -> ../share/locale
-- mozilla
  -- plugins
-- murasaki
-- netscape
  -- plugins
-- nptl
-- orbit-2.0
-- pango
  -- 1.4.0
  -- modules
-- perl5
  -- 5.8.5
  |  -- Attribute
  |  |  -- Handlers
  -- B
  -- CGI
  |  -- eg
  -- CPAN
  -- Carp
  -- Class
  -- DBM_Filter
  -- Devel
  -- Digest
  -- Encode
  |  -- CN
  |  -- JP
  |  -- KR
  |  -- MIME
  |  -- Unicode
  -- Exporter

```

```

-- ExtUtils
  -- Command
  -- Liblist
  -- MakeMaker
-- File
  -- Spec
-- Filter
  -- Simple
-- Getopt
  -- Long
-- Hash
-- I18N
  -- LangTags
-- IO
  -- Socket
-- IPC
-- List
-- Locale
  -- Codes
  -- Maketext
-- Math
  -- BigFloat
  -- BigInt
-- Memoize
-- NEXT
-- Net
  -- FTP
  -- Ping
  -- demos
-- PerlIO
  -- via
-- Pod
  -- Perldoc
  -- Text
-- Scalar
-- Search
-- Switch
-- Term
  -- ANSIColor
-- Test
  -- Harness
  -- Simple
-- Text
  -- Balanced
  -- TabsWrap
-- Thread
-- Tie
-- Time
-- Unicode
  -- Collate
-- User
-- auto
  -- POSIX
  -- SigAction
-- i386-linux-thread-multi
  -- B
  -- CORE
  -- Data
  -- Devel
  -- Digest
  -- Encode
  -- File
  -- Filter
  |  -- Util
  -- I18N
  -- IO
  -- IPC
  -- MIME
  -- PerlIO
  -- Sys

```

```

-- Time
-- Unicode
-- XS
-- asm
-- auto
  |-- B
  |  |-- C
  |-- ByteLoader
  |-- Cwd
  |-- Data
  |  |-- Dumper
  -- Devel
  |  |-- DProf
  |  |-- PPPort
  |  |-- Peek
  -- Digest
  |  |-- MD5
  -- DynaLoader
  -- Encode
  |  |-- Byte
  |  |-- CN
  |  |-- EBCDIC
  |  |-- JP
  |  |-- KR
  |  |-- Symbol
  |  |-- TW
  |  |-- Unicode
  -- Fcntl
  -- File
  |  |-- Glob
  -- Filter
  |  |-- Util
  |     |-- Call
  -- GDBM_File
  -- I18N
  |  |-- Langinfo
  -- IO
  -- IPC
  |  |-- SysV
  -- List
  |  |-- Util
  -- MIME
  |  |-- Base64
  -- NDBM_File
  -- Opcode
  -- POSIX
  -- PerlIO
  |  |-- encoding
  |  |-- scalar
  |  |-- via
  -- SDBM_File
  -- Socket
  -- Storable
  -- Sys
  |  |-- Hostname
  |  |-- Syslog
  -- Time
  |  |-- HiRes
  -- Unicode
  |  |-- Normalize
  -- XS
  |  |-- APITest
  |  |-- Typemap
  -- attrs
  -- re
  -- sdbm
  |-- threads
  |  |-- shared
-- libffi
-- linux

```

```

|-- scsi
  |-- threads
  -- pod
  -- unicore
  |  |-- To
  |  |-- lib
  |     |-- bc
  |     |-- ccc
  |     |-- dt
  |     |-- ea
  |     |-- gc_sc
  |     |-- hst
  |     |-- jt
  |     |-- lb
  |     |-- nt
  |-- warnings
  -- man
  |  |-- man3
  |-- site_perl
  |  |-- 5.8.5
  |     |-- i386-linux-thread-multi
  -- pkgconfig
  -- pppd
  |  |-- 2.4.3
  -- python2.4
  |  |-- lib-dynload
  |  |  |-- chineseccn
  |  |  |-- chinesetw
  |-- site-packages
  -- rpm
  |  |-- athlon-linux
  |  |-- i386-linux
  |  |-- i486-linux
  |  |-- i586-linux
  |  |-- i686-linux
  |  |-- noarch-linux
  |  |-- pentium3-linux
  |  |-- pentium4-linux
  -- sasl2
  -- scim-1.0
  |  |-- 1.4.0
  |  |  |-- Config
  |  |  |-- Filter
  |  |  |-- FrontEnd
  |  |  |-- Helper
  |  |  |-- IMEngine
  |  |  |-- SetupUI
  |  |-- Config
  |  |-- Filter
  |  |-- FrontEnd
  |  |-- Helper
  |  |-- IMEngine
  |  |-- SetupUI
  -- terminfo -> ../share/terminfo
  -- thunderbird
  |  |-- chrome
  |  |  |-- icons
  |  |  |  |-- default
  |  |  |-- overlayinfo
  |-- components
  |  |-- myspell
  -- defaults
  |  |-- autoconfig
  |  |-- isp
  |  |  |-- US
  |-- messenger
  |  |-- US
  -- pref
  -- profile
  |  |-- US

```

```

|-- wallet
|-- extensions
  |-- {972ce4c6-7e08-4474-
a285-3208198ce6fd}
  |-- greprefs
  |-- icons
  |-- init.d
  |-- res
    |-- dtd
    |-- entityTables
    |-- fonts
    |-- html
|-- tls -> nptl
|-- xfce4
  |-- mcs-plugins
  |-- modules
  |-- panel-plugins
  |-- splash
  |-- engines
-- libexec
  |-- awk
  |-- awk
  |-- gcc
  |-- i386-pc-linux
  |-- 3.4.3
  |-- getconf
  |-- openssh
-- local
  |-- bin
  |-- etc
  |-- games
  |-- include
  |-- lib
  |-- netscape
  |-- plugins
  |-- libexec
  |-- man
  |-- man1
  |-- sbin
  |-- share
  |-- info
  |-- man
    |-- man1
    |-- man2
    |-- man3
    |-- man4
    |-- man5
    |-- man6
    |-- man7
    |-- man8
    |-- man9
    |-- mann
  |-- src
-- sbin
  |-- busybox-symlinks
-- share
  |-- aclocal
  |-- alsa
  |-- cards
  |-- S17018
  |-- pcm
  |-- anthy
  |-- application-registry
  |-- applications
  |-- apps
  |-- kicker
  |-- pics
  |-- ksplash
  |-- Themes
  |-- Turbolinux

```

```

|-- locolor
  |-- pics
  |-- switchdesk
-- awk
  |-- awk
-- changefont
  |-- data
  |-- fontconfig_ja
  |-- modules
-- control-center-2.0
  |-- capplets
-- cups
  |-- model
-- curl
-- dict
-- doc
  |-- HTML
  |-- turbolinux
  |-- en_US
  |-- images
  |-- ja_JP
  |-- libxslt-1.1.15
  |-- html
  |-- packages
  |-- audiofile
  |-- docs
  |-- changefont
  |-- curl
  |-- distribution-X
  |-- distribution-release
  |-- esound
  |-- firefox
  |-- fontconfig
  |-- gnome-keyring
  |-- gnome-mime-data
  |-- gnome-vfs2
  |-- gzip
  |-- ipa-fonts-support
  |-- ipsec-tools
  |-- ipw2200-firmware
  |-- ipw3945-ucode
  |-- kernel-extramodules
  |-- libbonoboui
  |-- libgnome
  |-- libgnomeui
  |-- openssh
  |-- openssl
  |-- doc
    |-- HOWTO
    |-- apps
    |-- crypto
    |-- ssl
  |-- popt
  |-- preload
  |-- rp-pppoe
  |-- rpm
  |-- tar
  |-- thunderbird
  |-- unionfs
  |-- wstreams
  |-- wstreams-unicnf
-- emacs
  |-- site-lisp
-- empty
-- file
-- fonts
  |-- ja
  |-- TrueType
-- games
-- gnome-ppp

```

```

-- glade
-- gpg-pubkey
-- groff
  -- 1.18.1.1
    -- font
      -- devX100
      -- devX100-12
      -- devX75
      -- devX75-12
      -- devascii
      -- devascii8
      -- devdvi
      -- generate
      -- devhtml
      -- devkorean -> devnippon
      -- devlatinl
      -- devlbp
      -- devlj4
      -- generate
      -- devnippon
      -- devps
      -- generate
      -- devutf8
    -- tmac
      -- mdoc
      -- mm
  -- site-tmac
-- il8n
  -- charmaps
  -- locales
-- icons
  -- HighContrast
    -- 48x48
    -- apps
  -- HighContrastInverse
    -- 48x48
    -- apps
  -- LowContrast
    -- 48x48
    -- apps
  -- hicolor
    -- 128x128
    -- apps
    -- 16x16
    -- apps
    -- 192x192
    -- apps
    -- mimetypes
    -- 32x32
    -- apps
    -- 48x48
    -- apps
    -- mimetypes
    -- 64x64
    -- apps
    -- 72x72
    -- apps
    -- scalable
    -- apps
-- idl
  -- at-spi-1.0
  -- bonobo-2.0
  -- bonobo-activation-2.0
  -- orbit-2.0
-- info
-- locale
  -- af
    -- LC_MESSAGES
  -- am
    -- LC_MESSAGES

```

```

-- ang
  -- LC_MESSAGES
-- ar
  -- LC_MESSAGES
-- as
  -- LC_MESSAGES
-- az
  -- LC_MESSAGES
-- az_IR
  -- LC_MESSAGES
-- be
  -- LC_MESSAGES
-- bg
  -- LC_MESSAGES
-- bn
  -- LC_MESSAGES
-- bn_IN
  -- LC_MESSAGES
-- br
  -- LC_MESSAGES
-- bs
  -- LC_MESSAGES
-- ca
  -- LC_MESSAGES
-- cs
  -- LC_MESSAGES
-- cy
  -- LC_MESSAGES
-- da
  -- LC_MESSAGES
-- de
  -- LC_MESSAGES
-- de_DE
  -- LC_MESSAGES
-- el
  -- LC_MESSAGES
-- en
  -- en_CA
  -- LC_MESSAGES
  -- en_GB
  -- LC_MESSAGES
-- eo
  -- LC_MESSAGES
-- es
  -- LC_MESSAGES
-- es_MX
  -- LC_MESSAGES
-- et
  -- LC_MESSAGES
-- eu
  -- LC_MESSAGES
-- eu_ES
  -- LC_MESSAGES
-- fa
  -- LC_MESSAGES
-- fi
  -- LC_MESSAGES
-- fr
  -- LC_MESSAGES
-- fr_FR
  -- LC_MESSAGES
-- ga
  -- LC_MESSAGES
-- gl
  -- LC_MESSAGES
-- gu
  -- LC_MESSAGES
-- he
  -- LC_MESSAGES
-- hi

```

```
`-- LC_MESSAGES
-- hr
  `-- LC_MESSAGES
-- hu
  `-- LC_MESSAGES
-- hy
  `-- LC_MESSAGES
-- ia
  `-- LC_MESSAGES
-- id
  `-- LC_MESSAGES
-- is
  `-- LC_MESSAGES
-- it
  `-- LC_MESSAGES
-- ja
  `-- LC_MESSAGES
-- ja_JP
  `-- LC_MESSAGES
-- ja_JP.eucJP
  `-- LC_MESSAGES
-- ja_JP.eucjisx0213
  `-- LC_MESSAGES
-- ja_JP.ujis
  `-- LC_MESSAGES
-- ja_JP.utf8
  `-- LC_MESSAGES
-- ka
  `-- LC_MESSAGES
-- kn
  `-- LC_MESSAGES
-- ko
  `-- LC_MESSAGES
-- ku
  `-- LC_MESSAGES
-- ky
  `-- LC_MESSAGES
-- li
  `-- LC_MESSAGES
-- lo
  `-- LC_MESSAGES
-- lt
  `-- LC_MESSAGES
-- lv
  `-- LC_MESSAGES
-- mi
  `-- LC_MESSAGES
-- mk
  `-- LC_MESSAGES
-- ml
  `-- LC_MESSAGES
-- mn
  `-- LC_MESSAGES
-- mr
  `-- LC_MESSAGES
-- ms
  `-- LC_MESSAGES
-- nb
  `-- LC_MESSAGES
-- nb_NO
  `-- LC_MESSAGES
-- ne
  `-- LC_MESSAGES
-- nl
  `-- LC_MESSAGES
-- nn
  `-- LC_MESSAGES
-- no
  `-- LC_MESSAGES
-- nso
```

```
`-- LC_MESSAGES
-- or
  `-- LC_MESSAGES
-- pa
  `-- LC_MESSAGES
-- pl
  `-- LC_MESSAGES
-- pt
  `-- LC_MESSAGES
-- pt_BR
  `-- LC_MESSAGES
-- pt_PT
  `-- LC_MESSAGES
-- ro
  `-- LC_MESSAGES
-- ru
  `-- LC_MESSAGES
-- rw
  `-- LC_MESSAGES
-- si
  `-- LC_MESSAGES
-- sk
  `-- LC_MESSAGES
-- sl
  `-- LC_MESSAGES
-- sp
  `-- LC_MESSAGES
-- sq
  `-- LC_MESSAGES
-- sr
  `-- LC_MESSAGES
-- sr@Latn
  `-- LC_MESSAGES
-- sr@ije
  `-- LC_MESSAGES
-- sv
  `-- LC_MESSAGES
-- ta
  `-- LC_MESSAGES
-- th
  `-- LC_MESSAGES
-- tk
  `-- LC_MESSAGES
-- tr
  `-- LC_MESSAGES
-- uk
  `-- LC_MESSAGES
-- ur
  `-- LC_MESSAGES
-- uz
  `-- LC_MESSAGES
-- uz@Latn
  `-- LC_MESSAGES
-- vi
  `-- LC_MESSAGES
-- wa
  `-- LC_MESSAGES
-- xh
  `-- LC_MESSAGES
-- yi
  `-- LC_MESSAGES
-- zh
  `-- LC_MESSAGES
-- zh_CN
  `-- LC_MESSAGES
-- zh_CN.GB2312
  `-- LC_MESSAGES
-- zh_HK
  `-- LC_MESSAGES
-- zh_TW
```

```

    |-- LC_MESSAGES
    |-- zh_TW.Big5
    |-- LC_MESSAGES
    |-- zu
    |-- LC_MESSAGES
-- man
  |-- fr
  |-- man8
  |-- ja
  |-- man8
  |-- ko
  |-- man8
  |-- man1
  |-- man2
  |-- man3
  |-- man4
  |-- man5
  |-- man6
  |-- man7
  |-- man8
  |-- man9
  |-- mann
  |-- pl
  |-- man1
  |-- man8
  |-- pt_BR
  |-- man8
  |-- ru
  |-- man8
  |-- sk
  |-- man8
-- mime-info
-- mimesn
  |-- audio
-- misc
  |-- getopt
-- pixmaps
  |-- distro
-- printconf
  |-- mf_rules
  |-- tests
-- rdesktop
  |-- keymaps
-- scim
  |-- Anthy
  |-- style
  |-- icons
-- sgml
  |-- gconf
-- tabset
-- tcl8.4
  |-- encoding
  |-- http1.0
  |-- http2.5
  |-- msgcat1.3
  |-- opt0.4
  |-- tcltest2.2
-- terminfo
  |-- 1
  |-- 2
  |-- 3
  |-- 4
  |-- 5
  |-- 6
  |-- 7
  |-- 8
  |-- 9
  |-- A
  |-- E
  |-- L

```

```

-- M
-- N
-- P
-- Q
-- X
-- a
-- b
-- c
-- d
-- e
-- f
-- g
-- h
-- i
-- j
-- k
-- l
-- m
-- n
-- o
-- p
-- q
-- r
-- s
-- t
-- u
-- v
-- w
-- x
-- z
-- themes
  |-- Default
  |-- balou
  |-- gtk
  |-- gtk-2.0
  |-- gtk-2.0-key
  |-- xfw4
-- Default-4.0
  |-- xfw4
-- Emacs
  |-- gtk-2.0-key
-- Opta
  |-- xfw4
-- Symphony
  |-- xfw4
-- Therapy
  |-- xfw4
-- Turbolinux
  |-- gtk
  |-- gtk-2.0
-- Tyrex
  |-- xfw4
-- Waza
  |-- xfw4
  |-- Wildbush
  |-- xfw4
-- tk8.4
  |-- demos
  |-- images
  |-- images
  |-- msgs
-- tkppoe
-- tmac -> groff/1.18.1.1/tmac
-- xfce4
  |-- backdrops
  |-- doc
  |-- C
  |-- images
  |-- fr
  |-- images

```

```

|-- mime
|-- xfw4
|-- xml
|-- libglade
|-- xsessions
|-- zoneinfo
|-- Africa
|-- America
|   |-- Argentina
|   |-- Indiana
|   |-- Kentucky
|   |-- North_Dakota
|-- Antarctica
|-- Arctic
|-- Asia
|-- Atlantic
|-- Australia
|-- Brazil
|-- Canada
|-- Chile
|-- Etc
|-- Europe
|-- Indian
|-- Mexico
|-- Mideast
|-- Pacific
|-- SystemV
|-- US
|-- posix
|   |-- Africa
|   |-- America
|   |   |-- Argentina
|   |   |-- Indiana
|   |   |-- Kentucky
|   |   |-- North_Dakota
|   |-- Antarctica
|   |-- Arctic
|   |-- Asia
|   |-- Atlantic
|   |-- Australia
|   |-- Brazil
|   |-- Canada
|   |-- Chile
|   |-- Etc
|   |-- Europe
|   |-- Indian
|   |-- Mexico
|   |-- Mideast
|   |-- Pacific
|   |-- SystemV
|   |-- US
|-- right
|   |-- Africa
|   |-- America
|   |   |-- Argentina
|   |   |-- Indiana
|   |   |-- Kentucky
|   |   |-- North_Dakota
|   |-- Antarctica
|   |-- Arctic
|   |-- Asia
|   |-- Atlantic
|   |-- Australia
|-- Brazil
|-- Canada
|-- Chile
|-- Etc
|-- Europe
|-- Indian
|-- Mexico
|-- Mideast
|-- Pacific
|-- SystemV
|-- US
|-- src
|-- tmp -> ../var/tmp
|-- var
|   |-- account
|   |-- cache
|   |   |-- fonts
|   |   |-- man
|   |-- db
|   |-- devices
|   |-- empty
|   |-- games
|   |-- lib
|   |   |-- games
|   |   |-- misc
|   |   |-- pcmcia
|   |   |-- preload
|   |   |-- rpm
|   |   |-- alternatives
|   |   |-- runonce
|   |   |-- symbolic-dtool
|   |   |-- turboplus
|   |   |   |-- comps
|   |   |   |-- installed
|   |   |   |-- 0
|   |   |-- uniconf
|   |   |-- xdm
|   |   |-- xinit
|   |   |-- xkb
|   |-- local
|   |-- lock
|   |   |-- console
|   |   |-- rpm
|   |   |-- subsys
|   |-- log
|   |-- mail -> spool/mail
|   |-- nis
|   |-- opt
|   |-- preload
|   |-- preserve
|   |-- run
|   |   |-- netreport
|   |-- spool
|   |   |-- lpd
|   |   |-- mail
|   |   |-- repackage
|   |-- src
|   |-- ssl
|   |   |-- certs
|   |   |-- misc
|   |   |-- private
|   |-- tmp
|   |-- yp

```

上記のルートツリーに RPM ファイルを追加する際には、次に示す手順を行うことで追加することが

```

$ su-
Password
#
# cd /work/
#
# rpm -ivh --root=root-tree/ 'パッケージ名'
#

```

## A.2. ISO イメージファイルの作成

ISO イメージを作成する場合に利用するディレクトリ構造を次に示す。

```

iso
|-- appmod
|
|-- images
|
`-- isolinux

```

ISO ファイルを作成するには、squashfs を利用するため、作成する PC に以下のパッケージのインストールが必要である。

表 A-1:squashfs を利用するために必要なパッケージ

パッケージ名	説明
squashfs-tools-3.1-1	Squash Filesystem 作成用のツール

ISO ファイルを作るための手順を次に示す。

```

$ su -
Password
#
# cd /work
# mksquashfs /work/root-tree root.sqsh

---- squash fs の実行 ----

# cd iso
#
# mkisofs -V HelpPC -A Linux -R -T -J -hide-joliet instimage -hide-joliet-trans-tbl -

```

```
b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-  
info-table -o /tmp/test.iso ./
```

## 付録 B. サーバ側への問い合わせに関して

### B.1. ローカルでの収集情報の転送

ローカルの PC でハードウェア機器の情報収集を行った結果をサーバに問い合わせる際には、次のようにサーバ側に問い合わせることで、結果を得ることができる。

```
# /usr/sbin/lshw -xml > lshw.xml ← xml の生成
#
# /usr/bin/curl -F "xmlfile=@lshw.xml" http:// 'サーバ名' /tool/?p=receive
> /tmp/result.html
↑
上記のようにする事で、サーバ側への問い合わせが行われる。
行った結果は、リダイレクト (>) でhtml ファイルとして取得する事ができる。
```

## 付録 C. データベースへの自動投入に関して

ここでは、プロトタイプで利用するデータベースへ、自動的にデータ投入を行う場合に必要となる情報をまとめている。

### C.1. Linux のデバイス情報

各ディストリビュータで利用されている共通のデータとなる。この情報はインターネットにて配布されている情報であり、種類としては、PCI バスと USB バスに接続されるデバイスの情報のみとなる。

取得方法は、インターネットにて配布されているテキスト情報をダウンロードすることで取得することができる。取得できる URL を次に示す。

表 C-1:Linux で利用されるデバイス情報源

入手先 URL	説明	取得ファイル
The Linux PCI ID Repository <a href="http://pciids.sourceforge.net/">http://pciids.sourceforge.net/</a>	PCI バスに接続されるデバイスの情報	pci.ids
Linux USB <a href="http://www.linux-usb.org/">http://www.linux-usb.org/</a>	USB バスに接続されるデバイスの情報	usb.ids

### C.2. Linux Kernel モジュール情報

Linux Kernel モジュール情報は、大別すると3種類のデータとなる。1つ目は、モジュールをバス単位に分けて、全てのドライバ情報と関連を示している「マップ情報」と、2つ目は、モジュール(.ko)単位に分かれている「モジュール情報」、3つ目は、ブート時の利用する「boot kernel ドライバ情報」に分かれる。

これらの情報を取得するには、各ディストリビュータが配布している Linux Kernel パッケージに含まれおり、該当のパッケージを入手する必要がある。

### C.3. X Window システムモジュール情報の取得

X Window システム情報は、Video ドライバモジュールが持っている情報である。この情報を取得するには、各ディストリビュータが配布している X Window システムのパッケージに含まれており、該当のパッケージを入手する必要がある。

### C.4. 該当パッケージからの情報取得

該当パッケージを入手することができた場合には、次に示す手順を行うことで、情報の取得でき

る。

```
$ su-
Password
# cd /tmp
#
# mkdir distrol-work
#
# cd distrol-work/
#
※RPM系のパッケージの場合
# rpm2cpio “パッケージ名” | cpio -id
#
※deb系のパッケージの場合
# ar x “パッケージ名”
#
```

\* パッケージの入手方法は、ftpや配布 CD-ROM などの方法がある。

Linux Kernel の場合には、「マップ情報」の生成が必要になる。マップ情報生成のためには、次に示すパッケージを利用する必要がある。

表 C-2: マップ情報の生成に必要なパッケージ

パッケージ名	説明
modutils-2.4.25-22	Kernel モジュールのユーティリティ

```
$ su-
Password
# cd /tmp
#
# mkdir distrol-work
#
# cd distrol-work/
#
# /sbin/depmod -b /distrol-work -aq -F ./boot/System.map-`${kernel バージョン}` `${kernel
バージョン}`
```

## C.5. アプリケーションソフトウェアの情報

各ディストリビュータからは、Linux kernel と X Window システムのパッケージ以外に多くパッ

ページが配布されている。各パッケージには、さまざまな情報が含まれており、これらの情報を利用してパッケージの利用用途の情報とすることができる。この情報を取得するには、各ディストリビュータが配布しているパッケージを入手し、情報を取得する。今回の調査では、deb 系のパッケージと、rpm 系のパッケージを対象に調査を実施した。

### C.5.1. Debian 系 Linux ディストリビューションのパッケージ情報取得

Debian 系 Linux ディストリビューションパッケージからの情報取得は、dpkg コマンドと apt-cache コマンドを利用して、各パッケージの情報を取得する。情報取得する手順を次に示す。

```
# dpkg -l ← パッケージの一覧を表示する。
#
# apt-cache show ${パッケージ名} ← パッケージ情報
# dpkg -L ${パッケージ名} ← パッケージのインストール先リスト
# apt-cache showpkg ${パッケージ名} ← パッケージの依存関係リスト
```

### C.5.2. RPM 系 Linux ディストリビューションのパッケージ情報取得

RPM 系 Linux ディストリビューションパッケージからの情報取得は、rpm コマンドを利用し、各パッケージの情報を取得する。情報取得する手順を次に示す。

```
# rpm -qa ← パッケージの一覧を表示する。
#
# rpm -qil ${パッケージ名} ← パッケージ情報, インストール先リストの取得
# rpm -qR ${パッケージ名} ← パッケージの依存関係リスト
```

### C.5.3. データベースの投入データ生成 (Linux Kernel 編)

パッケージから取得した情報の中には、テキスト化されている情報とバイナリ化されている情報がある。データベースに投入する場合には、バイナリ化されている情報とテキスト化されている情報を CSV フォーマット(テキスト)に統一し、データベースに登録しやすい形に成形する。

データを成形するために、次に示すパッケージを利用する。パッケージを該当の PC にインストールし、各情報の CSV 生成をする。

表 C-3:投入データの成形に必要なパッケージ

パッケージ名	説明
deviceinfo-tools-0.0.0-1.i686.rpm	.ko となっていない Kernel 情報の CSV 化をする。
mi2csv-0.0.0-4.i686.rpm	.ko となっている Kernel 情報の CSV 化をする。

### (1)Linux でのデバイス情報の CSV 生成

前項に示した URL より該当する ids ファイルの入手をする。入手後、次に示すコマンドを利用することで、ids ファイルを CSV 化する事ができる。

```
$ su-
Password
# cd /tmp
# mkdir ids
# cd ids/
#
※PCI デバイスの場合
# pciids2csv ../pci.ids >pciids.csv
#
※USB デバイスの場合
# usbids2csv ../usb.ids >usbids.csv
```

### (2)Linux Kernel マップ情報の CSV 生成

前項に示したように、Linux Kernel パッケージに含まれている情報の1つである「マップ情報」を取得する。「マップ情報」取得後、次に示すコマンドを利用することで、マップ情報を CSV 化することができる。

```
$ su-
Password
# cd /tmp
# mkdir map
# cd distrol-work/
# cd lib/modules/ 'kernel バージョン' /
#
※PCI デバイスの場合
# pcimap2csv modules.pcimap >/tmp/map/pcimap.csv
#
※USB デバイスの場合
# usbmap2csv modules.usbmap >/tmp/map/usbmap.csv
```

### (3)Linux Kernel モジュール情報の CSV 生成

前項に示したように、Linux Kernel パッケージに含まれている情報の1つである「モジュール

ル情報」を取得する。「モジュール情報」は、テキスト化がなされていないため、テキスト化するために次の表に示すパッケージを利用することが必要となる。

表 C-4:モジュール情報の取得に必要なパッケージ

パッケージ名	説明
modutils-2.4.25-22	Kernel モジュールのユーティリティ

パッケージをインストール後、次に示すコマンドを利用することで、マップ情報から CSV を生成する。

```
$ su-
Password
# cd /tmp
# mkdir driver-info
# cd distrol-work/
#
# cd lib/modules/ 'kernel バージョン' /kernel
#
# cd drivers/scsi/
#
# mi2csv 'ドライバ名' >/tmp/driver-info/ 'ドライバ名' .csv
#
```

上記の手順では、複数ある Linux Kernel ドライバの CSV ファイルを作り出すために、かなりの作業となるため、サンプルのシェルスクリプトを次に示す。

#### サンプルシェルスクリプト

```
#!/bin/sh

kernel_dir=/lib/modules/`uname -r`/kernel          ← 該当するディレクトリ位置に変更
for arg in `find $kernel_dir -name "*.ko" -type f`; do
    if [ `echo $arg | sed -e 's/^.*/'/'` = 'ko' ]; then
        name=`basename $arg .ko`
        ./mi2csv `echo "$arg"` >$name.csv
    fi
done
```

#### (4) Linux Kernel の boot kernel ドライバ情報の CSV 生成

前項に示したように、Linux Kernel パッケージに含まれている情報の1つである「boot

kernelドライバ情報」を取得する。「boot kernelドライバ情報」は、バイナリ圧縮されているファイルからLinux Kernelのバイナリ情報取得を行い、取得したバイナリ情報からテキスト化することが必要になる。

バイナリ圧縮しているファイルからの情報取得は、手順が確立されておらず、困難である。このため、等価な他の手順にて取得する必要がある。今回の調査では、①,②に示す方法にて代用することとする。

- ① vmlinux ファイルを配布しているディストリビューションでは、vmlinux を利用してLinux kernel のバイナリ情報を取得する。
- ② vmlinux ファイルを配布していない場合には、コンパイルにてLinux Kernel のvmlinux を生成し、バイナリ情報を取得する。

コンパイル作業を行うには、各ディストリビューションがインストールされている環境とコンパイル作業が必要となる。コンパイル作業後に、Linux Kernel のバイナリ情報から、テキスト化するために次のに示すパッケージを利用し、CSV ファイルを生成する。

表 C-5:Linux Kernel のバイナリ情報をテキスト化するために必要なパッケージ

パッケージ名	説明
binutils-2.16.91.0.1-1	バイナリファイルのユーティリティツール

上記①,② のどちらの手順を利用するかは、各ディストリビューションの配布方法や、パッケージにする手法に依存するため、一意には決めることができない。本来は、各ディストリビューションに依存しない方法を選択することが望ましく、統一するためには、情報を入手するためのツールの開発を行うか、ルールの確立を行い、各ディストリビューションから情報を取得する際の手順を確立する必要があると考える。

本プロトタイプでは、統一する手順の確立よりも情報を入手することを優先し、調査対象となるディストリビューションに合った、手順の取捨選択を行い、選択した手順での情報収集を実施することとする。今回の対象ディストリビューションで利用した手順を次の表にまとめる。

表 C-6:ディストリビューション別入手方法

ディストリビューション名	利用した手順
Fedora 7	②
Open SUSE 10.2	①
Knoppix 5.1.1	②
Ubuntu 7.04	②
Turbolinux FUJI	①

調査での手順を簡易化するため、今回は各ディストリビューションの開発環境パッケージを含め、インストールした環境を利用し、実施する。

#### C.5.4. vmlinux ファイルから Linux Kernel のバイナリ情報を取得する場合

vmlinux は、インテルアーキテクチャにて Bootup に利用される vmlinuz を作り出す前の Linux Kernel のシンボル情報が入っているバイナリファイルである。

起動時に利用される vmlinuz に含まれるデバイスドライバ情報は、各ディストリビュータが固有に決めるため、各ディストリビューションの開発環境を調査することになってしまう。この開発環境の調査を行うには、ソースファイルや、Linux Kernel を構築するためのコンフィグファイルなどを見ながら、どのデバイスドライバが vmlinuz に含まれているかを調査するため、作業的には困難且つ時間を要することになる。今回の調査では、vmlinux 内に含まれるデバイスドライバのシンボル情報を検索することとする。この調査方法を選択した理由としては、調査作業を簡易化することができ且つ、ソースファイルのトレースを行わなくて済むので作業の簡易化ができることが挙げられる。

##### (1) ソースファイルの調査

ソースファイルを入手する。各ディストリビュータがネットワーク上で配布しているパッケージを利用する。もしくは、開発環境をインストールした環境であれば、配布しているパッケージはインストールされている。このソースファイルからデバイスドライバの情報を調査する方法を次に示す。

```
$ su-
Password
# cd /usr/src/linux
#
# grep -r pci_device_id * | grep "¥[¥]"
      ↑
      この名前を検索する
:
:
drivers/ide/ide.c:      static struct pci_device_id pci_default[] = {
drivers/ide/pci/siimage.c:static struct pci_device_id siimage_pci_tbl[] = {
drivers/ide/pci/aec62xx.c:static struct pci_device_id aec62xx_pci_tbl[] = {
drivers/ide/pci/it821x.c:static struct pci_device_id it821x_pci_tbl[] = {
drivers/ide/pci/it8172.c:static struct pci_device_id it8172_pci_tbl[] = {
drivers/ide/pci/slc90e66.c:static struct pci_device_id slc90e66_pci_tbl[] = {
:
:
      ↑
      この名前を利用する
```

```
#
```

この検索では、デバイスドライバが定義されているシンボル情報を検索し、このシンボル情報が `vmlinux` 内に含まれているかを検索する。この作業を簡易化するためのサンプルのシェルスクリプトを次に示す。

#### Linux Kernel 内のデバイスドライバ情報検索スクリプト

```
#!/bin/sh
grep -r pci_device_id * | grep "¥[¥]" >> check.list
```

#### 検索結果からシンボル情報を生成するスクリプト

```
#!/bin/sh
rm -f pciids.lst

while read i
do
chklist=`echo "$i" | cut -d '/' -f 1`
if [ "$chklist" = "Documentation" ]; then
echo "$chklist"
continue
fi
chksrc=`echo "$i" | cut -d ':' -f 1`
echo $chksrc
chkpci=`echo "$i" | cut -d ' ' -f 2`
if [ "$chkpci" != "pci_device_id" ]; then
    chkpci=`echo "$i" | cut -d ' ' -f 3`
    echo "Device check continue0!!"
echo $chkpci
fi
if [ "$chkpci" != "pci_device_id" ]; then
    chkpci=`echo "$i" | cut -d ' ' -f 5`
    echo "Device check continuel!!"
echo $chkpci
fi

tmp=`echo "$i" | cut -d ':' -f 2`
chktype=`echo "$tmp" | cut -d ' ' -f 1`
echo "Type:$chktype"

if [ "$chktype" = "extern" ]; then
    echo "extern detect!!"
```

```

        echo "!!!!!!!continue!!!!!!!"
continue
fi
if [ "$chkpci" = "pci_device_id" ]; then
    if [ "$chktype" = "struct" ]; then
        j=`echo "$i" | cut -d ' ' -f 3`
    else
        j=`echo "$i" | cut -d ' ' -f 4`
    fi
    if [ "$j" = "__devinitdata" ] || [ "$j" = "pci_device_id" ] || [ "$j" = "__
initdata" ] || [ "$j" = "__cpuinitdata" ] || [ "$j" = "const" ] || [ "$j" = "" ]
; then
        j=`echo "$i" | cut -d ' ' -f 5`
    fi
echo "check -f 5:$j"
    if [ "$j" = "const" ] || [ "$j" = "" ] || [ "$j" = "__devinitdata" ]; then
        j=`echo "$i" | cut -d ' ' -f 6`
    fi
echo "check -f 6:$j"
    if [ "$j" = "" ]; then
        j=`echo "$i" | cut -d ' ' -f 7`
    fi
echo "check -f 7:$j"
    tmpids=`echo "$j" | sed -e 's/¥[¥]//'\`
    pcichk=`echo "$tmpids" | sed -e 's/;$///'\`
    pciids=`echo "$tmpids" | sed -e 's/¥={//'\`
    if [ $pciids = $pcichk ]; then
        echo "$chksrc $pciids" >>pciids.lst
        echo "$pciids" >>sym.lst
    fi
fi
done <./check.list

```

これらのシェルスクリプトを利用することで、Linux Kernel のソースファイルからデバイスドライバ情報を取得する事ができる。

## (2)vmlinux の調査

vmlinux 内に含まれているシンボル情報と実際に定義されているデータ領域のダンプをする。このダンプしたシンボル情報から、(1)で生成したシンボルの検索し、シンボルがマッチした際に、ダンプ情報からデバイスドライバ情報が定義されている場所をダンプする。この作業

手順を次に示す。

```
$ su-
Password
# cd /boot
#
# objdump -j .data -t ${vmlinux 名} >check.sym
#
# grep “シンボル名” check.sym
#
c0341960 g      0 .data 00000054 k8_nb_ids
                        ↑
                        シンボル名がマッチした場合
#
# objdump -j .data -s --start-address=$startaddr --stop-address=$stopaddr ${vmlinux 名}
#
address=0xc03419b4 vmlinux-2.6.18.2-34-default

vmlinux-2.6.18.2-34-default:      file format elf32-i386

Contents of section .data:
c0341960 22100000 03110000 ffffffff ffffffff ”.....
c0341970 00000000 00000000 00000000 22100000 .....”...
c0341980 03120000 ffffffff ffffffff 00000000 .....

XXXXXXXX : の部分がデバイスドライバ情報になる。
```

上記の手順では、取得した情報をインテルアーキテクチャに則った成形(バイトスワップ)が必要になる。この作業を簡易化するためのサンプルスクリプトを次に示す。

#### vmlinux の内容をダンプするシェルスクリプト

```
#!/bin/sh

objdump -j .data -t $1 >check.sym
objdump -j .data -s $1 >check.data

※$1 には、vmlinux のファイル名を指定する。
```

#### vmlinux 内を検索して、デバイスドライバ情報をダンプするシェルスクリプト

```
#!/bin/sh
echo “#!/bin/sh”>check_csv.sh
```

```

echo "rm -rf csv/">>check_csv.sh
echo "mkdir -p csv/">>check_csv.sh
mkdir -p dt_lst

while read i
do
echo "check grep!! $i"
j=`grep $i ./check.sym`
echo "$j"

if [ "$j" != "" ]; then
sym=`echo "$j"|cut -d ' ' -f 9`
echo "Symbol : $sym"

    if [ "$i" = "$sym" ]; then
        addr=`echo "$j"|cut -d ' ' -f 1`
        k=`echo "$j"|cut -d ' ' -f 8`
        length=`echo "$k"|cut -d ' ' -f 2`
        echo $addr
        echo $length

        x=`echo $addr | tr ' [a-z]' '[A-Z]'`
tmpaddr=`dc -e "l6i $x p"`
x=`echo $length | tr ' [a-z]' '[A-Z]'`
tmplen=`dc -e "l6i $x p"`
echo "tmpaddr:$tmpaddr"
echo "tmplen:$tmplen"
tmpstop=`expr "$tmpaddr" + "$tmplen"`

hexaddr=`dc -e "l6o $tmpstop p"`
stopaddr=`echo "0x$hexaddr"`
startaddr=`echo "0x$addr"`
echo "$startaddr"
echo "$stopaddr"

        objdump -j .data -s --start-address=$startaddr --stop-address=$stopadd
r ./ $1 >dt_lst/$sym.dt

        echo "/work/kernel/kernel.sh/create_csv.sh dt_lst/$sym.dt">>check_csv.
sh

    else

```

```

        echo "!!!! continue symbol !!!!!"
    fi
fi

done <./sym.lst

chmod 755 check_csv.sh

```

スクリプトを起動すると、“check\_csv.sh”というシェルスクリプトが自動生成される。この自動生成されたスクリプトを起動するには、シェルスクリプトを同一ディレクトリに置く必要がある。

#### vmlinux のダンプ内容を CSV フォーマットに成形するシェルスクリプト

```

#!/bin/sh
function byteswap () {
local dt1=`echo "$1" | cut -c 1,2`
local dt2=`echo "$1" | cut -c 3,4`
local dt3=`echo "$1" | cut -c 5,6`
local dt4=`echo "$1" | cut -c 7,8`
swapresult=`echo "$dt4$dt3$dt2$dt1"`
}

n=1
declare -a ids_dt

while read i
do
addr=`echo "$i" | cut -d ' ' -f 1`
data1=`echo "$i" | cut -d ' ' -f 2`
data2=`echo "$i" | cut -d ' ' -f 3`
data3=`echo "$i" | cut -d ' ' -f 4`
data4=`echo "$i" | cut -d ' ' -f 5`

if [ "$addr" = "" ] || [ "$data1" = "" ] || [ "$data2" = "" ] || [ "$data3" = ""
] ||
[ "$data4" = "" ]; then
echo "Error read line!!"
echo $ascii
continue
fi

```

```

byteswap $data1
ids_dt[n]=$swapresult
n=`expr $n + 1`
byteswap $data2
ids_dt[n]=$swapresult
n=`expr $n + 1`
byteswap $data3
ids_dt[n]=$swapresult
n=`expr $n + 1`
byteswap $data4
ids_dt[n]=$swapresult
n=`expr $n + 1`

done <$1
name=`basename $1 .dt`
m=7

echo "vendor id,device id,sub vendor id,sub device id,class id,class mask,index"
>csv/$name.csv

for i in `seq 1 $n`
do
if [ $i = $m ]; then
echo "test dump!!"

if [ "${ids_dt[${m-6}]}" = "" ] || [ "${ids_dt[${m-5}]}" = "" ] || [ "${ids_dt[${m-4}]}" = "" ] || [ "${ids_dt[${m-3}]}" = "" ] || [ "${ids_dt[${m-2}]}" = "" ] || [ "${ids_dt[${m-1}]}" = "" ] || [ "${ids_dt[$m]}" = "" ]; then
continue
echo "null string detect!!"
elif [ "${ids_dt[${m-6}]}" = "00000000" ] && [ "${ids_dt[${m-5}]}" = "00000000" ] && [ "${ids_dt[${m-4}]}" = "00000000" ] && [ "${ids_dt[${m-3}]}" = "00000000" ] && [ "${ids_dt[${m-2}]}" = "00000000" ] && [ "${ids_dt[${m-1}]}" = "00000000" ] && [ "${ids_dt[$m]}" = "00000000" ] ; then
continue
echo "all 0 line detect!!"
fi

echo "${ids_dt[${m-6}]},${ids_dt[${m-5}]},${ids_dt[${m-4}]},${ids_dt[${m-3}]},${ids_dt[${m-2}]},${ids_dt[${m-1}]},${ids_dt[$m]}" >> csv/$name.Csv

```

```
m=`expr $m + 7`  
  
fi  
done  
exit 0
```

### C.5.5. コンパイルにてvmlinuxを生成し、Linux Kernelのバイナリ情報を取得する場合

vmlinuxが配布されていない場合は、前項に示した方法を行うことができない。このため、ソースファイルからvmlinuxを生成して、このvmlinuxから前項で示した方法にてデバイスドライバ情報を取得することができる。

#### (1) RPM系のパッケージをコンパイルする場合

\*.src.rpmとなっているファイルを入手する。各ディストリビューションにて配布されている方法(ftpなど)にて入手することができる。

入手後、各ディストリビューションがインストールされている環境にてKernelを作成する。この時に、パッケージングまでを行ってしまうと、vmlinuxが入手できないので、パッケージング前までの段階にて終了させ、vmlinuxを入手する。

#### RPM系のパッケージを利用する場合

```
# rpm -ivh kernel-2.6.22.4-65.fc7.src.rpm  
#  
# cd /usr/src/redhat/SPEC ← ※各ディストリビューションに合わせる。  
#  
# rpmbuild -bi kernel-2.6.spec  
:  
: ビルドが開始される。  
:  
# cd ../BUILDE  
# cd kernel-2.6.22/linux-2.6.22.i686  
#  
# ls vmlinux  
vmlinux
```

#### Debian系のパッケージを利用する場合

```
# apt-get source linux-generic  
# dpkg-source -x linux-source-2.6.20_2.6.20-16.31.dsc  
#  
# cd linux-source-2.6.20-2.6.20/
```

```
#
# dpkg-buildpackage -rfakeroot -us -uc
:
: ビルドが開始される。
:
# cd debian/build/build-386
#
# ls vmlinux
vmlinux
```

ここまでの作業を行った後、前項に示す手順を行うことで、デバイスドライバ情報を取得する事ができる。

## C.6. データベースの投入データ生成(X Window システム編)

パッケージから取得できる情報は、バイナリ化されている情報となる。データベースに投入する場合には、バイナリ化されている情報を CSV フォーマット(テキスト)に統一し、データベースに登録しやすい形に成形する。CSV を生成するために必要なツール類のパッケージは、Linux Kernel の場合と同様である。

Xorg, XFree86 などの Open Source 系の描画を行うソフトウェアが、各種ハードウェアの対応情報を収集する。Xorg などのソフトウェアは、Video カード以外のハードウェアの情報も持っているが、今回の調査では Video カードだけを対象とする。各種の Video カードのハードウェア対応情報を抜き出すために、適用できる手順は、以下の①～③の手順にてできると想定される。

- ① Binary ファイル(xxx\_drv.so)から情報を入手する。
- ②各ディストリビューションが配布しているテキスト情報から入手する。
- ③各ディストリビューションが配布しているソースコードから情報を入手する。

上記①～③のどの手順を利用するかは、各ディストリビューションの配布方法や、配布するパッケージ形式に依存するため、一意には決めることができない。本来は、各ディストリビューションに依存しない方法を選択することが望ましいが、統一するためには、情報を入手するためのツール開発や、提供形式などの確立をする必要があると考える。今回の調査では、統一する手法の確立よりも情報を入手することとし、調査対象となるディストリビューションにより、手順の取捨選択を行い、選択した手順での情報収集を実施することとする。

### C.6.1. Binary ファイル(xxx\_drv.so)から情報を入手する場合

情報収集に際しての制約事項として、Xorg, XFree86 などでは、各種の Video カードに対応す

る情報をソースコード中に埋め込んでいる。これら情報が何であるかの調査が最初に必要となる。

このため、Binary からの情報収集とはいえ、ソースコードの調査が必要となってしまう。この手順を用いる場合には、ある程度ソースコードに精通している必要があるので、標準的な手順としては利用することが難しいと思われる。この課題をクリアするためには、Binary ファイルと連携するツール類を開発し、自動的に情報を収集するようにすることが望ましいと考える。今回の調査では、ツール開発に主眼を置くことはせずに、情報収集に主眼を置いている。このため、ツール開発は今後の課題とし、ソースコードのトレースポイントを手順として提示し、情報収集を行うこととする。

## (1) ソースコードの入手

Xorg, XFree86 などを配布している本家サイトより、最新版のドライバソースを入手する。本家サイトより入手する理由としては、各ディストリビューションにてのソースへのカスタマイズ状況を把握することもできるためである。

Xorg : ftp://ftp.x.org/pub/individual/driver/

XFree86 : ftp://ftp.xfree86.org/pub/XFree86/

※入手方法としては、ftp などを利用する。

入手対象となるファイルとしては、ファイル拡張子が「xxx.tar.bz2」となっているファイルを対象とする。

## (2) ソースコードの解析

対象となるファイルを入手したら、ファイルの解凍を行いソースコードの閲覧できるようにする。ファイルが多数になるため、サンプルとしてファイルを一括して解凍できるシェルスクリプトを次に示す。

```
#!/bin/sh
mkdir -p driver

for i in `ls *.tar.bz2`
do
    tar jxvf $i
    mvfile=`basename $i .tar.bz2`
    mv $mvfile driver
done
```

解凍終了後、ソースコードの入手後、各ソースコードをトレースする必要がある。この作業は、

各ソースコードがどのような Video カードに対応しているかを判別するために必要な作業となる。このためには、各ソースコードの構造を理解することが必要ではあるが、今回は、技術的に難易度を求められる点についての課題をクリアするためにあるキーワードを検索する事で回避をする。

ソースを検索するためのキーワードとしては、「SymTabRec」、「PciChipsets」と定義しているデータ型を検索する事が考えられる。このデータ型は、各ソースコード中に定義がされており、各 Video カードの検出時に利用されるためのデータ型になっている。

```
/* Supported chipsets */
static SymTabRec VoodooChipsets[] = {
    { PCI_CHIP_VOODOO01, "Voodoo 1" },
    { PCI_CHIP_VOODOO02, "Voodoo 2" },
    {-1, NULL }
};

:

static PciChipsets VoodooPCIChipsets[] = {
    { PCI_CHIP_VOODOO01, PCI_CHIP_VOODOO01, 0 },
    { PCI_CHIP_VOODOO02, PCI_CHIP_VOODOO02, 0 },
    { -1,          -1, RES_UNDEFINED }
};

*. XXXXX : どちらかをキーワードにする
```

今回の調査を行った際に、どちらのキーワードを検索キーワードとするかに関しても調査を実施した。調査した結果、ソースコード中に「PciChipsets」を定義していない場合もあり、今回の調査では「SymTabRec」を検索キーワードとして利用することとする。

### (3) キーワードの検索

キーワードの検索に関しては、「grep」コマンドを利用する。grep コマンドを利用するためには、次のパッケージが必要となる。

表 C-7:キーワード検索に必要なパッケージ

パッケージ名	説明
grep-2.5.1-6	grep コマンドを利用するためのパッケージ

grep コマンドを利用した検索に関して、コマンドの利用方法に関しての説明は避けるが、今回の調査で必要となる検索を行う場合、grep コマンドの利用に関して次に示す。

```

$ su-
# cd /work/xorg/diver
# ls -l

xf86-video-amd-2.7.7.0/      xf86-video-apm-1.1.1/
xf86-video-ark-0.6.0/      xf86-video-ast-0.81.0/
xf86-video-ati-6.7.192/    xf86-video-chips-1.1.1/
xf86-video-cirrus-1.1.0/   xf86-video-cyrix-1.1.0/
                               :
xf86-video-nv-2.1.3/       xf86-video-rendition-4.1.3/
xf86-video-s3-0.5.0/       xf86-video-s3virge-1.9.1/
xf86-video-savage-2.1.3/   xf86-video-siliconmotion-1.5.1/
xf86-video-sis-0.9.3/      xf86-video-sisusb-0.8.1/
xf86-video-sunbw2-1.0.0.5/  xf86-video-sunbw2-1.1.0/
xf86-video-suncgl4-1.1.0/  xf86-video-suncg3-1.1.0/
xf86-video-suncg6-1.1.0/   xf86-video-sunffb-1.1.0/
xf86-video-sunleo-1.1.0/   xf86-video-suntcx-1.1.0/
xf86-video-tdfx-1.3.0/     xf86-video-tga-1.1.0/
xf86-video-trident-1.2.3/  xf86-video-tseng-1.1.1/
xf86-video-v4l-0.1.1/      xf86-video-vermilion-1.0.1/
xf86-video-vesa-1.3.0/     xf86-video-vga-4.1.0/
xf86-video-via-0.2.2/      xf86-video-vmware-10.15.0/
xf86-video-voodoo-1.1.1/   xf86-video-wsfb-0.2.1/
xf86-video-xgi-1.5.0/

# grep -r SymTabRec * | grep "¥[¥]"
:
xf86-video-amd-2.7.7.0/src/amd_driver.c:SymTabRec GeodeChipsets[] = {
xf86-video-apm-1.1.1/src/apm_driver.c:static SymTabRec ApmChipsets[] = {
xf86-video-ark-0.6.0/src/ark_driver.c:static SymTabRec ARKChipsets[] = {
xf86-video-ast-0.81.0/src/ast_driver.c:static SymTabRec ASTChipsets[] = {
xf86-video-ati-6.7.192/src/atidac.c:const SymTabRec ATIDACDescriptors[] =
:

```

実施することで、SymTabRec を定義している箇所が表示される。この情報をファイルに保存して Binary 検索を行うためのキーワードとする。サンプルとして、検索を行うために利用できるスクリプトを次に示す。

```
#!/bin/sh
grep -r $1 * | grep "¥[¥]" >> check.list
```

○利用方法

```
# ./grep.sh SymTabRec
```

※\$1 には検索キーワードを指定することができる。

#### (4) 検索キーワードの成形と生成

Binary ファイルを検索する際には、(3)で行った検索結果が全て必要となる訳ではなく、ソースコード中のシンボル情報があれば、Binary ファイルを検索することができる。ソースコードのシンボル情報の切り出しを行っておくことが必要になる。切り出しを行った結果を別ファイルに納めておき、Binary ファイルの検索に利用する。

##### grep 検索結果の成形部分

```
xf86-video-amd-2.7.7.0/src/amd_driver.c:SymTabRec GeodeChipsets[] = {
    ↑                               ↑
    この部分が必要                 この部分が必要
```

##### Binary 検索のための情報ファイル内容

```
amd GeodeChipsets
↑   ↑                               ①対象のソースコード名
①  ②                               ②対象のシンボル名
```

これらの作業を手作業にて実施する事は、時間を要するため、作業軽減ができるためのスクリプトを次に示す。このスクリプトは、ソースコードが変更になれば調整が必要となるので注意が必要である。

```
#!/bin/sh

rm -f pciids.lst
rm -f sym.lst
```

```

while read i
do
tmpsym=`echo $i |cut -d ':' -f 2`
chkpci=`echo "$tmpsym" |cut -d ' ' -f 1`

if [ "$chkpci" != "SymTabRec" ]; then
    chkpci=`echo $i | cut -d ':' -f 2`
fi

if [ "$chkpci" != "SymTabRec" ]; then
    chkpci=`echo $i | cut -d ' ' -f 2`
fi

if [ "$chkpci" != "SymTabRec" ]; then
    chkpci=`echo "$i" | cut -d ' ' -f 3`
fi

tmp=`echo "$i" | cut -d ':' -f 2`
chktype=`echo "$tmp" | cut -d ' ' -f 1`
chksrc=`echo "$i" | cut -d ':' -f 1`
if [ "$chktype" = "extern" ]; then
continue
fi

if [ "$chkpci" = "SymTabRec" ]; then
    j=`echo "$i" | cut -d ' ' -f 3`
    if [ "$j" = "=" ]; then
        j=`echo "$i" | cut -d ' ' -f 2`
    fi
    tmpids=`echo "$j" | sed -e 's/¥[¥]//`
    pcichk=`echo "$tmpids" | sed -e 's/;$//`
    pciids=`echo "$tmpids" | sed -e 's/¥={//`
    if [ $pciids = $pcichk ]; then
        echo "$chksrc $pciids" >>pciids.lst
        drvname=`echo $chksrc |cut -d '-' -f 3`
        if [ "$drvname" = "nsc" ]; then
            # nsc sorce work around.
            drvname=`echo "geode"`
        fi
        echo "$drvname $pciids" >>sym.lst
    fi
fi
done <./check.list

```

この他にベンダーの情報が必要となる。これは、xorg が server ソースと driver ソースが分かれたため、driver ソースコードだけでは、ベンダーの情報を入手することができないためである。ベンダー情報の生成には、手作業では時間を要するため、次に現在までに解っているベンダーの一覧を示す。この情報を利用することでベンダー情報を検索することが可能となる。

amd 00001022	glint 0000104C	nsc 0000100b	tseng 0000100C
apm 00001142	glide fffffff	nv 000010de	vesa 000000FF
ark 0000edd8	i128 0000105d	nv 000012d2	vga 000000FF
ast 00001a03	i740 00008086	rendition 00001163	vermilion
ati 00001002	i810 00008086	s3 00005333	via 00001106
chips 0000102c	impact fffffff	s3virage 00005333	vmware 000015ad
cirrus 00001013	intel 00008086	siliconmotion 0000126f	voodoo 0000121A
cyrix 00001078	imstt 000010e0	sis 00001039	wsfb 000000FF
dummy 000000FF	mga 0000102b	tdfx 0000121A	xgi 000018CA
fbdev 000000FF	neomagic 000010c8	tga 00001011	
glint 00003d3d	newport fffffff	trident 00001023	

## (5) Binary ファイルの入手

Binary ファイルの検索を実施する前に、検索対象となる Binary ファイルを入手することが必要である。入手手順としては、以下の方法がある。

- ① xorg-x11-xxxxx.ixxx.rpm となっているパッケージを入手する。
- ② xorg-x11-xxx-debug-xxxxx.ixxx.rpm となっているととなっているパッケージを入手する。
- ③ xorg-x11-xxx-debug-xxxxx.ixxx.rpm となっているととなっているパッケージをコンパイルする。

①～③に分かれるのは、各ディストリビューションの配布方法や本家でのソースコードの配布形態の変化により分かれる。よって、ディストリビューション毎の特性を判断し、入手方法を取捨選択することが必要となる。今回の調査で対象としてディストリビューションでは、次に示すことを実施した。

表 C-8:ディストリビューション別入手方法

ディストリビューション名	利用した手順
Fedora 7	③
Open SUSE 10.2	②
Knoppix 5.1.1	-
Ubuntu 7.04	③
Turbolinux FUJI	①

①の場合

この方法は、ディストリビューションが配布している Binary パッケージにシンボル情報が含まれている場合に適用することができる。今回の調査では、Turbolinux FUJI だけが、この方法を実施することができた。

```
# cd /work/distro-work/  
# mkdir xorg  
#  
# cd xorg  
# rpm2cpio xorg-x11-xxxxx.ixxx.rpm | cpio -id  
#  
# mkdir -p /work/xorg/driver/binary/distrol/  
#  
# cp usr/X11R6/lib/modules/drivers/* /work/xorg/driver/binary/distrol/
```

## ②の場合

この方法は、ディストリビューションが debug 用の Package を配布している場合に適用できる。今回の調査では、OpenSUSE 10.2 だけが、この方法を実施することができた。

```
# cd /work/distro-work/  
# mkdir xorg  
#  
# cd xorg  
# rpm2cpio xorg-x11-xxxxx-debuginfo.ixxx.rpm | cpio -id  
#  
# mkdir -p /work/xorg/driver/binary/distro2/  
#  
# cp usr/lib/debug/usr/xorg/lib/modules/drivers/* /work/xorg/driver/binary/distro2/  
#  
# rpm2cpio ../xorg-x11-xxxxx.ixxx.rpm | cpio -id  
#  
# cp usr/lib/xorg/modules/driver/* /work/xorg/driver/binary/distro2/
```

## ③の場合

この方法は、コンパイルの作業が必要となるため、各ディストリビューションをインストールした環境で、コンパイル可能な環境が必要になる。また、ディストリビューションにて、Debug Package の生成ができる環境も必要になる。今回の調査では、Fedora 7,Ubuntu にて、この方法を実施した。

## Fedora 7 の場合

```
# rpm -ivh xorg-x11-xxxxx.src.rpm
#
# cd /usr/src/redhat/SPEC
#
# rpmbuild -ba xorg-x11-drv-xxxx.spec ← Architecture に依存することも
#                                     こともある。
# cd ../RPM/i386
# cp *.rpm /work/distro-work
#
# cd /work/distro-work/
# mkdir xorg
#
# cd xorg
# rpm2cpio ../xorg-x11-xxxxx-debuginfo.i386.rpm | cpio -id
# rpm2cpio ../xorg-x11-xxxxx.i386.rpm | cpio -id
#
# mkdir -p /work/xorg/driver/binary/distro3/
#
# cp usr/lib/debug/usr/xorg/lib/modules/drivers/* /work/xorg/driver/binary/distro3/
# cp usr/lib/xorg/modules/driver/* /work/xorg/driver/binary/distro3/
```

## Ubuntu の場合

```
# apt-get source xserver-xorg-video-xxxx
#
# dpkg-source -x xserver-xorg-video-xxxx.dsc
#
# cd xserver-xorg-video-xxxx
#
# dpkg-buildpackage -rfakeroot -us -uc ← 依存関係の解決が必要になる場合がある。
#
# cd xserver-xorg-video-xxxx
#
# cd obj-i486-linux-gnu
# cd src
# cd .libs
#
# cp *.so /work/xorg/driver/binary/distro3
```

※Ubuntu での依存関係の解決

Debian 系のディストリビューションでは Package 間の依存関係が解決できずにコンパイルが実行できない場合がある。この場合には、必要なパッケージを入手し、インストールすることで解決ができる。依存関係の解決する方法を次に示す。

#### Ubuntu の場合

```
# dpkg-buildpackage -rfakeroot -us -uc
#
dpkg-buildpackage: source package is xserver-xorg-video-ati
dpkg-buildpackage: source version is 1:6.6.3-2ubuntu6
dpkg-buildpackage: source changed by Timo Aaltonen <tepsipakki@ubuntu.com>
dpkg-buildpackage: host architecture i386
dpkg-buildpackage: source version without epoch 6.6.3-2ubuntu6
dpkg-checkbuilddeps: Unmet build dependencies: quilt ← この行に注目する
dpkg-buildpackage: Build dependencies/conflicts unsatisfied; aborting.
dpkg-buildpackage: (Use -d flag to override.)

# apt-get install quilt
#
※この作業を行うことで、コンパイルに必要な状態にすることができる。
```

#### (6) Binary ファイルの検索

Binary ファイルの検索には、2つの方法がある。1つは、debuginfo が付いているパッケージを利用して検索する方法、もう1つは、Binary ファイルから直接検索する方法がある。2つの手順の違いは、検索対象とするファイルが変わるだけで、実施する検索手順は変わらない。今回は、debuginfo が付いているパッケージを利用した方法を次に実際に検索する場合の手順を示す。検索を行う際には、objdump コマンドを利用するので、次に示すパッケージが必要である。

表 C-9:objdump コマンドを利用するために必要なパッケージ

パッケージ名	説明
binutils-2.16.91.0.1-1	バイナリファイルのユーティリティツール

```
# cd /work/xorg/driver/binary/distro3/
#
# objdump -j .data -t amd_drv.so.debug | grep ${シンボル名}
00032ea0 l      0 .data 00000018 .hidden GeodeChipsets ← シンボルに適した位置
                                     が標記される
```

```
# objdump -j .data -t --start-address=0x32ea0 --stop-address=0x32eb8 amd_drv.so >
check.dt
# cat check.dt
amd_drv.so:      ファイル形式 elf32-i386

セクション .data の内容 :
32ea0 81200000 75f10200 30000000 7df10200  . . .u...0...}...
32eb0 ffffffff 00000000  . . . . .
```

作成した check.dt とベンダー情報を利用して、以下のフォーマットを作成するし、データベースに登録する CSV ファイルを作成する。

```
# cat data.csv

vendor id,device id,index,driver name
"00001022","00002081","0002f175","*", "amd"
"00001022","00000030","0002f17d","*", "amd"
```

上記の手順を実施することで、CSV ファイルを作成することができるが、多数に渡る作業となり時間を要する。サンプルとして、次に利用できるサンプルのスクリプトを示す。

#### Binary ファイルからの Dump を実施するシェルスクリプト

```
#!/bin/sh

echo "#!/bin/sh">check_csv.sh
echo "rm -rf csv/">>check_csv.sh
echo "mkdir -p csv/">>check_csv.sh
mkdir -p dt_lst

while read i
do
drvname=`echo $i | cut -d ' ' -f 1`
echo "driver name -> $drvname"
objdump -j .data -t "$drvname"_drv.so.debug >check.sym ← ファイル名の変更は必要
mkdir -p dt_lst/$drvname

chksym=`echo $i | cut -d ' ' -f 2`
echo "check grep!! $chksym"
j=`grep $chksym ./check.sym`
echo "$j"
if [ "$j" != "" ]; then
```

```

sym=`echo "$j"|cut -d ' ' -f 9`
echo "Symbol : $sym"

    if [ "$chksym" != "$sym" ] && [ "$drvname" = "amd" ]; then
        sym=`echo "$j"|cut -d ' ' -f 10`
        echo "Symbol : $sym"
    fi
if [ "$chksym" = "$sym" ]; then
    addr=`echo "$j"|cut -d ' ' -f 1`
    k=`echo "$j"|cut -d ' ' -f 8`
    length=`echo "$k"|cut -d ' ' -f 2`
    x=`echo $addr | tr ' [a-z]' '[A-Z]'`
    tmpaddr=`dc -e "16i $x p"`
    x=`echo $length | tr ' [a-z]' '[A-Z]'`
    tmpflen=`dc -e "16i $x p"`
    tmpstop=`expr "$tmpaddr" + "$tmpflen"`
    hexaddr=`dc -e "16o $tmpstop p"`
    stopaddr=`echo "0x$hexaddr"`
    startaddr=`echo "0x$addr"`
objdump -j .data -s --start-address=$
$startaddr --stop-address=$stopaddr ¥
"$drvname"_drv.so>dt_lst/$drvname/$sym.dt

    if [ "$sym" = "XGIChipsets" ]; then
        # XGI work around
        echo ". /create_csv2.sh¥
dt_lst/$drvname/$sym.dt xgi">>check_csv.sh
    else
        echo ". /create_csv2.sh¥
dt_lst/$drvname/$sym.dt $drvname">>check_
csv.sh
    fi
    else
        echo "!!!! symbol not found !!!!!"
    fi
fi

```

## DumpしたファイルからCSVを自動生成するスクリプト

```
#!/bin/sh
```

```

function byteswap () {
local dt1=`echo "$1" | cut -c 1,2`
local dt2=`echo "$1" | cut -c 3,4`
local dt3=`echo "$1" | cut -c 5,6`
local dt4=`echo "$1" | cut -c 7,8`
swapresult=`echo "$dt4$dt3$dt2$dt1"`
}

n=1
declare -a ids_dt

while read i
do
addr=`echo "$i" | cut -d ' ' -f 1`
data1=`echo "$i" | cut -d ' ' -f 2`
data2=`echo "$i" | cut -d ' ' -f 3`
data3=`echo "$i" | cut -d ' ' -f 4`
data4=`echo "$i" | cut -d ' ' -f 5`

if [ "$addr" = "" ] ||
[ "$data1" = "" ] ||
[ "$data2" = "" ] ||
["$data3" = "" ] ||
[ "$data4" = "" ]; then
echo $ascii
continue
fi

byteswap $data1
ids_dt[n]=$swapresult
n=`expr $n + 1`
byteswap $data2
ids_dt[n]=$swapresult
n=`expr $n + 1`

byteswap $data3
ids_dt[n]=$swapresult
n=`expr $n + 1`
byteswap $data4
ids_dt[n]=$swapresult
n=`expr $n + 1`

```

```

done <$1
### vendor list file ###
declare -a ven_dt
c=0
while read i
do
chkven=`echo $i |cut -d ' ' -f 1`
echo $chkven
echo $2
if [ "$chkven" = "$2" ]; then
    vendor=`echo $i |cut -d ' ' -f 2`
    break
fi
done <./vendor.lst
### creat csv file ###
name=`basename $1 .dt`
m=2
echo "vendor id,device id,index,dirver name">csv/$name.csv

for i in `seq 1 $n`
do
if [ $i = $m ]; then
echo "test dump!!"
if [ "${ids_dt[$(($m-1))]}" = "" ] ||
[ "${ids_dt[$m]}" = "" ]; then
continue
elif [ "${ids_dt[$(($m-1))]}" = "ffffffff" ] && [ "${ids_dt[$m]}" = "00000000" ] ;
then
continue
fi
tmpven=`expr ${ids_dt[$(($m-1))]} : "¥(....¥)"`
tmpdev=`expr ${ids_dt[$(($m-1))]} : "....¥(....¥)"`
if [ $2 = "xgi" ] && [ "$name" = "XGIChipsets" ]; then
echo "¥"$vendor¥",¥"0000$tmpdev¥",¥"${ids_dt[$m]}¥",¥"*¥",¥"sis¥"">>csv/$name.csv
else
echo
echo
"¥"$vendor¥",¥"0000$tmpdev¥",¥"${ids_dt[$m]}¥",¥"*¥",¥"$2¥"">>csv/$name.csv
fi
m=`expr $m + 2`

fi
done

```

```

# cd /work/xorg/driver/binary/distro3/
#
# cp /work/xorg/driver/sym.lst .
#
# cp /work/xorg/driver/vendor.lst .
#
# ./check_binary.sh ← シェルスクリプト
#
# ./check_csv.sh ← 上記のシェルから自動生成するシェルスクリプト
#
# cd csv/
#
# ls
ARKChipsets.csv      I810Chipsets.csv    SMIChipsets.csv
ASTChipsets.csv      I830BIOSChipsets.csv SavageChipsets.csv
ApmChipsets.csv      IMSTTChipsets.csv   TDFXChipsets.csv
CHIPSChipsets.csv    MGAChipsets.csv     TGAChipsets.csv
CIRChipsets.csv      NEOChipsets.csv     TRIDENTChipsets.csv
CYRIXChipsets.csv    NVKnownChipsets.csv TsengChipsets.csv
DUMMYChipsets.csv    NewportChipsets.csv VESAChipsets.csv
FBDevChipsets.csv    R128Chipsets.csv    VIAChipsets.csv
GLINTChipsets.csv    RADEONChipsets.csv  VMWAREChipsets.csv
GLINTVGAChipsets.csv S3Chipsets.csv      VoodooChipsets.csv
GenericChipsets.csv  S3VChipsets.csv     XGIChipsets.csv
I128Chipsets.csv    SISChipsets.csv     renditionChipsets.csv
I740Chipsets.csv    SISUSBChipsets.csv

```

### C.6.2. 各ディストリビューションが配布しているテキスト情報から入手する場合

ディストリビューションのよっては、Video カードの情報をテキストファイルで配布している場合がある。これらの情報を利用することにより、Video カード情報をデータベースへの登録データを生成する事ができる。この方法は、Fedora7,Knoppix にて利用できる方法である。

#### Fedora で行う場合

```

# rpm2cpio ../xorg-x11-xxxxx-debuginfo.ixxx.rpm | cpio -id
or
# rpm2cpio ../xorg-x11-xxxxx.ixxx.rpm | cpio -id
#

```

```
# cd usr/share/hwdata/videoaliases
# cp * /work/xorg/driver/text/distrol
# mi2csv /work/xorg/driver/text/distrol/xxx.xinf >xxx.csv
# cp xxx.csv /work/xorg/driver/text/distrol/
```

### Knoppixで行う場合

※CD boot にて起動をしてから以下の手順を実施する。

```
#
# cd /usr/share/hwdata/
#
#
# scp videoaliases `ホスト名`:/work/xorg/driver/text/distrol/
#
```

※Server 側での作業

```
# cd /work/xorg/driver/text/distrol
#
# mv videoaliases videoaliases.xinf
#
# mi2csv `pwd`/videoaliases.xinf >videoaliases.csv
#
```

### C.6.3. 各ディストリビューションが配布しているソースコードから情報を入手する場合

ソースコードを検索するには、前項(1)~(3)に定義している手順を実施することで、該当のデバイスドライバ情報を得ることができる。この場合、ソースコードでは#define 定義という定義にてサポートデバイスを定義している。このため、#define 定義にて定義されている名前の検索も必要となる。

#### 定義箇所例

```
/* Supported chipsets */
static SymTabRec VoodooChipsets[] = {
    { PCI_CHIP_VOODOO1, "Voodoo 1" },
    { PCI_CHIP_VOODOO2, "Voodoo 2" },
    {-1, NULL }
};
```

#### define 定義の検索

```
# cd /work/xorg/driver/distrol
○Fedora, OpenSUSE, TurboLinux の場合
# rpm2cpio xorg-x11-xxx-voodooxx.src.rpm | cpio -id
```

```
#
# tar jxvf xorg-x11-xxxx.tar.bz2
# cd xf86-video-vooodoo-1.1.1/
# grep -r PCI_CHIP_VOODOO1 *
#
○Ubuntuの場合
# apt-get source xserver-xorg-video-vooodooxxxx
#
# cd xserver-xorg-video-vooodoo_xxxx
# grep -r PCI_CHIP_VOODOO1 *
#
```

#### C.6.4. データベースの投入データ生成(パッケージ情報編)

各ディストリビューションにて配布しているパッケージから、パッケージ情報を取得して、データベースに登録する。パッケージ情報は、Debian 系パッケージと RPM パッケージにて取得できる形態が違うので、CSV フォーマット(テキスト)に統一し、データベースに登録しやすい形に成形する。

#### C.6.5. Debian 系パッケージの場合

前項に示した手順にて、パッケージの情報を得ることができる。配布されているパッケージは多々あるため、自動的にパッケージ情報を取得するスクリプトを次に示す。

パッケージの一覧を取得するスクリプト

```
#!/bin/sh

#apt-cache pkgnames > pkg.lst
dpkg -l >pkg.lst
```

パッケージの情報を取得するスクリプト

```
#!/bin/sh
mkdir -p csv

while read i
do
chktop=`echo $i |cut -d ' ' -f 1`
if [ "$chktop" = "ii" ] || [ "$chktop" = "hi" ]; then
chkpkg=`echo $i |cut -d ' ' -f 2`
echo "$chkpkg"
else
continue
fi
```

```

echo "--- list pkg information ---"
apt-cache show "$chkpkg" >./csv/"$chkpkg".csv

echo "--- list pkg list ---"
echo "" >> ./csv/"$chkpkg".csv
echo "¥Install list¥"," >> ./csv/"$chkpkg".csv
chkinstlist=`dpkg -L "$chkpkg"`
echo "¥$chkinstlist¥"">> ./csv/"$chkpkg".csv

echo "--- list pkg requirelist ---"
echo "" >> ./csv/"$chkpkg".csv
echo "¥Requires list¥"," >> ./csv/"$chkpkg".csv
chkreqlist=`apt-cache showpkg "$chkpkg"`

echo "¥$chkreqlist¥"">> ./csv/"$chkpkg".csv
done < ./pkg.lst

```

上記スクリプトを実行することで、データベースに登録するためのパッケージ情報を得ることが出来る。この情報を基に、前項に示した登録用の CSV に変換する。CSV 変換する際には、各ディストリビューションで取得できる内容が異なる。このため、各ディストリビューションに合わせた形で、CSV 化する必要がある。CSV 化するために必要となるサンプルスクリプトを次に示す。

#### Ubuntu での CSV 変換スクリプト

```

#!/bin/sh
mkdir -p dbcsv

for i in `ls *.csv`; do
declare -a descripiton
descnt=1

declare -a install
instcnt=1

declare -a require
reqcnt=1

while read j
do

chkinfo=`echo "$j" |cut -d ':' -f 2`
chkpkg=`echo "$j" |cut -d ':' -f 1`

```

```

case $chpkg in
"Package")
pkgname=`echo "$chkinfo"`
;;
"Status")
pkgstatus=`echo "$chkinfo"`
;;
"Priority")
priority=`echo "$chkinfo"`
;;
"Section")
pkggroup=`echo "$chkinfo"`
;;
"Installed-Size")
pkgsize=`echo "$chkinfo"`
;;
"Maintainer")
pkgpackager=`echo "$chkinfo"`
;;
"Architecture")
pkgarch=`echo "$chkinfo"`
;;
"Source")
pkgsource=`echo "$chkinfo"`
;;
"Version")
pkgverison=`echo "$chkinfo"`
;;
"Replaces")
pkgreplaces=`echo "$chkinfo"`
;;
"Provides")
if [ "$chkinfo" != "" ]; then
pkgprovides=`echo "$chkinfo"`
echo "$pkgprovides"
fi
;;
"Depends")
pkgdepends=`echo "$chkinfo"`
;;

```

```

"Conflicts")
pkgconflicts=`echo "$chkinfo"`
;;
"Filename")
pkgfilename=`echo "$chkinfo"`
;;
"Size")
pkgfilesize=`echo "$chkinfo"`
;;
"Original-Maintainer")
pkgorinmaintainer=`echo "$chkinfo"`
;;
"MD5sum")
pkgmd5=`echo "$chkinfo"`
;;
"SHA1")
pkgsha1=`echo "$chkinfo"`
;;
"SHA256")
pkgsha256=`echo "$chkinfo"`
;;
"Origin")
echo "$chkinfo"
pkgorigin=`echo "$chkinfo"`
;;
"Task")
echo "$chkinfo"
pkgtask=`echo "$chkinfo"`
;;
"Description-ja" | "Description")
pkgdescription=`echo "$chkinfo"`
while [ "$j" != "" ]
do
read str
chkstop=`echo $str | cut -d ':' -f 1`
if [ "$chkstop" = "Bugs" ]; then
pkgbug=`echo "$str" | cut -d ' ' -f 2`
break
fi
description[descnt]=`echo "$str"`
descnt=`expr $descnt + 1`

```

```

j=`echo $str`
done
;;
*)
### install list ###
chklist=`echo "$j" |cut -d ',' -f 1`
if [ "$chklist" = "¥"Install list¥" ]; then
pkginstlist=`echo "$chklist,"`

while [ "$j" != "" ]
do
read str
install[instcnt]=`echo "$str"`
instcnt=`expr $instcnt + 1`
j=`echo $str`
done
reqcnt=1
fi

require[reqcnt]=`echo "$j"`
reqcnt=`expr $reqcnt + 1`
;;
esac
done < $i

### Create CSV format(common header) #####
echo "¥"NAME¥",¥"$pkgname¥"" > ./dbcsv/$i
echo "¥"Group¥",¥"$pkggroup¥"" >> ./dbcsv/$i
echo "¥"Size¥",¥"$pkgsize¥"" >> ./dbcsv/$i
echo "¥"Packager¥",¥"$pkgpackager¥"" >> ./dbcsv/$i
echo "¥"Architecture¥",¥"$pkgarc¥"" >> ./dbcsv/$i
echo "¥"Source RPM¥",¥"$pkgsource¥"" >> ./dbcsv/$i
echo "¥"Version¥",¥"$pkgversion¥"" >> ./dbcsv/$i
echo "¥"Signature¥",¥"MD5:$pkgmd5 SHA1:$pkgshal SHA256:$pkgsha256 ¥"" >> ./dbcsv/$i
echo "¥"Description¥",¥"$pkgdescription" >> ./dbcsv/$i
cnt=1
for cnt in `seq 1 $descent`;do
chkdesc=`echo "${description[$cnt]}"`
cnt=`expr $cnt + 1`
if [ "$chkdesc" = "" ]; then
continue

```

```

else
echo "$chkdesc" >> ./dbcsv/$i
fi
done
echo "" >> ./dbcsv/$i
### Create CSV format(proper header) #####
echo "Provides", "$pkgprovides" >> ./dbcsv/$i
echo "Priority", "$priority" >> ./dbcsv/$i
echo "Confilicts", "$pkgconfilicts" >> ./dbcsv/$i
echo "Bugs", "$pkgbug" >> ./dbcsv/$i
echo "Origin", "$pkgorigin Task:$pkgtask" >> ./dbcsv/$i
echo "Original-Maintainer", "$pkgorinmaintainer" >> ./dbcsv/$i
echo "Filename", "$pkgfilename Size:$pkgsize" >> ./dbcsv/$i

### Create Install list ###
echo "" >> ./dbcsv/$i
echo "$pkginstlist" >> ./dbcsv/$i
cnt=1
for cnt in `seq 1 $instcnt`;do
chkinst=`echo "${install[$cnt]}"`
cnt=`expr $cnt + 1`
if [ "$chkinst" = "" ]; then
continue
else
echo "$chkinst" >> ./dbcsv/$i
fi
done

echo "" >> ./dbcsv/$i
### Create Require list ###
cnt=1
for cnt in `seq 1 $reqcnt`;do
chkreq=`echo "${require[$cnt]}"`
cnt=`expr $cnt + 1`
if [ "$chkreq" = "" ]; then
continue
else
echo "$chkreq" >> ./dbcsv/$i
fi
done

```

## Knoppix での CSV 変換スクリプト

```
#!/bin/sh
mkdir -p dbcsv

for i in `ls *.csv`; do

declare -a descripiton
desct=1

declare -a install
instcnt=1

declare -a require
reqcnt=1

while read j
do

chkinfo=`echo "$j" |cut -d ':' -f 2`
chkpkg=`echo "$j" |cut -d ':' -f 1`

case $chkpkg in
"Package")
pkgname=`echo "$chkinfo"`
;;
>Status")
pkgstatus=`echo "$chkinfo"`
;;
"Priority")
priority=`echo "$chkinfo"`
;;
"Section")
pkggroup=`echo "$chkinfo"`
;;
"Installed-Size")
pkgsize=`echo "$chkinfo"`
;;
"Maintainer")
pkgpackager=`echo "$chkinfo"`
```

```

;;
"Architecture")
pkgarch=`echo "$chkinfo"`
;;
"Source")
pkgsources=`echo "$chkinfo"`
;;
"Version")
pkgversion=`echo "$chkinfo"`
;;
"Replaces")
pkgreplaces=`echo "$chkinfo"`
;;
"Provides")
if [ "$chkinfo" != "" ]; then
pkgprovides=`echo "$chkinfo"`
fi
;;
"Depends")
pkgdepends=`echo "$chkinfo"`
;;
"Conflicts")
pkgconflicts=`echo "$chkinfo"`
;;
"Description")
pkgdescription=`echo "$chkinfo"`
while [ "$j" != "" ]
do
read str
description[descnt]=`echo "$str"`
descnt=`expr $descnt + 1`
j=`echo $str`
done
;;

*)
### install list ###
chklist=`echo "$j" |cut -d ',' -f 1`
if [ "$chklist" = "Y"Install listY ]; then
pkginstalllist=`echo "$chklist,"`

```

```

while [ "$j" != "" ]
do
read str
install[instcnt]=`echo "$str"`
instcnt=`expr $instcnt + 1`
j=`echo $str`
done
reqcnt=1
fi
require[reqcnt]=`echo "$j"`
reqcnt=`expr $reqcnt + 1`
;;
esac
done < $i

### Create CSV format(common header) #####
echo "$NAME", "$pkgname" > ./dbcsv/$i
echo "$Group", "$pkggroup" >> ./dbcsv/$i
echo "$Size", "$pkgsize" >> ./dbcsv/$i
echo "$Packager", "$pkgpackager" >> ./dbcsv/$i
echo "$Architecture", "$pkgarc" >> ./dbcsv/$i
echo "$Source RPM", "$pkgsourc" >> ./dbcsv/$i
echo "$Version", "$pkgversion" >> ./dbcsv/$i
echo "$Signature", "$none" >> ./dbcsv/$i
echo "$Description", "$pkgdescription" >> ./dbcsv/$i
cnt=1
echo $descnt
for cnt in `seq 1 $descnt`;do
chkdesc=`echo "${description[$cnt]}"`
cnt=`expr $cnt + 1`
if [ "$chkdesc" = "" ]; then
continue
else
echo "$chkdesc" >> ./dbcsv/$i
fi
done
echo "$" >> ./dbcsv/$i
### Create CSV format(proper header) #####
echo "$Provides", "$pkgprovides" >> ./dbcsv/$i
echo "$Priority", "$priority" >> ./dbcsv/$i
echo "$Depends", "$pkgdepends" >> ./dbcsv/$i

```

```

echo "$Conflicts","$pkgconflicts" >> ./dbcsv/$i
echo "$Status","$pkgstatus" >> ./dbcsv/$i
echo "$Replaces","$pkgreplaces" >> ./dbcsv/$i
echo "$License","none" >> ./dbcsv/$i

### Create Install list ###
echo "" >>./dbcsv/$i
echo "$pkginstlist" >>./dbcsv/$i
cnt=1
echo $instcnt
for cnt in `seq 1 $instcnt`;do
chkinst=`echo "${install[$cnt]}"`
cnt=`expr $cnt + 1`
if [ "$chkinst" = "" ]; then
continue
else
echo "$chkinst" >> ./dbcsv/$i
fi
done

### Create Require list ###
echo "" >>./dbcsv/$i
cnt=1
echo $reqcnt
for cnt in `seq 1 $reqcnt`;do
chkreq=`echo "${require[$cnt]}"`
cnt=`expr $cnt + 1`
if [ "$chkreq" = "" ]; then
continue
else
echo "$chkreq" >> ./dbcsv/$i
fi
done

done

```

### C.6.6. RPM 系パッケージの場合

前項に示した手順でパッケージの情報を得ることができる。配布されているパッケージは多々あるため、自動的にパッケージ情報を取得するスクリプトを次に示す。

RPM 系での CSV 変換スクリプト

---

```

#!/bin/sh

mkdir -p csv
RPMCOMMAND=`echo "rpm -qp --nosignature"`
RPMINSTINFO=`echo "rpm -qpl --nosignature"`
RPMREQINFO=`echo "rpm -qpR --nosignature"`

for i in `ls *.rpm`; do
echo $i

csvname=`$RPMCOMMAND $i`
echo "pkgname"
pkgname=`$RPMCOMMAND $i --queryformat "%{NAME}¥n"`
echo "pkgreloptions"
pkgreloptions=`$RPMCOMMAND $i --queryformat "%|PREFIXES?{[%{PREFIXES}]}: {not
relocateable}|¥n"`
echo "pkgversion"
pkgversion=`$RPMCOMMAND $i --queryformat "%{VERSION}¥n"`
echo "pkgvendor"
pkgvendor=`$RPMCOMMAND $i --queryformat "%{VENDOR}¥n"`
echo "pkgrelease"
pkgrelease=`$RPMCOMMAND $i --queryformat "%{RELEASE}¥n"`
echo "pkgdate"
pkgdate=`$RPMCOMMAND $i --queryformat "%{builddate:date}¥n"`
echo "pkgghost"
pkgghost=`$RPMCOMMAND $i --queryformat "%{BUILDHOST}¥n"`
echo "pkgsize"
pkgsize=`$RPMCOMMAND $i --queryformat "%{SIZE}¥n"`
echo "pkglicense"
pkglicense=`$RPMCOMMAND $i --queryformat "%{LICENSE}¥n"`
echo "pkgsignature"
pkgsignature=`$RPMCOMMAND $i --queryformat "%{SIGPG:pgpsig}¥n"`
echo "pkgpackager"
pkgpackager=`$RPMCOMMAND $i --queryformat "%{Packager}¥n"`
echo "pkgurl"
pkgurl=`$RPMCOMMAND $i --queryformat "%{url}¥n"`
echo "pkgsummary"
pkgsummary=`$RPMCOMMAND $i --queryformat "%{SUMMARY}¥n"`
echo "pkgdescription"
pkgdescription=`$RPMCOMMAND $i --queryformat "%{Description}¥n"`
echo "pkgsource"
pkgsource=`$RPMCOMMAND $i --queryformat "%{SOURCERPM}¥n"`

```

```

echo "pkggroup"
pkggroup=`$RPMCOMMAND $i --queryformat "%{GROUP}¥n"`
echo "pkgarc"
pkgarc=`$RPMCOMMAND $i --queryformat "%{arch}¥n"`

### Query Format to CSV format(common header) ###
echo "¥NAME¥",¥$pkgname¥" >>./csv/"$csvname".csv
echo "¥Group¥",¥$pkggroup¥" >>./csv/"$csvname".csv
echo "¥Size¥",¥$pkgsize¥" >>./csv/"$csvname".csv
echo "¥Packager¥",¥$pkgpackager¥" >>./csv/"$csvname".csv
echo "¥Architecture¥",¥$pkgarc¥" >>./csv/"$csvname".csv
echo "¥Source RPM¥",¥$pkgsources¥" >>./csv/"$csvname".csv
echo "¥Version¥",¥$pkgversion-¥pkgrelease¥" >>./csv/"$csvname".csv
echo "¥Signature¥",¥$pkgsignature¥" >>./csv/"$csvname".csv
echo "¥Description¥",¥$pkgdescription¥" >>./csv/"$csvname".csv

### Query Format to CSV format(proper header) ###
echo "¥URL¥",¥$pkgurl¥" >>./csv/"$csvname".csv
echo "¥Summary¥",¥$pkgsummary¥" >>./csv/"$csvname".csv
echo "¥Build Date¥",¥$pkgdate¥" >>./csv/"$csvname".csv
echo "¥Build Host¥",¥$pkgghost¥" >>./csv/"$csvname".csv
echo "¥Vendor¥",¥$pkgvendor¥" >>./csv/"$csvname".csv
echo "¥License¥",¥$pkglicense¥" >>./csv/"$csvname".csv
echo "¥Relocations¥",¥$pkgrelocations¥" >>./csv/"$csvname".csv

echo "">>./csv/"$csvname".csv
echo "¥Install list¥",">>./csv/"$csvname".csv
pkginstlist=`$RPMINSTINFO $i`
echo "¥$pkginstlist¥" >>./csv/"$csvname".csv

echo "">>./csv/"$csvname".csv
echo "¥Requires list¥",">>./csv/"$csvname".csv
pkgreqlist=`$RPMREQINFO $i`
echo "¥$pkgreqlist¥" >>./csv/"$csvname".csv

done

```

## C.7. データベースへの自動投入

生成した CSV ファイルをデータベースに投入する場合には、次のディレクトリを用意する必要がある。ディレクトリの配置が完了した後、実際に CSV ファイルの配置をする。ファイル配置は、

Linux Kernel, X Window システム, アプリケーションの CSV を当該のディレクトリに配置する。CSV ファイルの配置が完了した後、データベースとの連携をするための情報ファイルの作成を行い、当該ディレクトリに配置する。

/Database	← Data Base に Update するための root ディレクトリ
-- distrol	← ディストリビューション毎のデータディレクトリ
-- applications	← パッケージ情報のディレクトリ
-- kernel	← Linux kernel 情報ディレクトリ
-- driver-info	← モジュール情報ディレクトリ
-- ieee1394	← IEEE1394 モジュール情報
-- pci	← PCI モジュール情報
-- pcmcia	← PCMCIA モジュール情報
-- pnp	← ISA PNP モジュール情報
-- serio	← シリアルモジュール情報
`-- usb	← USB モジュール情報
-- kernel-bin	← boot kernel ドライバ情報
`-- map	← Linux Kernel のマップ情報
`-- x-window	← X-Window System 情報ディレクトリ
-- share	← Linux デバイス情報
`-- template	← データベースアップ用のディレクトリテンプレート
-- applications	
-- kernel	
-- driver-info	
-- kernel-bin	
`-- map	
`-- x-window	

## (1) ディレクトリへの配置

Linux Kernel のモジュール情報(driver-info)以外のディレクトリは、単純に配置が可能となる。配置方法は、サーバ上での作業であった場合には「コピー」コマンドにて行うことがで

きる。リモートより行う場合には、ファイル転送のコマンド類を使い実施する。

```
$ su-
Password
# cd /tmp
#
※サーバ上での作業
# cp driver-info/* /Database/distrol/kernel/driver-info/.
#
※リモートからの作業
# scp -r driver-info/ /Database/distrol/kernel/driver-info/.
```

Linux Kernel のモジュール情報だけは、ファイル数が多くなるのでデータベースへの登録を効率良くするために、デバイスが接続されているバス単位に分けて配置する。

配置にあたっては、CVS ファイル内の情報に従いコピー処理を行うことになる。サンプルとして、次に配置用のシェルスクリプトを示す。

```
#!/bin/bash
mkdir -p usb
mkdir -p pci
mkdir -p pcmcia
mkdir -p ieeel394
mkdir -p pnp
mkdir -p serio

for movef in $( ls *.csv); do
usb=`grep "alias","usb" $movef| cut -d , -f 2`
pci=`grep "alias","pci" $movef| cut -d , -f 2`
pcmcia=`grep "alias","pcmcia" $movef| cut -d , -f 2`
ieeel394=`grep "alias","ieeel394" $movef| cut -d , -f 2`
pnp=`grep "alias","pnp" $movef| cut -d , -f 2`
serio=`grep "alias","serio" $movef| cut -d , -f 2`

    if [ "$usb" != "" ]; then
        mv $movef usb/.
        echo "move to usb driver file."
    fi

if [ "$pci" != "" ]; then
```

```

        mv $movef pci/.
        echo "move to pci driver file."
fi
if [ "$pcmcia" != "" ]; then
    mv $movef pcmcia/.
    echo "move to pcmcia driver file."
fi

if [ "$ieeel394" != "" ]; then
    mv $movef ieeel394/.
    echo "move to ieeel394 driver file."
fi

if [ "$pnp" != "" ]; then
    mv $movef pnp/.
    echo "move to pnp driver file."
fi

if [ "$serio" != "" ]; then
    mv $movef serio/.
    echo "move to serio driver file."
fi

done

```

## (2) Unique Key の配置

データベースへの登録については、重複する情報の登録を避けるため、各ディレクトリに次に示すテキスト形式のファイルを配置する。

表 C-10: Unique Key ファイルの種類

ファイル名	配置ディレクトリ	説明
Distro.info	/Database/distro1	ディストリビューション情報登録時の Unique Key
Kernel.info	/Database/distro1/kernel	Kernel パッケージ情報登録時の Unique Key
X.info	/Database/distro1/x-window	X Window パッケージ情報登録時の Unique Key

配置する各ファイルの内容は次に示すとおりである。

表 C-11:Distro.info の内容

UniqueKey 名	意味	記述例
Linux Distro 名	ディストリビューション名	Turbolinux FUJI
Version	ディストリビューションのバージョン情報	11
Architecture	CPU アーキテクチャ	X32

表 C-12:Kerenl.info の内容

UniqueKey 名	意味	記述例
Package 名	Kernel パッケージ名	Kernel-2.6.13-19
Version	Kernel Version	2.6.13

表 C-13:x.info の内容

UniqueKey 名	意味	記述例
Package 名	X Window パッケージ名	xorg-x11-6.8.2-50
Version	X Window Version	6.8.2

2006年度  
オープンソースソフトウェア基盤整備事業  
「Linux デистриビューション仕様記述様式に関する調査」

自動判別ツール(プロトタイプ)

外部仕様書

2007年10月

株式会社三菱総合研究所

本文書は、OpenOffice.org を利用して作成された。