

openSUSE

13.1

www.suse.com

2016/11/13

KVM を利用した仮想化



KVM を利用した仮想化

Copyright © 2006–2016 SUSE LLC and contributors. All rights reserved.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or (at your option) version 1.3; with the Invariant Section being this copyright notice and license. A copy of the license version 1.2 is included in the section entitled 「GNU Free Documentation License」.

For SUSE or Novell trademarks, see the Novell Trademark and Service Mark list <http://www.novell.com/company/legal/trademarks/tmlist.html>. All other third party trademarks are the property of their respective owners. A trademark symbol (®, # etc.) denotes a SUSE or Novell trademark; an asterisk (*) denotes a third party trademark.

All information found in this book has been compiled with utmost attention to detail. However, this does not guarantee complete accuracy. Neither SUSE LLC, its affiliates, the authors nor the translators shall be held liable for possible errors or the consequences thereof.

下記に上記の日本語翻訳を掲載します。日本語の翻訳は公式なものではないことに注意してください。

Copyright © 2006–2016 SUSE LLC および貢献者が全権利を留保しています。

この文書を、フリーソフトウェア財団発行の GNU フリー文書利用許諾契約書 バージョン 1.2 または (希望すれば) 1.3 が定める条件の下で複製、頒布、あるいは 改変することを許可する。ただし、この著作権とライセンス表記については変更不可部分 とする。この利用許諾契約書の複製物は「GNU フリー文書利用許諾契約書」という章に含まれている。

SUSE 社, Novell 社の商標については、Novell 社の商標とサービスマーク一覧 <http://www.novell.com/company/legal/trademarks/tmlist.html> をご覧ください。その他の商標は各所有者の所有物です。商標シンボル (®, # など) は、それぞれ SUSE 社および Novell 社の商標であることを示しています。また、アスタリスク (*) は第三者の商標を示しています。

この書籍内にある全ての情報は細部に至るまで最大限の注意を払って制作されていますが、完全に正確であることを保証するものではありません。SUSE LLC, SUSE LINUX Products GmbH, 著者, 翻訳者のいずれも、本書籍内の誤りとそこから生じる結果について、一切の保証はいたしません。

目次

このマニュアルについて	vii
1 利用可能な文書	vii
2 フィードバック	viii
3 文書規約	ix
4 このマニュアルの作成について	ix
5 ソースコード	x
6 謝辞	x

I システム要件と制限事項 1

1 KVM のインストールと要件 3

1.1 ハードウェア要件	3
1.2 対応済みゲスト側オペレーティングシステム	4
1.3 kvm パッケージ	8
1.4 KVM のインストール	9

2 KVM の制限事項 11

2.1 一般的な制限事項	11
2.2 ハードウェア面の制限事項	12
2.3 性能面の制限事項	13

II libvirt を利用した仮想マシンの管理	15
3 概要	17
4 ゲストのインストール	21
4.1 Virtual Machine Manager を利用したゲストのインストール	21
4.2 vm-install を使用したコマンドラインからのインストール	28
4.3 高度なゲストインストール方法	32
5 基本的な VM ゲスト の管理	35
5.1 VM ゲスト の一覧表示	35
5.2 グラフィカルコンソールの表示	36
5.3 VM ゲスト の状態変更: 起動, 停止, 一時停止	38
5.4 VM ゲスト の保存と復元	41
5.5 VM ゲスト の削除	42
6 接続と権限	45
6.1 認証	45
6.2 リモート接続の設定	55
6.3 VM ホストサーバ への接続	63
7 ストレージの管理	69
7.1 Virtual Machine Manager を利用したストレージ管理	72
7.2 virsh を利用したストレージの管理	78
7.3 virtlockd を利用したディスクファイルとブロックデバイスの施錠	82
8 仮想マシンの設定	87
8.1 シームレスなカーソル移動とカーソル移動の同期	88
8.2 Virtual Machine Manager を利用した CD/DVD-ROM デバイスの追加	89
8.3 Virtual Machine Manager を利用したフロッピーデバイスの追加	90

8.4 Virtual Machine Manager を利用したフロッピーディスクまたは CD/ DVD-ROM メディアの取り出しと交換	91
8.5 Virtual Machine Manager を利用した PCI デバイスの追加	92
8.6 virsh を利用した PCI デバイスの追加	93
8.7 時計の設定	96
9 VM ゲスト の管理	99
9.1 VM ゲスト の移行	99
9.2 監視	103
III QEMU を利用した仮想マシンの管理	107
10 QEMU の概要	109
11 ゲストのインストール	111
11.1 qemu-kvm を利用した基本インストール	112
11.2 qemu-img を利用したディスク管理	114
12 qemu-kvm を利用した仮想マシンの実行	127
12.1 基本的な qemu-kvm の実行	127
12.2 一般的な qemu-kvm オプション	128
12.3 QEMU における仮想デバイスの使用	132
12.4 QEMU でのネットワーキング	143
12.5 VNC を利用した VM ゲスト の閲覧	150
12.6 VirtFS: ホストとゲストの間でのフォルダ共有	154
12.7 KSM: ゲスト間でのメモリページ共有	155
13 QEMU モニタを利用した仮想マシンの管理	157
13.1 モニタコンソールへのアクセス	157
13.2 ゲストシステムについての情報取得	157
13.3 VNC パスワードの変更	160

13.4 デバイスの管理	161
13.5 キーボードとマウスの操作	161
13.6 利用可能なメモリ量の変更	162
13.7 仮想マシンのメモリダンプ	162
13.8 仮想マシンのスナップショット管理	163
13.9 仮想マシンの一時停止 (サスペンド) と復元	164
13.10 ライブマイグレーション	165

A 補足 167

A.1 擬似仮想化ドライバのインストール	167
A.2 x509 クライアント／サーバ証明書の生成	172

B GNU ライセンス 175

B.1 GNU General Public License	175
B.2 GNU 一般公衆利用許諾契約書 (日本語訳)	179
B.3 GNU Free Documentation License	183
B.4 GNU フリー文書利用許諾契約書	188

このマニュアルについて

このマニュアルでは、KVM (カーネルベースの仮想マシン) による仮想化を openSUSE 上で利用するにあたって、その設定方法や管理方法を紹介しています。最初のパートでは要件や SUSE におけるサポート状態を説明し、次のパートでは libvirt を利用した KVM の管理方法を、そして最後のパートでは QEMU を利用した管理方法をそれぞれ紹介しています。

このマニュアルの多くの章では、追加のドキュメンテーション資源へのリンクが書かれています。これらの資源には、システム内に含まれているもののほか、インターネットを介して提供されているものも含まれています。

お使いの製品に対してどのようなドキュメンテーションが提供されているのか、もしくは最新のドキュメンテーション更新について知るには、<http://doc.opensuse.org/> をご覧ください。

1 利用可能な文書

HTML 版や PDF 版の各マニュアルは、それぞれ各種の言語に翻訳されています。この製品に対しては、それぞれ下記に示す ユーザ向けおよび管理者向けマニュアルが用意されています:

スタートアップ (↑ スタートアップ)

DVD や ISO イメージから openSUSE のインストールを行ない、GNOME や KDE デスクトップの簡単な説明と、そこで動作する主なアプリケーションを紹介するまでの範囲を説明しています。また、LibreOffice の概要説明のほか、文書作成や表計算での作業、およびグラフィックやプレゼンテーションの作成を行うためのモジュールについても説明しています。

リファレンス (↑ リファレンス)

openSUSE に関する一般的な理解を深め、より詳しいシステム管理作業を行うための情報が書かれています。主にシステム管理者のほか、システム管理知識のあるホームユーザに向けた文書です。また、複雑な配置シナリオやシステムの管理方法、主なシステムコンポーネントとのやりとりや openSUSE が提供するネットワークサービス、ファイルサービスに関する詳しい情報も書かれています。

セキュリティガイド (↑セキュリティガイド)

ローカル環境やネットワークセキュリティを含めた、システムセキュリティに関する基本的な考え方が書かれています。AppArmor のようなセキュリティソフトウェア (プログラムが読み書きしたり実行したりするファイルをプログラム単位で指定できるもの) の一般的な使い方を示しているほか、セキュリティ関連のイベント情報を確実に収集するための監査システムの使い方も示しています。

システム分析とチューニングガイド (↑システム分析とチューニングガイド)

問題の検出や解決、最適化に対する管理者向けのガイドです。お使いのシステムに関して監視ツールを利用し点検と最適化を行なう方法や、効率的に資源を管理するための手順が記されています。また、一般的によくある問題やそれに対する解決方法、追加のヘルプや文書資源についても示しています。

KVM を利用した仮想化 (i ページ)

このマニュアルでは、openSUSE で KVM (カーネルベースの仮想マシン) による仮想化を設定したり、管理したりするための手順を紹介しています。また、libvirt や QEMU を利用した VM ゲストの管理方法についても紹介しています。

ほとんどの製品マニュアルは HTML 版の形で、インストール済みシステムの `/usr/share/doc/manual` に置かれています。またデスクトップのヘルプセンターからもアクセスすることができます。最新の文書は、<http://www.suse.com/documentation> に置いています。ここからお使いの製品について、PDF 版と HTML 版をダウンロードすることができます。

2 フィードバック

いくつかの方法でフィードバックを送ることができます:

バグや機能追加リクエスト

製品のコンポーネントに対してバグの報告を行なったり、もしくは機能の追加リクエストを送信したりしたい場合は、<https://bugzilla.novell.com/> をご利用ください。文書内の間違いについては、各製品の *[Documentation]* コンポーネントに対してバグ報告をお願いいたします。

Bugzilla を初めてお使いになる場合は、下記の記事をお読みください:

- http://ja.opensuse.org/Submitting_bug_reports
- http://ja.opensuse.org/Bug_reporting_FAQ

ユーザコメント

このマニュアルに対するコメントや提案のほか、この製品に含まれる他のドキュメント 類に対するコメントを歓迎します。オンラインドキュメントの場合は、それぞれの ページ下部にあるコメント機能をご利用いただくか、もしくは <http://www.suse.com/documentation/feedback.html> から コメントをお送りください。

メール

この製品に対するフィードバックを送信するには、`doc-team@suse.de` 宛のメールもお使いいただけます。それぞれドキュメントのタイトルと製品バージョン、発行日時を添えてお送りください。また、間違いの報告や加筆に対する提案につきましては、その簡潔な説明と、セクション番号およびページ (または URL) をお送りください。

3 文書規約

このマニュアルでは、下記のルールで文書を記述しています:

- `/etc/passwd`: ディレクトリ名やファイル名を示しています
- *placeholder*: 置き換えを示しています *placeholder* を実際の値に置き換えます
- `PATH`: `PATH` という名前の環境変数を示しています
- `ls, --help`: コマンドやオプション、パラメータ を示しています
- `user`: ユーザまたはグループ
- `Alt, Alt + F1`: 入力するキーやキーの組み合わせを示しています; キーはキーボードに書かれている とおりに大文字で示されます
- `[ファイル], [ファイル] > [名前を付けて保存]`: メニュー項目やボタンなどを示しています
- ダンシングペンギン (他のマニュアル内 ペンギン の章): 他のマニュアル内にある章を示しています

4 このマニュアルの作成について

この書籍は、DocBook (詳しくは <http://www.docbook.org> をご覧ください) のサブセットである Novdoc で書かれています。XML のソースファイルは `xmllint`

で検証された後に xsltproc で処理され、Norman Walsh 氏のスタイルシートのカスタマイズ版を利用して XSL-FO に変換されます。最終的な PDF ファイルは RenderX 提供の XEP で生成しています。また、この マニュアルを構築するために使用するオープンソースツールとその環境は、openSUSE と共に公開されている daps パッケージ内にあります。なお、daps の Web ページは <http://daps.sf.net/> です。

5 ソースコード

openSUSE のソースコードは、どなたにでもご利用いただけます。ダウンロードのリンクやその他の説明については、http://ja.opensuse.org/Source_code をお読みください。

6 謝辞

多数の無償貢献のお陰で、Linux 開発者はその開発にあたってグローバルな協力を行なうことができています。我々は彼らのそのような努力に感謝します— 彼らの貢献がなければ本ディストリビューションは存在していませんでした。また、Frank Zappa 氏と Pawar 氏にも感謝しています。もちろん Linus Torvalds 氏には特に感謝しています。

Have a lot of fun!

SUSE チームより

パート I. システム要件と制限事項

KVM のインストールと要件

KVM はハードウェア仮想化 (Intel VT または AMD-V) 機能のある x86 プロセッサ 向けに開発された完全仮想化ソリューションです。KVM は主に 2 種類の コンポーネントから構成されています。1 つは仮想化の中核基盤を提供し、プロセッサ 固有のドライバでもあるカーネルモジュールのセット (kvm.ko, kvm-intel.ko, kvm-amd.ko)、もう 1 つは仮想デバイス向けの エミュレーションを提供したり、VM ゲスト (仮想マシン) の管理機構を提供したり するユーザスペース側のプログラムです。KVM という用語は厳密にはカーネル レベルの仮想化機能のことを指しますが、 実際にはユーザスペース側のコンポーネント までを含めた総称として利用しています。

VM ゲスト (仮想マシン) と仮想ストレージ、仮想ネットワークはそれぞれ libvirt ベースのツールと QEMU ツールで管理することができます。libvirt は KVM や Xen など、異なる仮想化ソリューションをベースにした VM ゲスト について、これらの管理 API を提供するライブラリです。グラフィカル ユーザインターフェイスのほか、コマンドラインプログラムも提供しています。一方の QEMU ツールは KVM/QEMU 固有のプログラムで、コマンドライン インターフェイスのみを提供しています。

1.1 ハードウェア要件

現時点では SUSE は x86_64 ホスト上での KVM 完全仮想化のみをサポートしています。また、KVM は AMD (AMD-V) や Intel (VT-x) CPU に含まれる ハードウェア仮想化機能を利用するように設計されています。チップセットや PCI デバイスの仮想化機能も提供されていて、たとえば I/O メモリマッピング ユニット (IOMMU)

やシングルルート I/O 仮想化 (SR-IOV) などを利用することができるようになっています。

お使いの CPU がハードウェア仮想化機能に対応しているかどうか調べるには、下記のコマンドを入力してください:

```
egrep '(vmx|svm)' /proc/cpuinfo
```

上記のコマンドを入力しても何も出力されない場合は、お使いのプロセッサがハードウェア仮想化に対応していないか、もしくは BIOS でハードウェア仮想化機能が無効化されていることを示します。

上記以外にも、下記 Web サイトでハードウェア仮想化に対応するプロセッサを確認することができます: <http://ark.intel.com/Products/VirtualizationTechnology> (Intel CPU 向け) <http://products.amd.com/> (AMD CPU 向け)。

注記

KVM のカーネルモジュールは、お使いの CPU がハードウェア仮想化に対応していない場合や、BIOS でハードウェア仮想化機能を有効に設定していない場合には、読み込みが行なわれません。

VM ホストサーバを利用するにあたって、最小限となるハードウェア要件は openSUSE と同じです。に示されている概要と同じです。しかしながら、それぞれの仮想化ゲストに対して、少なくとも物理的な (仮想化を伴わない) インストールと同じだけの RAM を割り当ててください。また、それぞれの仮想化ゲストに対して、1 つ以上のプロセッサコアまたはハイパースレッドを割り当てておことを強くお勧めします。

1.2 対応済みゲスト側オペレーティングシステム

下記の表には、テスト済みのゲスト側オペレーティングシステムと、それらの SUSE におけるサポート状態を示しています。すべてのゲスト側オペレーティングシステムは完全仮想化と擬似仮想化の両方に対応していますが、Windows ゲストのみ、完全仮想化のみの対応となります。また、OES、Netware ゲストについては、擬似仮想化のみの対応となります。さらに、Netware は 32 ビットのみの対応となりますが、それ以外のすべてのオペレーティングシステムは 32 ビットと 64 ビットの両方の x86 版に対応しています。その他のオペレーティングシステムに対するサポー

ト状態 (SUSE ではテスト していないもの) について、詳しくは http://www.linux-kvm.org/page/Guest_Support_Status をお読みください。

また、擬似仮想化ドライバ (PV ドライバ) についても、利用可能であればその旨を示しています。

KVM 向けの擬似仮想化ドライバ

- virtio-net: 仮想ネットワークドライバ。
- virtio-blk: 擬似仮想化ブロックデバイス向けの仮想 ブロックデバイスドライバ。
- virtio-balloon: 動的なメモリ割り当てのための メモリドライバ。ゲストに対して 割り当てるメモリを、動的に変更することが できるようになります。
- virtio-scsi: 高度な SCSI ハードウェアに対応する ストレージインターフェイス。
- kvm-clock: 時刻同期ドライバ。

表 1.1 openSUSE における KVM 対応済み KVM ゲスト側オペレーティングシステム

openSUSE 11.2 またはそれ以降	
PVドライバ:	kvm-clock, virtio-net, virtio-blk, virtio-balloon
サポート状態:	完全サポート
SLES 11 SP1/ SP2 / SP3	
PVドライバ:	kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console, virtio-rng
サポート状態:	完全サポート
SLES 10 SP4	
PVドライバ:	kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console
サポート状態:	完全サポート

SLES 9 SP4

<i>PVドライバ:</i>	提供されていません
<i>サポート状態:</i>	完全サポート
<i>必須起動パラメータ:</i>	32 ビット版カーネルの場合: clock=pmtmr 64 ビット版カーネルの場合: ignore_lost_ticks

SLED 11 SP1 / SP2 / SP3

<i>PVドライバ:</i>	kvm-clock, virtio-net, virtio-blk, virtio-balloon, virtio-console, virtio-rng
<i>サポート状態:</i>	技術プレビュー状態

RedHat Enterprise Linux 4.x / 5.x / 6.x

<i>PVドライバ:</i>	http://www.redhat.com/ をお読みください。
<i>サポート状態:</i>	現状ありのまま
<i>注意:</i>	詳しくは RHEL 仮想化ガイドをお読みください。

Windows 2003 SP2+ / 2008 SP2+ / 2008 R2+ / 2012

<i>PVドライバ:</i>	仮想マシンドライバパック (http://www.suse.com/products/vmdriverpack/) 内にある virtio-net, virtio-blk, virtio-balloon の各ドライバの使用を推奨します
<i>サポート状態:</i>	Server Virtualization Validation Program (サーバ仮想化検証プログラム; SVVP) による完全サポート (完全な仮想化に対応)
<i>注意:</i>	ホスト側のプロセッサが、constant_tsc と呼ばれる CPU 機能に対応していなければなりません (下記

のコマンドで確認できます: `grep "constant_tsc" /proc/cpuinfo`

Windows XP SP3+ / 2003 SP2+ / Vista SP2+ / 7 SP1+ / 8

<i>PVドライバ:</i>	仮想マシンドライバパック (http://www.suse.com/products/vmdriverpack/) 内にある virtio-net, virtio-blk, virtio-balloon の各ドライバの使用を推奨します
<i>サポート状態:</i>	現状ありのまま

重要

SUSE Linux Enterprise Server 11 SP1 で作成されたゲストイメージには対応していますが、それ以前の SUSE Linux Enterprise で作成されたものには対応していません。

1.2.1 擬似仮想化ドライバの利用できる環境

仮想化を利用する際、ゲスト側のオペレーティングシステムの性能を向上させる目的で、擬似仮想化ドライバが提供されている場合があります。これらは必須というわけではありませんが、ご利用になることを強くお勧めします。擬似仮想化ドライバは下記のとおり提供されています:

SUSE Linux Enterprise Server 11 SP1 / SP2 / SP3
カーネル内に同梱

SUSE Linux Enterprise Server 10 SP4
カーネルに同梱

SUSE Linux Enterprise Server 9 SP4
用意されていません

RedHat
RedHat Enterprise Linux 5.4 またはそれ以降のバージョンであれば 利用できます

Windows

SUSE では virtio ベースの Windows 向けドライバを開発していて、これらは Virtual Machine Driver Pack (仮想マシンドライバパック; VMDP) として利用できます。詳しくは <http://www.suse.com/products/vmdriverpack/> をお読みください。

1.3 kvm パッケージ

kvm パッケージには、VM ゲスト 向けに I/O エミュレーションを行なうことのできるプログラム qemu-kvm が提供されています。qemu-kvm プログラムに加え、kvm パッケージには デバッグレベルの監視ユーティリティ (kvm_stat) や ファームウェアコンポーネント、キーマッピングファイルやスクリプトが含まれています。そのほか、古い擬似仮想化向け Windows ドライバ (/usr/share/qemu-kvm/win-virtio-drivers.iso) が含まれています。

元々は kvm パッケージで KVM のカーネルモジュールも提供していましたが、これらのモジュールはカーネル内に 同梱されるようになったため、kvm パッケージでは ユーザスペース側のコンポーネントのみを提供するようになっています。

また、VM ゲスト を管理するにあたっては libvirt ベースのツールを使用するのがお勧めです。これは、他の仮想化ツールとの相互運用性がテストされていて、こちらのツールを使用することが SUSE でのサポートからも推奨されている ためです。各種のツールは、そのツールと同名のパッケージで提供されています。

- libvirt: VM ゲスト や仮想ネットワーク、仮想ストレージなどを管理する ツールキット。libvirt は API のほか、デーモンやシェル (virsh) を提供しています。
- virt-manager (Virtual Machine Manager): VM ゲスト 向けのグラフィカルな管理ツールです。
- vm-install: VM ゲスト を設定し、オペレーティング システムをインストールするツールです。
- virt-viewer: VM ゲスト 向けの X ビューアクライアント です。x509 証明書による TLS/SSL 暗号化に対応するほか、SASL 認証にも 対応しています。

ファイルベースのディスクイメージを作成したり取り扱ったりするには、qemu-img をお使いください。qemu-img は virt-utils パッケージで 提供されています。

1.4 KVM のインストール

KVM は既定ではインストールされていません。KVM とすべての仮想化ツールをインストールするには、下記の手順で作業を行なってください:

- 1 YaST を起動し、[仮想化] > [ハイパーバイザとツールのインストール] を選択します。
- 2 [KVM] を選択して [了解] を押します。
- 3 インストールされるパッケージが表示されますので、内容を確認して [インストール] を押します。
- 4 次に [はい] を押してネットワークブリッジの設定を行ないます。ブリッジ設定は VM ホストサーバ (仮想マシンのホスト) 上で推奨される設定です。異なるネットワーク設定を手作業で行ないたい場合は、[いいえ] を押して手順を飛ばすことができます。
- 5 設定作業が完了すると、YaST はマシンの再起動を求めます。再起動を行なう代わりに、カーネルモジュールを手作業で読み込んで libvirtd を起動することもできます:

```
modprobe kvm-intel # on Intel machines only
modprobe kvm-amd   # AMD マシンの場合
modprobe vhost-net
rclibvirtd start
```

注記: vhost-net カーネルモジュールについて

vhost-net カーネルモジュールは、ゲストに対してより効率的なネットワーク転送機能を提供します。これは qemu-kvm を読み込んだり利用したりした際に、ネットワークオプションに、vhost=on を追記すれば libvirt で自動的に使用します。

KVM の制限事項

仮想化されたマシンは物理的な (仮想化をしていない) マシンのように振る舞いますが、いくつかの制限事項があります。これらは VM ホストサーバ 側のシステムに当てはまるものがあるほか、VM ゲスト 側にも当てはまるものがあります。

2.1 一般的な制限事項

KVM を利用するにあたって、一般的な制限事項は下記の通りです:

オーバーコミット

KVM ではメモリとディスク領域の両方に対してオーバーコミットを許可しています。これが何を意味するのかは利用形態によって変わりますが、利用可能なリソースを超過することによって発生するハードウェアエラーは、ゲスト側のエラーにもつながります。また CPU のオーバーコミットにも対応していますが、性能面で問題を引き起こす可能性があります。

時刻同期

多くのゲストでは、時刻を正確に維持するのに追加のサポートを必要とします。利用可能であれば、`kvm-clock` を使用してください。それ以外にも NTP やその他のネットワークベースの時刻同期プロトコルを利用し、安定した時刻を維持することを強くお勧めします (VM ホストサーバと VM ゲスト の 両方に対して)。ゲスト内での NTP は、`kvm-clock` を利用している場合には推奨されません。詳しくは 8.7 項「時計の設定」(96 ページ) をお読みください。

MAC アドレス

NIC に対して MAC アドレスを指定しない場合は、既定の MAC アドレスが割り当てられます。この場合、複数の NIC が同じ MAC アドレスになってしまい、ネットワーク問題を引き起こすことが考えられます。それぞれの NIC に対してユニークな MAC アドレスを割り当てることをお勧めします。

ライブマイグレーション

ライブマイグレーション機能は同じ機能の CPU を利用した VM ホストサーバ間のみで実現することができます。また、マイグレーションに対応する CPU モデルは、`-cpu qemu64` (既定値) のみであり、他の追加機能が有効化されていない場合のみであることにも注意が必要です。さらに、物理デバイスをホストからゲストに渡すこともできません。ゲスト側のストレージは両方の VM ホストサーバからアクセスできない限り、ゲストの定義がそれぞれ互換性のあるものでなければなりません。また、VM ホストサーバと VM ゲストの間では、適切な時刻維持機能が必要です。AHCI インターフェイス (`virtfs` インターフェイス) を使用する場合や、`-mem-path` コマンドラインオプションもマイグレーションには不適切です。SP3 から SP2 もしくは SP1 でホスティングされた環境にも移行することはできません。

ユーザのアクセス許可

管理ツール (Virtual Machine Manager, `virsh`, `vm-install`) は `libvirt` に対して認証を行なう必要があります (詳しくは 第6章 [接続と権限](#) (45 ページ) をお読みください)。また、`qemu-kvm` をコマンドラインから実行するには、ユーザが `kvm` のメンバーでなければなりません。

VM ホストサーバのサスペンド／ハイバネート

ゲストが稼働中の場合、VM ホストサーバシステムのサスペンドやハイバネートには対応していません。

2.2 ハードウェア面の制限事項

ゲストに対する仮想的なハードウェア制限が確認されています。ただし、以下のよう
な制限に到達しても、最新のリリース以降であれば、ホストと VM のインストールと
動作の問題はありませんし、大幅な性能劣化 (CPU, メモリ, ディスク, ネットワーク)
もありません。

ゲスト側の最大 RAM サイズ	512 GB
-----------------	--------

ゲスト 1 つあたりの最大仮想 CPU 数	64
ゲスト 1 つあたりの最大仮想ネットワークデバイス数	8
ゲスト 1 つあたりの最大ブロックデバイス数	エミュレーションで 4 個 (IDE)、擬似仮想化で 20 個 (virtio-blk を使用) もしくは 100 個 (virtio-scsi を使用)
VM ホストサーバ 1 台あたりの最大 VM ゲスト 数	全てのゲストに割り当てた仮想 CPU 数の合計が、ホスト側の CPU コア数の 8 倍を超えない範囲

2.3 性能面の制限事項

基本的には物理的な (仮想化を伴わない) インストール向けの作業はそのまま 仮想化することができるため、新しい仮想化技術の利点はそのまま享受できる ことになります。しかしながら、仮想化を利用することで、それなりに性能に 影響することがあります。そのため、CPU や I/O に対して考えられる最大限の 仕事量を与えて、仮想化に耐えられるかどうかをお確かめください。様々な 要件に適合するかどうか、広範囲な仮想化ソリューションを利用して確認は 行なわれていますが、場合によっては KVM での仮想化に適さない作業である ことがありうるためです。

このことから、下記の通りガイドラインとしてゲスト側の性能予想を示しています。下記の表でのパーセント値は、非仮想化時と同じ仕事量を与えた場合に どれだけの性能を達成できるのかを示した値です。なお、下記の値はおおよその 値であり、性能が保証されるものではありません。

分類	完全仮想化	擬似仮想化	ホスト側でのパススルー
CPU, MMU	7%	該当せず	97% (拡張ページテーブル (Intel) またはネステッドページテーブル

分類	完全仮想化	擬似仮想化	ホスト側でのパススルー
			(AMD) を利用したハードウェア仮想化の場合) 85% (シャドウページテーブルを利用したハードウェア仮想化の場合)
ネットワーク I/O (1GB LAN)	60% (e1000 エミュレーションの NIC の場合)	75% (virtio-net)	95%
ディスク I/O	40% (IDE エミュレーションの場合)	85% (virtio-blk)	95%
Graphics (アクセラレーション無効の場合)	50% (VGA または Cirrus)	該当せず	該当せず
時刻の正確さ (NTP を利用しない推奨設定の場合の最悪ケース)	95% - 105% (100% を正確とした値)	100% (kvm-clock)	該当せず

パート II. libvirt を利 用した仮想マシンの管理

概要

libvirt は KVM や Xen などの有名な仮想化ソリューションを管理するための、汎用 API を提供するライブラリです。本ライブラリはこれらの仮想化ソリューションに対して標準化された管理 API を提供しているため、安定した形でハイパーバイザにも依存しない、高レベルな管理ツールを提供することができます。このライブラリには、VM ホストサーバ上での仮想ネットワークや仮想ストレージを管理する API も含まれています。なお、それぞれの VM ゲストに関する設定は XML ファイル内に保存されます。

libvirt を利用することで、VM ゲストをネットワーク上離れた場所 (リモート) から管理することもできます。それ以外にも、TLS による暗号化や x509 証明書、SASL による認証などにも対応しています。

仮想化ソリューションと libvirt 間の通信は libvirtd と呼ばれるデーモンで管理されます。このツールは管理ツールとしても使用するものです。libvirtd は VM ホストサーバ上のほか、libvirtd ベースのツールを動作させるリモートのマシン上でも動作させる必要があります。それぞれ下記のコマンドで起動や停止、状態の確認などを行なうことができます:

```
~ # rclibvirtd start
Starting libvirtd                                done
~ # rclibvirtd status
Checking status of libvirtd                      running
~ # rclibvirtd stop
Shutting down libvirtd                          done
~ # rclibvirtd status
Checking status of libvirtd                      unused
```

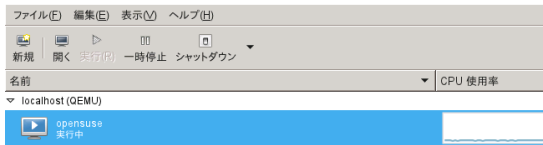
システムの起動時に libvirtd を自動で開始するには、YaST の [システムサービス (ランレベル)] を利用して設定するか、もしくは下記のコマンドを実行します:

```
insserv libvirtd
```

openSUSE では、下記の libvirt ベースのツールが用意されています：

Virtual Machine Manager (virt-manager)

Virtual Machine Manager は VM ゲスト を管理するためのデスクトップツールです。このツールは 既存のマシンのライフサイクル (起動やシャットダウン、一時停止や再開、サスペンドや復元) を制御する機能を持っています。また、新しい VM ゲスト を作成することができるほか、様々な種類のストレージを作成したり、仮想 ネットワークを管理したりすることもできます。それ以外にも、内蔵の VNC ビューアを利用して VM ゲスト のグラフィカルコンソールにアクセスすることができるほか、性能の統計情報を表示したりすることもできます。いずれの機能とも、ローカルとリモートの両方で動作させることができます。



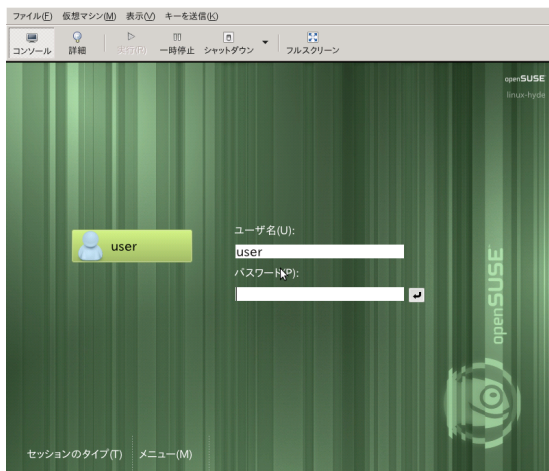
Virtual Machine Manager は VM ホストサーバ 上で実行する必要はありません。リモートの接続を利用して VM ゲスト を制御することができます。これにより、それぞれの VM ホストサーバ に対して わざわざログインを行なうことなく、単一のワークステーションから VM ゲスト を集中管理することができます。

Virtual Machine Manager を起動するには、コマンドプロンプトから `virt-manager` と入力します。

virt-viewer

VM ゲスト のグラフィカルなコンソールを表示するビューアです。VNC プロトコル を 利用した仕組みで、TLS と x509 証明書に対応しています。VM ゲスト の指定は 名前や ID 、UUID で指定できます。ゲストがその時点で動作してい

ない場合は、ビューアはコンソールに接続しようとする前に、ゲストが起動するまで待機を するようにすることができます。



vm-install

VM ゲスト をセットアップするためのツールで、デバイスの設定やオペレーティング システムのインストールを開始することができます。グラフィカルなユーザ インターフェイスから呼び出された場合は GUI ウィザードが起動します。端末から 起動した場合は、コマンドラインモードでウィザードが起動します。なお、vm-install は Virtual Machine Manager で新しい仮想マシンを作成した場合 にも 起動します。

virsh

Virtual Machine Manager と似通った機能を持つ VM ゲスト の管理用コマンドラインツールです。VM ゲスト の状態 (起動、停止、一時停止など) を変化させ、新しいゲストの 設定を行ったり、既存の設定を編集したりすることができます。virsh は VM ゲスト の管理操作をスクリプトから行なう ような場合に 便利なツールです。

virsh は基本的に Subversion の svn コマンドや zypper コマンドのように動作します。つまり、最初のパラメータとしてコマンドを、それ以降のパラメータとして そのコマンドに対するさらなるオプションを指定します:

```
virsh [-c URI] command domain-id [OPTIONS]
```

zypper と同様、virsh はコマンド無し でも起動することができます。この場合、コマンド入力を待つシェルとして起動 することができます。このモードは、一連の コマンドを実行したい場合に 便利です:

```
~> virsh -c qemu+ssh://wilber@mercury.example.com/system  
Enter passphrase for key '/home/wilber/.ssh/id_rsa':  
Welcome to virsh, the virtualization interactive terminal.
```

```
Type: 'help' for help with commands  
      'quit' to quit
```

```
virsh # hostname  
mercury.example.com
```

ゲストのインストール

VM ゲスト はオペレーティングシステムを含むイメージファイルとデータファイル、そして VM ゲスト の仮想ハードウェア資源を定義する設定ファイルから構成されています。VM ゲスト は VM ホストサーバ 上で動作し、制御されます。

4.1 Virtual Machine Manager を利用したゲストのインストール

Virtual Machine Manager 内で **[新規]** を押すと、`vm-install` を起動することができます。このプログラムはグラフィカルな **[仮想マシンの作成ウィザード]** を提供し、ゲストのインストール 作業をガイドします。なお、`vm-install` はコマンドラインから 直接起動することができるほか、YaST から **[仮想化]** > **[仮想マシンの作成]** を選択することでも起動する ことができます。

- 1 まずは上述の通りに **[仮想マシンの作成ウィザード]** を起動し、**[進む]** を押します。
- 2 オペレーティングシステムをインストールするか、もしくはすでに存在する イメージやディスクからインストールを行なうかを選択します。
- 3 一覧からインストールを行ないたいオペレーティングシステムを選択します。それぞれの項目を選択すると、選択したオペレーティングシステム向けの 既定値が設定されます。
- 4 **[概要]** ページでは、選択したオペレーティングシステムでの 既定の設定が表示されます。それぞれのヘッダラインを選択することで、設定を 編集することができます。

まず、システムのインストールを選択すると、イメージの 指定や起動に使用する CD/DVD デバイスの選択、もしくは PXE 起動のいずれかを 選択することになります。[OK] を押して設定を完了すると、ゲストシステムのインストール作業が始まります。

4.1.1 既定の設定に対するカスタマイズ

仮想マシンの作成ウィザードでは、概要ページからヘッドラインを押すことで、各種の設定を行なうことができます：



概要

変更する場合は、ヘッドラインを押してください。設定が正しい場合は、OK を押して仮想マシンを作成してください。

仮想マシン名
opensuse11

ハードウェア
初期メモリ: 512 MB
最大メモリ: 512 MB
仮想プロセッサ数: 1

グラフィックとキーボード
Cirrus Logic GD5446 VGA

ディスク
1: 8.0 GB ハードディスク (file:/var/lib/kvm/images/opensuse11/disk0.raw)

ネットワークアダプタ
1: QEMU Virtualized NIC Card; ランダムに生成された MAC アドレス

OS インストール
オペレーティングシステム: openSUSE 11
インストール元:

キャンセル(C) 戻る(B) OK(O)

4.1.1.1 [仮想マシンの名前]

ゲストに対してそれぞれ [名前] を設定します。また、必要であれば [説明] を入力することもできます。[名前] 欄には半角の英数字と _ . : + の各記号を利用することができます。この項目は VM ホストサーバ 内で唯一のものでなければなりません。この名前はゲストの 設定ファイルを作成する際に使用されるもので、virsh から ゲストにアクセスする際にも利用します。

4.1.1.2 [ハードウェア]

この画面ではメモリと CPU の割り当てを変更することができます。なお、VM ホストサーバ が提供可能な量よりも大きな値を設定する (オーバーコミット) ことは お勧めできません。これはエラーや性能面の問題が発生する可能性があるためです。

このダイアログでは、PCI デバイス (たとえばネットワークカードなど) を割り当てて、VM ゲスト で直接使用するように設定することもできます (PCI パススルーと 呼び ます) 。[*Host Devices*] > [*Manage VM Devices*] を選択して、まずは利用可 能なデバイスの一覧を 表示させてください。その後、一覧からデバイスを選択して [*Add*] を押すと、VM ゲスト のデバイス一覧に追加することができます。この場合、 デバイス に対して [*Managed*] オプションを有効にしておくことをお勧め します。こ れは、libvirt に対してドライバとのやりとりを管理させるための オプションです。 詳しくは 'managed' と 'unmanaged' の違いについて (94 ページ) をお読みく ださい。

重要: PCI デバイスの共有について

PCI デバイスは、ホストと VM ゲスト の間でも、VM ゲスト 同士の間でも、共有 することはできません。各デバイスは単一のインスタンス内でのみ使用することが できます。設定の際は、PCI デバイスが他の箇所で使われていないかどうか、ご 確認ください。

[*高度な設定*] では、ACPI や APIC, PAE などそれぞれ有効／ 無効にすること ができます。ここでの設定を既定値から変更することはお勧め できません。また、 virtio を利用した擬似仮想化 I/O の有効化／無効化や、起動時にカーネルを実 行するように設定 (Linux のみ) することもできます。

重要: 擬似仮想化 I/O

[*virtio*] を有効に設定して擬似仮想化 I/O を利用した場合、作成した すべての ハードディスクが virtio のディスクとして 設定されます。これにより、お使いのオ ペレーティングシステムに適切なドライバが 存在しない場合は、インストールが失 敗することになります。Windows オペレーティングシステムの場合は、ドライバが 提供されていてもエラーに なります。既定ではこの機能は、オペレーティングシス テム側に virtio ドライバが存在していることがわかっている 場合にのみ有効に なります。

4.1.1.3 [周辺デバイス]

このダイアログでは、仮想化に使用するグラフィックハードウェアとキーマップ、およびサウンドデバイスについて設定を行ないます。グラフィックカードのサポートを無効にした場合、マシンはネットワークサービス (ssh) やシリアルポートからしか アクセスできなくなります。また、VM ゲスト におけるサウンドデバイスについては、SUSE ではサポートしていません。そのため、[Sound] の設定は None (無し) に設定しておいてください。

4.1.1.4 [ディスク]

[ディスク]: このダイアログでは、仮想ハードディスクと 仮想 CD/DVD ドライブを管理します。なお VM ゲスト には少なくとも 1 台の 仮想ディスクが必要です。既存のものを選択することができるほか、新しく 作成したディスクを設定することもできます。仮想ディスクは、それぞれ以下の タイプから選択することができます:

- 固定サイズの単一ファイル
- 必要に応じてサイズを拡張する単一ファイル (スパースイメージ ファイル)

重要: スパース (まばらな) イメージファイルを作成する場合のディスク領域について

スパース (実際に使用しているディスク領域よりもファイルサイズのほうが 大きい、ファイル内に実データが点在するファイル) イメージファイルを 作成する場合、それらを作成するパーティションには十分な空き容量が必要です。VM ゲスト は VM ホストサーバ のディスク領域を調べるようなことは しませんので、ホスト側のパーティションでディスク領域が不足すると、ゲストシステム側では書き込みエラーが発生し、データの損失が発生することになります。

- ディスク全体やパーティション、ネットワークボリュームを指し示す ブロックデバイス

最大限の性能を引き出したい場合は、作成する各仮想ディスクはディスク全体を使用するか、もしくはパーティション全体を使用してください。これらの作成が 難しい場合は、スパースイメージファイルを使用しない固定サイズのファイルを作成してください。スパースイメージファイルを使用した仮想ディスクの場合、ディスク領域の柔軟性を増すことができますが、インストールやディスクの アクセス速度が遅くなります。

ヒント: ライブマイグレーション

シャットダウンを行なうことなく、お使いの VM ゲスト を他のホストに移行 (ライブマイグレーション) したい場合は、両方のホストからアクセスできるようにするため、すべてのディスクをネットワーク上 (ネットワークファイル システム、または iSCSI ボリューム) に存在させなければなりません。

既定では、単一のスパース形式の raw ディスクイメージファイルを、`/var/lib/kvm/images/VM 名/` 内に作成します。ここで *VM 名* には、仮想マシンの 名前が入ります。

注記: サポートするディスク形式

現時点では SUSE は、raw, qed, qcow2 の各形式のみをサポートします。

手順 4.1 仮想ディスクの作成

- 1 [ハードディスク] を押します。
- 2 [ソース] に入力します。ファイルベースのディスクを作成 したい場合は、パスを直接入力するか、もしくは **[新規]** を押します。デバイスからディスクを作成したい場合は、デバイスノードを 入力します。たとえば `/dev/disk/by-path/path` のようになります。なお、`/dev/sdb` や `/dev/sda5` のように、デバイスパスを利用した名前は 指定しないことを強くお勧めします。これは、これらの名前が今後変わってしまう (ディスクの追加や BIOS 内でのディスク順序変更などにより) 可能性があるためです。
- 3 次に **[プロトコル]** を指定し、raw ディスクを作成する場合は、ファイルベースの場合 **[file]** を、デバイスベースの場合 **[phy]** を選択します。これ以外にも、qcow2 や qed ディスクも選択することができます。
- 4 次に **[サイズ]** を GB 単位で入力します。このオプションは、ファイルベースのディスクを作成する場合にのみ表示されます。
- 5 **[スパースイメージファイル]** を作成するかどうかを選択 します。このオプションは、ファイルベースのディスクを作成する場合にのみ 表示されます。また、ディスクへの書き込みアクセスを禁止したい場合は、**[読み込み専用アクセス]** を選択します。

DVD や CD-ROM からインストールを行ないたい場合は、利用可能なハードディスクの 一覧内にドライブを追加します。利用可能な光学デバイスのデバイスノードを知りたい 場合は、下記のコマンドを実行します:

```
hwinfo --cdrom | egrep "(Device File:|Model:)"
```

実際の (物理的な) DVD や CD-ROM ドライブを指定する代わりに、インストールメディアの ISO イメージを指定することもできます。ただし、CD-ROM ドライブと ISO イメージは、1 つのゲストに対して同時にいずれか 1 つのみを使用できます。

CD/DVD-ROM デバイスや ISO イメージを追加するには、下記の手順で行ないます:

- 1 [CD-ROM] を押します。
- 2 [ソース] に値を入力します。デバイスを追加するには、その ノードを指定します。ISO イメージを追加するには、パスを直接指定するか、もしくは [参照] を押してファイルブラウザからファイルを 指定してください。
- 3 続いて [プロトコル] を指定します。ISO イメージの場合は [ファイル] を、デバイスの場合は [物理] を選んでください。

ディスクは作成された順序で一覧表示されます。この順序は起動時の順序を示すものでもありますので、必要であれば [上へ] と [下へ] のボタンを利用して順序を入れ替えてください。

4.1.1.5 [ネットワークアダプタ]

既定では仮想マシンに対して、単一の仮想ネットワークカードが作成されます。MAC アドレスは乱数を利用して設定されますが、要件にあわせて後から修正することもできます。また、VM ホストサーバ上にブリッジが存在する場合、仮想ネットワーク カードはそのブリッジに接続されますが、存在しない場合は libvirt の既定の仮想ブリッジ (virbr1) に接続されます。

ネットワークアダプタを追加したり編集したりするには、下記の手順で行ないます:

- 1 カードを追加するには [新規] を、選択したカードの設定を 編集するには [編集] をそれぞれ押します。
- 2 ドロップダウンリストから [種類] を選択します。

注記: サポートする仮想ネットワークアダプタの種類

現時点では、Novell は [完全仮想化での Realtek 8139] と [完全仮想化での Intel e1000] のほか、擬似仮想化での [QEMU 仮想化 NIC カード] (virtio) をサポートしています。

- 3 ドロップダウンリストから [ソース] を選択します。
- 4 ランダムに生成する MAC アドレスを割り当てるか、もしくは指定した MAC アドレスを割り当てるかを選択します。

注記: MAC アドレスの唯一性について

手作業で MAC アドレスを指定する場合は、指定する MAC アドレスがお使いの ネットワーク内に存在していないことを確認してください。重複する MAC アドレスを利用してしまうと、特に DHCP 使用時などに問題が発生します。そのため、各インターフェイスには固定値で 52:54:00:12:34:56 や 52:54:00:11:22:33 のように MAC アドレスを指定せず、ランダムに生成された MAC アドレスを設定することを強くお勧めします。

4.1.1.6 [オペレーティングシステムのインストール]

このダイアログは、オペレーティングシステムをインストールするように選択した場合にのみ表示されます。インストールは仮想ディスクから起動して行なうことができるほか、CD/DVD デバイスや ISO イメージ、およびネットワーク リソースやネットワーク経由の PXE ブートで起動することができます。この ダイアログでは、起動デバイスの設定を行ないます。

また、このダイアログではオペレーティングシステムが電源を切ったり再起動したり、クラッシュしたりした場合の VM ゲスト の動作を設定することもできます。下記のオプションから選択できます。

[*destroy*]

通常のクリーンアップ処理を行ないます

[*restart*]

新しい VM ゲスト を起動します

[*preserve*]

クリーンアップを行なわず、一時データの削除や設定／イメージファイルの 削除などを行ないません

[*rename-restart*]

VM ゲスト はクリーンアップされませんが、新しい VM ゲスト を別名で 起動します

[*coredump-destroy*]

通常のクリーンアップ処理の前に、マシンのクラッシュコアダンプを 採取します

[*coredump-restart*]

通常の再起動処理が行なわれる前に、マシンのクラッシュコアダンプを 採取します

4.2 vm-install を使用したコマンドラインからのインストール

\$DISPLAY が設定されていない場合 (たとえばコンソールで作業を行なっている場合や、X フォワード機能を利用しない ssh シェルを利用しているような場合) は、`vm-install` は VM ゲスト をインストール用に 設定するための、対話型コマンドラインウィザードを提供します。設定作業が 完了すれば、VNC 経由で接続のできるインストールシステムを開始することが できるようになります。

重要: インストール時のグラフィカルユーザインターフェイスについて

インストール用に VM ゲスト を起動したあとは、インストールを続行するため、VNC を利用したグラフィカルコンソールに接続する必要があります。そのため、VNC ビューアを GUI 環境から起動する必要があります。

グラフィカルな環境にアクセスできる環境をお持ちでなく、テキストコンソールなどで作業を行なっている場合は、いったん VM ゲスト の設定を行なったあと、グラフィカルな環境にアクセスできるようになってからインストールを行なうようにすることができます。詳しい手順については、4.2.1 項「インストールを起動しない形での VM ゲスト の設定」(32 ページ) をお読みください。

ウィザードを起動するには、コマンドラインから `vm-install` と入力します。インストールウィザードでは、多数のパラメータに対して既定値が 設定されています。それぞれ Enter を押して確認を行なってください。下記に SUSE Linux Enterprise Server 11 のインストールを行なうにあたっての 対話セットアップログを記載します:

例 4.1 `vm-install` を使用したコマンドラインからの対話型セットアップ

```
~ # vm-install
Gathering settings...
```

```
Please specify the type of operating system that will run within the virtual
machine. This defines many defaults, and helps decide how to start
```

paravirtualized operating systems.

Press 'q' or the Escape key to exit.

- 1: Novell Open Enterprise Server 2 (Linux)
- 2: Novell Open Enterprise Server 2 (NetWare)
- 3: Other operating system
- 4: PXE
- 5: RedHat (other)
- 6: RedHat Enterprise Linux 3
- 7: RedHat Enterprise Linux 4
- 8: RedHat Enterprise Linux 5
- 9: SUSE (other)
- 10: SUSE Linux Enterprise Desktop 10
- 11: SUSE Linux Enterprise Desktop 11
- 12: SUSE Linux Enterprise Server 8
- 13: SUSE Linux Enterprise Server 9
- 14: SUSE Linux Enterprise Server 10
- 15: SUSE Linux Enterprise Server 11
- 16: Solaris 9 and older
- 17: Solaris 10
- 18: Windows (other)
- 19: Windows (other, x64)
- 20: Windows NT
- 21: Windows Server 2008
- 22: Windows Server 2008 (x64)
- 23: Windows Vista, Windows 7
- 24: Windows Vista, Windows 7 (x64)
- 25: Windows XP, 2000, 2003
- 26: Windows XP, 2003 (x64)
- 27: openSUSE
- 28: openSUSE 11

[15] >

PXE Boot

(Y / N) [N] >

Please choose a name for the virtual machine.

[sles11] >

Description > SLES 11 SP1

Specify the amount of memory and number of processors to allocate for the VM.

Initial Memory [512] >

Maximum Memory [512] > 768

Warning: Setting the maximum memory greater than the initial memory requires the VM operating system to have a memory balloon driver.

Virtual Processors [2] >

Please specify the type of virtualized graphics hardware.

- 1: Cirrus Logic GD5446 VGA
- 2: No Graphics Support
- 3: VESA VGA

[1] >

Virtual Disks:

```

    (None)
Do you want to add another virtual disk?
(Y / N) [Y] >

Create a virtual disk based on a device (CD or other block device), an existing
image file (ISO), or a new file. Specify a device by its device node, such as
/dev/cdrom, not its mount point.
What type of virtual disk do you want to add?
    1: CD-ROM or DVD
    2: Floppy
    3: Hard Disk
[3] > 3
Where will the virtual disk physically reside?
[/var/lib/kvm/images/sles11/hda] >
Size (GB) [4.0] > 8.0
Create a sparse image file for the virtual disk?
(Y / N) [Y] >

Virtual Disks:
    8.0 GB Hard Disk (file:/var/lib/kvm/images/sles11/hda)
Do you want to add another virtual disk?
(Y / N) [N] > y

Create a virtual disk based on a device (CD or other block device), an existing
image file (ISO), or a new file. Specify a device by its device node, such as
/dev/cdrom, not its mount point.
What type of virtual disk do you want to add?
    1: CD-ROM or DVD
    2: Floppy
    3: Hard Disk
[3] > 1
Where will the virtual disk physically reside?
[/var/lib/kvm/images/sles11/hdb] > /isos/SLES-11-SP1-CD-i386-GM-CD1.iso

Virtual Disks:
    8.0 GB Hard Disk (file:/var/lib/kvm/images/sles11/hda)
    2.9 GB CD-ROM or DVD (file:/isos/SLES-11-SP1-DVD-x86_64-GM-DVD1.iso)
Do you want to add another virtual disk?
(Y / N) [N] >

Network Adapters
    (None)
Do you want to add another virtual network adapter?
(Y / N) [Y] >
What type of virtual network adapter do you want to add?
    1: Fully Virtualized AMD PCnet 32
    2: Fully Virtualized Intel e100
    3: Fully Virtualized Intel e1000
    4: Fully Virtualized NE2000 (ISA Bus)
    5: Fully Virtualized NE2000 (PCI Bus)
    6: Fully Virtualized Realtek 8139
    7: Paravirtualized
[6] > 7

```



```
Network Adapters
  Paravirtualized; Randomly generated MAC address
Do you want to add another virtual network adapter?
(Y / N) [N] >
```

Preparing to start the installation...

Installing...

なお、パラメータをコマンドラインから指定することもできます。この場合、ウィザードは コマンドラインで指定されていないものについて問い合わせを表示します。下記では 例4.1「vm-install を使用したコマンドラインからの対話型セットアップ」(28 ページ) で指定した すべてのパラメータについて、存在するコマンドラインスイッチを指定しています。パラメータの全リストについては、`man 8 vm-install` コマンドで 表示されるマニュアルページをお読みください。

例 4.2 vm-install コマンドラインスイッチ

```
vm-install --os-type sles11❶ --name "sles11_test"❷ ¥
--vcpu 2❸ --memory 512❹ --max-memory 768❺ ¥
--disk /var/lib/kvm/images/sles11/hda,0,disk,w,8000,sparse=1❻ ¥
--disk /iso/SLES-11-SP1-DVD-x86_64-GM-DVD1.iso,1,cdrom❼ ¥
--nic mac=52:54:00:05:11:11,model=virtio❽ ¥
--graphics cirrus❾ --config-dir "/etc/libvirt/qemu"❿
```

- ❶ 適切な既定値を設定するため、ゲスト側のオペレーティングシステムを 指定しています。指定可能な値の一覧を表示するには、`vm-install -0` コマンドを実行してください。
- ❷ VM ゲスト の名前を指定しています。ここでの名前はユニークなもので なければなりません。
- ❸ 仮想プロセッサ数を指定しています。
- ❹ メモリの初期量を指定しています。
- ❺ メモリの最大量を指定しています。オペレーティングシステム側に 擬似仮想化 [virtio-balloon] ドライバが必要です。
- ❻ 仮想ハードディスクを指定しています。このファイルは `/var/lib/kvm/images/sles11/hda` に存在しています。このハードディスクは最初 (0) のハードディスク (disk) として使用されるものです。また、このディスクは 書き込み可能 (w) で、8GB (8000) のサイズです。さらに、VM ホストサーバ 上のファイルはスパースファイル (sparse=1) です。
- ❼ ISO イメージを CD-ROM ドライブとして、2 番目の (1) ブロックデバイスに割り当てています。
- ❽ MAC アドレスを 52:54:00:05:11:11 として、擬似仮想化ネットワークデバイスを設定しています。
- ❾ グラフィックカードを指定しています。

- ⑩ 仮想マシンの XML 設定ファイルを保存するためのディレクトリを指定しています。なお、既定のディレクトリ `/etc/libvirt/qemu` を利用することを強くお勧めします。

4.2.1 インストールを起動しない形での VM ゲストの設定

`vm-install` コマンドには `--no-install` というパラメータがあります。このパラメータを指定すると VM ゲストを定義する XML 設定ファイルが作成されるものの、ゲストが自動では起動しなくなります。このパラメータは、`vm-install` をウィザードモードで起動するか、もしくはコマンドラインですべてのパラメータを指定するかに関わらず利用することができます。

警告: 仮想ディスクの作成について

`vm-install` を実行する際、`--no-install` パラメータを指定すると、仮想ディスクの作成が行われなくなります。そのため、`qemu-img` や `virsh` を利用して、後からディスクを作成しなければなりません。

VM ゲストに対する XML 設定ファイルを作成したあとは、Virtual Machine Manager や `virsh` 側で認識されるようにするため、「登録」を行なう必要があります。登録は下記のようにして行ないます:

```
virsh -c qemu:///system define XML ファイルへのパス
```

4.3 高度なゲストインストール方法

この章では、通常の方法以外でのインストール手順、たとえばアドオンパッケージを追加したりするための手順について説明しています。

4.3.1 インストール時にアドオン製品を含める方法

openSUSE や SUSE Linux Enterprise Server などのオペレーティングシステムの場合、インストール作業の途中でアドオン製品を含めるようにすることができます。アドオン製品のインストール元がネットワーク上に存在する場合、VM ゲストに

ついて特別な設定を行なう必要はありません。CD/DVD や ISO イメージで提供されている場合は、VM ゲスト の インストールシステムに対して、標準のインストールイメージとアドオン製品の 両方のイメージを指定する必要があります。

この場合は、1 つめとして標準のインストールイメージを、2 つめとして 実際の CD/DVD-ROM ドライブやアドオンのイメージを指定してください。これは、最初に設定したイメージやデバイスが起動用に使用されるためです。openSUSE や SUSE Linux Enterprise Server をインストールする場合、/dev/sr0 に標準のインストールイメージを、/dev/sr1 にアドオン製品をそれぞれ設定してください。

基本的な VM ゲスト の管理

VM ゲスト の起動や停止など、基本的な管理作業は Virtual Machine Manager のようなグラフィカルな アプリケーションから行なうことができるほか、`virsh` を利用して コマンドラインが行なうこともできます。VNC を介したグラフィカルコンソールへの 接続は、お使いのマシンでグラフィカルユーザインターフェイスを利用している場合に のみ可能です。

5.1 VM ゲスト の一覧表示

VM ゲスト の一覧を表示するには、まず VM ホストサーバ に接続する必要があります。VM ホストサーバ 自身で管理ツールを起動している場合は、既に接続が完了しています。リモートから操作している場合は、6.3項「VM ホストサーバ への接続」(63 ページ)にある手順をお読みください。

5.1.1 Virtual Machine Manager を利用した VM ゲスト の一覧表示

Virtual Machine Manager のメインウィンドウには、接続中の VM ホストサーバ で定義されているすべての VM ゲスト が表示されています。それぞれの VM ゲスト にはマシンの名前のほか、その状態 (`[Running]`, `[Paused]`, または `[Shutoff]`) がアイコンとテキストの両方で表示され、CPU の使用率の表示バーが併記されます。

5.1.2 virsh を利用した VM ゲスト の一覧表示

VM ゲスト の一覧を表示するには、`virsh list` コマンドを実行します:

localhost 上で動作しているゲストの一覧

```
virsh -c qemu:///system list
```

TLS 接続を利用した、リモートの `wilber` で動作する実行中および停止中のゲスト一覧

```
virsh -c qemu+tls://wilber@mercury.example.com/system list --all
```

SSH トンネルを利用した、リモートの `tux` で動作する実行中および停止中のゲスト一覧

```
virsh -c qemu+ssh://tux@mercury.example.com/system list --inactive
```

5.2 グラフィカルコンソールの表示

VM ゲスト のグラフィカルコンソールを開くことで、実際のマシンと同様に VNC 経由でマシンに接続することができるようになります。認証が必要な VNC サーバにアクセスしている場合は、ユーザ名 (設定されていれば) とパスワードをそれぞれ尋ねられます。

なお、VNC コンソールのウインドウ内でマウスのボタンを押すと、マウスカーソルは「捕獲された」状態になり、コンソールの外側には移動できなくなります。捕獲された状態を解除するには、`Alt + Ctrl` を押してください。

ヒント: シームレスなカーソルの移動について

カーソルがキャプチャ (捕捉) されてしまうことを防ぎ、かつゲストとホストの 間でシームレスにカーソルを移動できるようにするには、VM ゲスト にタブレット 入力デバイスを追加します。詳しくは 8.1 項「シームレスなカーソル移動とカーソル移動の同期」(88 ページ) をお読みください。

また、`Ctrl + Alt + Del` などの特定のキー入力については ホストシステム側で解釈され、VM ゲスト 側に渡すことはできません。

VM ゲスト に対してこのようなキー入力を渡したい場合は、VNC ウインドウから `[Send Key]` メニューを開き、送信したいキー入力を選択してください。 `[Send`

Key] のメニューは、Virtual Machine Manager と virt-viewer を使用した場合にのみ利用できます。

注記: 対応する VNC ビューア

一般的にはすべての VNC ビューアが VM ゲスト に接続できますが、SASL 認証や TLS/SSL での接続を行なっている場合には選択肢が限られたものになります。一般的な VNC ビューアとしては `tightvnc` や `tigervnc` などがありますが、いずれも SASL 認証や TLS/SSL 接続には対応していません。Virtual Machine Manager と `virt-viewer` 以外で対応可能なものは、`vinagre` だけしかありません。

5.2.1 Virtual Machine Manager を利用したグラフィカルコンソールの表示

- 1 Virtual Machine Manager 内で VM ゲスト の項目を選択し、マウスの右ボタンを押します。
- 2 ポップアップメニューから `[Open]` を選択します。

5.2.2 virt-viewer を利用したグラフィカルコンソールの表示

`virt-viewer` はシンプルな VNC ビューアで、VM ゲスト の コンソールを表示するための追加機能が備わっています。たとえば本ソフトウェア では「待機」モードに対応していて、VNC に接続する前に VM ゲスト が起動するのを待つことができます。また、VM ゲスト が再起動したような場合でも、自動的に再接続する機能が備わっています。

`virt-viewer` は VM ゲスト を名前で指定できるほか、ID や UUID でも指定することができます。`virsh list --all` と入力することで一覧を表示できます。

起動中や一時停止中のゲストに接続する場合は ID, UUID, 名前のいずれかを指定して 接続することができます。一方、シャットダウンされている VM ゲスト には ID が 付与されていないため、UUID または名前で接続します。

ID 8 のゲストにローカル接続する

```
virt-viewer -c qemu:///system 8
```

現時点では動作していないゲスト `sles11` に対して、ゲストが起動され次第接続する

```
virt-viewer -c qemu:///system --wait sles11
```

`--wait` オプションを指定すると、VM ゲスト がその時点で 動作していても、その接続を行なうことができるようになるまで待機します。ゲストが起動すると、ビューアが起動します。

ssh を利用したリモート接続

```
virt-viewer -c qemu+ssh://tux@mercury.example.com/system -w sles11
```

詳しくは `virt-viewer --help` で表示されるヘルプか、もしくは `man 1 virt-viewer` で表示されるマニュアルページをお読みください。

5.3 VM ゲスト の状態変更: 起動, 停止, 一時停止

VM ゲスト の起動／停止／一時停止は、Virtual Machine Manager または `virsh` から行なうことができます。また、VM ホストサーバ の起動時に自動的に VM ゲスト を起動 するように設定することもできます。

VM ゲスト をシャットダウンする場合、シャットダウン作業をグレースフル (`graceful`, "上品な" という意味) モードまたは強制モードで行なうことができます。強制モードは通常のマシンで言うところの電源プラグを引き抜く行為と同じ意味で、他の シャットダウン方法が存在しない場合にのみ行なうべきものです。また、強制シャット ダウンはファイルシステムの破壊をもたらす場合があるほか、VM ゲスト 上のデータを失う可能性もあります。

ヒント: グレースフル (`graceful`) シャットダウン

グレースフルシャットダウンを実現できるようにするためには、VM ゲスト 側を ACPI に対応させなければなりません。`vm-install` や Virtual Machine Manager でゲストを 作成した場合には、ACPI が利用できるようになっているはずです。ACPI に対応できているかどうか、Virtual Machine Manager から確認するには下記の手順を実施します:

まずは Virtual Machine Manager 内で確認を行ないたい VM ゲスト の項目を選び、ダブルクリックします。あとはメニューから `[View] > [Details]` を選択

し、[Overview] > [Machine Settings] を選択します。表示された画面で、[ACPI] がチェックされていれば対応済みです。

ゲスト側のオペレーティングシステムにもよりますが、ACPI を有効にするだけでは 足りない場合があります。そのため、本番運用をはじめる前に、ゲスト側のシャットダウンと再起動がそれぞれ正しく動作するかどうか、事前に確かめておくことを強くお勧めします。たとえば openSUSE や SUSE Linux Enterprise Desktop の場合は、シャットダウンや再起動を行なうのに PolKit 認証が必要になる場合がありますが、このポリシーが 全ての VM ゲストで無効化されていることをご確認ください。

また、Windows XP/Server 2003 のゲストをインストールする際に ACPI が有効化されている場合、VM ゲスト の設定だけの有効化では不十分です。詳しくは下記の記事をお読みください:

<http://support.microsoft.com/kb/314088/>
<http://support.microsoft.com/?kbid=309283>

グレースフルシャットダウンは、VM ゲスト の設定だけでなく、もちろんゲスト側のオペレーティングシステムで有効化されていなければなりません。

5.3.1 Virtual Machine Manager を利用した VM ゲスト の状態変更

VM ゲスト の状態変更は、Virtual Machine Manager のメインウィンドウから行なうことができるほか、VNC ウィンドウからも変更することができます。

手順 5.1 Virtual Machine Manager ウィンドウからの状態変更

- 1 VM ゲスト の項目を選び、マウスの右ボタンを押します。
- 2 表示されたポップアップウィンドウから、[Run], [Pause] または [Shutdown options] 内にあるいずれかの項目を選択します。

手順 5.2 VNC ウィンドウからの状態変更

- 1 5.2.1項「Virtual Machine Manager を利用したグラフィカルコンソールの表示」(37 ページ) に書かれた手順に従い、VNC ウィンドウを開きます。
- 2 ツールバー、もしくは [Virtual Machine] のメニューから、[Run], [Pause] または [Shutdown options] 内にあるいずれかの項目を選択します。

5.3.1.1 VM ゲスト の自動起動

VM ホストサーバ の起動時にゲスト側を自動的に起動するようにする設定は、既定では 有効になっていません。また、この機能はそれぞれの VM ゲスト で個別に有効化する 必要があります。一括で有効化する方法はありません。

- 1 Virtual Machine Manager 内から設定したい VM ゲスト を選択し、ダブルクリックしてコンソールを 開きます。
- 2 [View] > [Details] を選択し、VM ゲスト の設定ウインドウを開きます。
- 3 [Boot Options] を選択し、[Start virtual machine on host boot up] にチェックを 入れます。
- 4 あとは [Apply] を押せば、設定を保存することができます。

5.3.2 virsh を利用した VM ゲスト の状態変更

下記の例では、「sles11」という名前の VM ゲスト の状態を 変更しています。

起動

```
virsh -c qemu:///system start sles11
```

一時停止

```
virsh -c qemu:///system suspend sles11
```

再起動

```
virsh -c qemu:///system reboot sles11
```

グレースフルシャットダウン

```
virsh -c qemu:///system shutdown sles11
```

強制シャットダウン

```
virsh -c qemu:///system destroy sles11
```

自動起動の有効化

```
virsh -c qemu:///system autostart sles11
```

自動起動の無効化

```
virsh -c qemu:///system autostart --disable sles11
```

5.4 VM ゲスト の保存と復元

VM ゲスト を保存すると、ゲスト内のメモリ内にある情報を全て保存することができます。実際のコンピュータに置き換えると、コンピュータの *ハイバネーション* と似たようなことを行なうことになります。保存された VM ゲスト は、その保存された 状態に素早く復元することができます。

いったん保存作業を行なうと VM ゲスト は一時停止状態になり、メモリ上に保持されていた情報をディスクに書き込んでから停止状態に移行します。この操作では VM ゲスト の仮想ディスクのコピーは作成されません。また、仮想マシンを保存するのにかかる 時間は割り当てたメモリ量に依存して決まります。なお、VM ゲスト の保存が完了すると、VM ゲスト が確保していたメモリは VM ホストサーバ 側に返却されます。

一方、復元操作はメモリ状態を保存していたファイルを VM ゲスト に読み込んで起動する 動作を行ないます。ゲストは起動処理を行なうことはなく、保存された状態に戻ることに なります。実際のコンピュータに置き換えると、ハイバネーションからの復元処理と似た ようなことを行なうことになります。

VM ゲスト は、そのメモリ状態をファイルに保存するため、保存先のパーティションに十分な領域が存在することをご確認ください。ゲスト側で下記のコマンドを実行すると、どれだけの領域が必要になるのかを概算で見積もることができます：

```
free -m | awk '/^Mem:/ {print $3}'
```

警告

保存処理が完了したあとは、保存した VM ゲスト を起動したり開始したりしてはなりません。これを行なってしまうと、マシンの仮想ディスクと保存されたメモリ状態の 同期が取れなくなってしまう、ゲストの復元時に致命的なエラーが発生する結果に なります。

5.4.1 Virtual Machine Manager を利用した保存と復元

手順 5.3 VM ゲスト の保存

- 1 VM ゲスト に対する VNC 接続ウィンドウを開きます。また、ゲストが動作していることを確認します。

- 2 [Virtual Machine] > [Save] を選択します。
- 3 保存先の場所とファイル名を選択します。
- 4 [Save] を押します。ゲストの状態保存には、しばらく時間が 必要です。操作が完了した後は、VM ゲスト は自動的にシャットダウン状態に なります。

手順 5.4 VM ゲスト の復元

- 1 Virtual Machine Manager を起動します。
- 2 Alt + R を押すか、もしくは [File] > [Restore Saved Machine] を選択します。
- 3 復元したいファイルを選択し、[Open] を押します。ファイルが正しく読み込まれると、VM ゲスト の動作が元に戻ります。

5.4.2 virsh を利用した保存と復元

起動中の VM ゲスト の保存は、`virsh save` を実行することで行ないます。このとき、保存先のファイル名を指定します。

opensuse11 という名前のゲストを保存する

```
virsh save opensuse11 /virtual/saves/opensuse11.vmsav
```

ID 37 のゲストを保存する

```
virsh save 37 /virtual/saves/opensuse11.vmsave
```

復元を行なうには、`virsh restore` を実行します:

```
virsh restore /virtual/saves/opensuse11.vmsave
```

5.5 VM ゲスト の削除

VM ゲスト を削除すると、既定では XML 設定ファイルを削除します。接続されていたストレージは既定では削除されないため、他の VM ゲスト を設定してそこから使用することができます。Virtual Machine Manager を利用している場合は、ゲスト側のストレージファイルも 削除するようにすることができます。これにより、ゲストに対する全ての資材を削除することができます。

VM ゲスト を削除するには、まずシャットダウンを行ないます (手順については 5.3 項「VM ゲスト の状態変更: 起動, 停止, 一時停止」(38 ページ) をお読みください)。実行中のゲストを削除することはできません。

5.5.1 Virtual Machine Manager を利用した VM ゲスト の削除

- 1 Virtual Machine Manager 内から、削除したい VM ゲスト の項目を選んでマウスの右ボタンを押します。
- 2 表示されたポップアップメニューから、[*Delete*] を選択します。
- 3 確認ウインドウが表示されますので、さらに [*Delete*] を押すと VM ゲスト を恒久的に削除することができます。削除処理は取り消すことができない のでご注意ください。

なお、ゲストの仮想ディスクについても同時に削除する場合は、[*Delete Associated Storage Files*] にチェックを入れます。この削除作業についても、取り消すことはできないのでご注意ください。

5.5.2 virsh を利用した VM ゲスト の削除

virsh を利用して VM ゲスト を削除するには、`virsh undefine VM 名` を実行します。ゲスト側の仮想ディスクを自動的に削除する機能は用意されていません。

接続と権限

複数の VM ゲストを受け持つ VM ホストサーバが複数台存在するような環境では、すぐに管理が行き詰まってしまう。libvirt には複数の VM ホストサーバに対して同時接続できるという機能があることから、これを利用してすべての VM ゲストとそれらのグラフィカルコンソールを管理することができます。

権限のあるユーザだけがアクセスできるようにするため、libvirt では各種の接続タイプ (TLS, SSH, Unix ソケット, TCP) と認証メカニズム (socket, PolKit, SASL, Kerberos) が提供されています。

6.1 認証

VM ゲストを管理したり、それらのグラフィカルコンソールにアクセスしたりする権利は、限られたユーザに対してのみ提供されるべきものです。これを実現するため、VM ホストサーバ側で下記のような認証方式を利用することができます：

- パーミッションとグループ所有者を設定できる、UNIX ソケットによるアクセス制御。この方式は libvirtd 接続でのみ利用できます。
- PolKit を利用した UNIX ソケットのアクセス制御。この方式はローカルの libvirtd 接続でのみ利用できます。
- SASL (Simple Authentication and Security Layer) を利用した、ユーザ名とパスワードによる認証。この方式は libvirtd と VNC の両方で利用できます。なお SASL での認証を行なう場合、サーバ側に実際のユーザアカウントを作成する必要はありません。これは、ユーザ名とパスワードを独自のデータベース内に保存するためです。なお、SASL での認証接続は暗号化されます。

- Kerberos による認証。この方式は libvirtd 接続でのみ利用できますが、このマニュアルでは説明していません。詳しくは http://libvirt.org/auth.html#ACL_server_kerberos をお読みください。
- パスワードのみの認証。この方式は VNC 接続でのみ利用できます。

重要: libvirtd と VNC の認証について

VM ゲスト の管理機能にアクセスする機能とそれらのグラフィカルコンソールにアクセスする機能は、それぞれ別々に設定する必要があります。管理ツールへのアクセスを制限しても、それらの制限が VNC 接続に自動で反映されることはありません。

TLS/SSL 接続を利用して VM ゲスト へのリモートアクセスを行なう場合、各 クライアントのアクセス制御は、証明書の鍵ファイルの読み取り権限を特定の グループに付与して制限することによって、間接的に実現することができます。詳しくは 6.2.2.5 項「アクセス制限 (セキュリティ面の考慮事項)」(60 ページ) をお読みください。

6.1.1 libvirtd の認証

libvirtd の認証は、`/etc/libvirt/libvirtd.conf` で設定します。このファイルで設定した内容は、Virtual Machine Manager や `virsh` など、すべての libvirt ツールに適用されます。

libvirt では 2 種類のソケットを提供しています。1 つは監視用に使用する 読み込みのみのソケット、もう 1 つは管理用に使用する読み書き可能なソケットです。両方のソケットに対するアクセスは別々に設定します。既定では両方のソケットは `root.root` が所有する設定になっています。また、既定では読み書き可能なソケットに対するアクセス権は `root` だけに 制限 (0700) され、読み込みのみのソケットに対するアクセス権は 誰にでもできるように設定 (0777) されています。

下記では、読み書き可能なソケットに対してアクセス権を設定する手順を説明しています。読み込みのみのソケットに対しても同じ手順で設定できます。なお、すべての手順は VM ホストサーバ 上で実施します。

注記: openSUSE における既定の認可設定

openSUSE における既定の認証方法は、UNIX ソケットを利用した アクセス制御です。この方法では `root` だけが認証されます。VM ホストサーバ 内に存在

する root 以外のユーザで libvirt ツールにアクセスすると 最初の 1 回だけ、PolKit による root のパスワード認証が求められます。パスワードが正しいことが確認されると、その時点でのセッションと今後の セッションでアクセスが許可されるようになっています。

それ以外にも、libvirt を設定することで非特権ユーザに「システム」アクセスを許可するように設定することもできます。詳しくは 6.3.1 項「非特権ユーザに対する「システム」アクセス」(65 ページ) をお読みください。

推奨される認証設定

ローカル接続

6.1.1.2 項「PolKit を利用した UNIX ソケットによるローカルアクセス制御」(48 ページ)

6.1.1.1 項「パーミッションとグループ所有者を利用した UNIX ソケット向けのアクセス制御」(47 ページ)

SSH を利用したリモートからのトンネル

6.1.1.1 項「パーミッションとグループ所有者を利用した UNIX ソケット向けのアクセス制御」(47 ページ)

TLS/SSL を利用したリモート接続

6.1.1.3 項「SASL を利用したユーザ名とパスワードでの認証」(49 ページ)
なし (証明書に対するアクセスを制限することで、クライアント側からの アクセス制御を行なう)

6.1.1.1 パーミッションとグループ所有者を利用した UNIX ソケット向けのアクセス制御

root 以外のアカウントに対してアクセスを許可するように設定するには、ソケットの所有者グループを変更することで実現することができます (下記の例では libvirt グループに設定する場合を示しています)。この認可方法は、ローカル接続と SSH 接続に対して適用できます。

1 まずソケットを所有させるグループが存在しない場合は、それを作成します:

```
groupadd libvirt
```

重要: グループの存在について

libvirtd を再起動するまでの間にグループを作成しなければなりません。作成を行っていないと、再起動が失敗します。

2 アクセスを許可したいユーザをグループに追加します:

```
usermod -A libvirt tux
```

3 あとは設定ファイル /etc/libvirt/libvirtd.conf を下記のように変更します:

```
unix_sock_group = "libvirt"❶  
unix_sock_rw_perms = "0770"❷  
auth_unix_rw = "none"❸
```

- ❶ 所有者グループを libvirt に設定しています。
- ❷ ソケットに対するパーミッションを設定しています (srwxrwx---)。
- ❸ その他の認証方法 (PolKit や SASL など) を無効化しています。アクセスはソケットのパーミッションでのみ設定します。

4 libvirtd を再起動します:

```
rclibvirtd restart
```

6.1.1.2 PolKit を利用した UNIX ソケットによるローカルアクセス制御

PolKit を利用した UNIX ソケットのアクセス制御は、ローカル接続の場合の openSUSE における既定の制御方法です。そのため、libvirt の設定 変更を行なう必要はありません。PolKit による制御が働いている場合は、既定では両方のソケットともパーミッションが 0777 に設定され、ソケットに対してアクセスするアプリケーションは PolKit による認証を受けなければならなくなります。

重要: PolKit の認証はローカル接続に対してのみ動作します

PolKit はリモートの認証に対応していないため、PolKit による認証は、VM ホストサーバ上のローカル接続に対してだけ使用することができます。

libvirt のソケットにアクセスする場合、2 種類のポリシーが存在します:

- *org.libvirt.unix.monitor*: 読み込み専用のソケットに対するアクセス
- *org.libvirt.unix.manage*: 読み書き可能なソケットに対するアクセス

既定では読み書き可能なソケットに対するアクセスポリシーは、root のパスワードによる認証を 1 度だけ実施し、あとはその時点のセッションと 今後のセッションでアクセスが許可されるようになっていきます (auth_admin_keep_always)。

root のパスワードを入力することなく、読み書き可能なソケットに対して アクセスを許可するようになるには、2 種類の方法があります：

1. polkit-auth コマンドを利用することで、制限のない アクセスを許すことができます：

```
polkit-auth --user tux --grant org.libvirt.unix.manage # 権限の付与
polkit-auth --user tux --revoke org.libvirt.unix.manage # 権限の取り消し
```

2. /etc/PolicyKit/PolicyKit.conf ファイルでは、より 詳しいオプションを提供しています。既存の <config version="0.1"> から </config> までの間に、下記の XML テキストを 入力してください：

```
<match action="org.libvirt.unix.manage">①
  <match user="tux">②
    <return result="yes"/>③
  </match>
</match>
```

- ① ポリシーの名前を指定しています。org.libvirt.unix.manage とは、読み書き可能なソケットに対するアクセスを意味します。
- ② 権限を与えるユーザを指定しています。複数を指定したい場合は、| で区切ってください (user="tux|wilber")。
- ③ 与える権限の種類を指定します。下記のいずれかを指定します：yes (制限なくアクセスを許可する), no (完全にアクセスを拒否する), auth_self または auth_admin (それぞれ権限が要求された場合に、自分自身のパスワードか root のパスワードを確認する), auth_self_keep_session または auth_admin_keep_session (権限が要求された場合に 自分自身のパスワードか root のパスワードを確認する。いったん 確認が完了すると、以後の同セッションでは確認を不要とする), auth_self_keep_always または auth_admin_keep_always (権限が要求された場合に 自分自身のパスワードか root のパスワードを確認する。いったん 確認が完了すると、以後ずっと確認を不要とする)。

6.1.1.3 SASL を利用したユーザ名とパスワードでの認証

SASL はユーザ名とパスワードによる認証を提供するほか、データの暗号化 (既定では digest-md5) も提供します。SASL は独自のユーザデータベースを 持っています。

るため、VM ホストサーバ上に実際のユーザを作成する必要はありません。また、SASL を利用するには TCP 接続と、TLS/SSL 接続が必要です。

重要: 暗号化しない環境での TCP と SASL 認証

暗号化をしない環境で TCP 接続の digest-md5 認証を行なう場合、これは 本番環境で利用するには十分なセキュリティを確保できません。このような 使用形態は、テスト環境でのみ使用されるべきものです。

ヒント: TLS/SSL を利用した SASL 認証

リモートからの TLS/SSL 接続を受け付ける場合、証明書の鍵ファイルに対するアクセス制限を行なうことで、クライアント/側でのアクセス制御を実現することができます。ただし、多数のクライアントからのアクセスを受け付ける場合は、設定を間違えてしまいやすい仕組みでもあります。TLS 経由での SASL を利用し、サーバ側でのアクセス制御を併用してください。

SASL 認証を設定するには、下記の手順で行ないます:

1 /etc/libvirt/libvirtd.conf 内の設定を、下記のように 変更します:

1a TCP 接続に対して SASL を有効にする場合:

```
auth_tcp = "sasl"
```

1b TLS/SSL 接続に対して SASL を有効にする場合:

```
auth_tls = "sasl"
```

2 libvirtd を再起動します:

```
rc libvirtd restart
```

3 libvirt の SASL 設定ファイルは /etc/sasl2/libvirtd.conf にあります。通常は、この設定ファイルを既定値から変更する必要はありません。しかしながら、TLS 経由で SASL を使用するような場合は、処理の重複を防ぐために セッションの (SASL 内の) 暗号化を無効化することができます。セッションの暗号化を無効にするには、mech_list をコメントアウトしてください。TCP 接続の場合は digest-md5 に設定しなければなりません:

```
mech_list: digest-md5    # TCP 接続での必須設定
#mech_list: digest-md5   # TLS/SSL 接続でのみ適用されるべき設定
```

- 4 既定では SASL のユーザは何も設定されていないため、誰もログインできません。それぞれ下記のコマンドを利用してユーザの追加や一覧表示、削除を行なってください:

```
mercury:~ # saslpasswd2 -a libvirt tux                # ユーザ tux の追加
Password:
Again (for verification):
mercury:~ # sasldblistusers2 -f /etc/libvirt/passwd.db # ユーザの一覧表示
tux@mercury.example.com: userPassword
mercury:~ # saslpasswd2 -a libvirt -d tux              # ユーザ tux の削除
```

ヒント: virsh と SASL 認証

SASL 認証を使用している場合、virsh コマンドを実行するたびにユーザ名とパスワードの入力を求められます。これを回避するには、virsh をシェルモードで起動してください。

6.1.2 VNC 認証

VM ゲストのグラフィカルコンソールに対するアクセス制御は、libvirt や QEMU で制御されているものではないため、VNC 認証の設定には追加の作業が必要になります。中心となる設定ファイルは /etc/libvirt/qemu.conf です。

VNC 認証では 2 種類の認証方法を提供しています。1 つは SASL による認証、もう 1 つはパスワードだけを利用する認証です。libvirt の認証に SASL を使用している場合は、VNC 認証に対しても SASL を利用しておくことを強くお勧めします。これは同じデータベースを共有することができるためです。

VM ゲストに対するアクセスを制限するための 3 つめの方法として、VNC サーバ上で TLS による暗号化を使用する方法があります。これは VNC クライアント側に x509 クライアント証明書を配置して認証する方法で、これらの証明書に対するアクセスをクライアント側で制御することで、アクセスを間接的に制限することができますようになります。詳しくは「TLS/SSL を利用する VNC: クライアントの設定」(59 ページ)をお読みください。

6.1.2.1 SASL を利用したユーザ名とパスワードでの認証

SASL はユーザ名とパスワードによる認証を提供するほか、データの暗号化も提供します。SASL は独自のユーザデータベースを持っているため、VM ホストサーバ上に実際のユーザを作成する必要はありません。また、libvirt に対し

て SASL を利用するには、TLS/SSL 接続を利用することもできます。これらの接続を設定する方法については、「TLS/SSL を利用する VNC: クライアントの設定」(59 ページ) をお読みください。

VNC 用に SASL 認証を設定するには、下記の手順で行ないます:

- 1 まずは SASL の設定ファイルを作成します。libvirt 向けに用意されたファイルをお使いになるのがお勧めです。既に libvirt 向けに SASL を設定していて、同じユーザ名／パスワードのデータベースを利用したい場合は、単純にリンクを作成するだけでかまいません:

```
ln -s /etc/sasl2/libvirt.conf /etc/sasl2/qemu.conf
```

VNC にのみ SASL を設定したい場合や、libvirt とは異なる設定を使用したい場合は、既存のファイルをベースとして使用するため、いったんコピーしてから必要な修正を行なってください:

```
cp /etc/sasl2/libvirt.conf /etc/sasl2/qemu.conf
```

- 2 既定では SASL のユーザは何も設定されていないため、誰もログインできません。それぞれ下記のコマンドを利用してユーザの追加や一覧表示、削除を行なってください:

```
mercury:~ # saslpasswd2 -a libvirt tux                # ユーザ tux の追加
Password:
Again (for verification):
mercury:~ # sasldblistusers2 -f /etc/libvirt/passwd.db # ユーザの一覧表示
tux@mercury.example.com: userPassword
mercury:~ # saslpasswd2 -a libvirt -d tux             # ユーザ tux の削除
```

- 3 /etc/libvirt/qemu.conf 内の設定を 下記のように修正します:

```
vnc_listen = "0.0.0.0"
vnc_sasl = 1
```

最初のパラメータでは、VNC をすべてのインターフェイスに対して (ローカルホストだけでなく) 公開し、接続を受け付けるように設定しています。2 つめのパラメータでは SASL 認証を有効にしています。

- 4 libvirtd を再起動します:

```
rclibvirtd restart
```

- 5 設定を変更する前から起動していたすべての VM ゲストについて、再起動を行ないます。VM ゲスト を再起動しない場合、VNC 接続時の SASL 認証は 有効化されません。

注記: 対応する VNC ビューア

現時点では TLS/SSL 接続と SASL 認証に対応した VNC ビューアは、Virtual Machine Manager, virt-viewer, vinagre の 3 種類です。

6.1.2.2 パスワードのみの認証

VNC サーバへのアクセスは、VNC パスワードで制御することもできます。すべての VM ゲスト に対して設定するグローバルパスワードを設定することができるほか、各ゲストに対して個別のパスワードを設定することもできます。後者の場合、VM ゲスト の設定ファイルを編集する必要があります。

注記: グローバルパスワードの設定について

パスワードのみの認証を使用している場合、それぞれの VM ゲスト に対して 個別のパスワードを指定している場合でも、グローバルパスワードを設定しておくことをお勧めします。これはマシン単位でのパスワードを設定し忘れてしまった場合でも、お使いの仮想マシンを「第二の」パスワードで 保護できるためです。また、グローバルパスワードは、その他のパスワードが そのマシンに設定されなかった場合にのみ使用されます。

手順 6.1 グローバル VNC パスワードの設定

- 1 /etc/libvirt/qemu.conf 内の設定を下記のように 変更します:

```
vnc_listen = "0.0.0.0"
vnc_password = "パスワード"
```

最初のパラメータでは、VNC をすべてのインターフェイスに対して (ローカルホストだけでなく) 公開し、接続を受け付けるように設定しています。2 つめのパラメータではパスワードを設定しています。パスワードに設定可能な最大の長さは 8 文字です。

- 2 libvirtd を再起動します:

```
relibvirtd restart
```

- 3 設定を変更する前から起動していたすべての VM ゲスト について、再起動を 行ないます。VM ゲスト を再起動しない場合、VNC 接続時のパスワード 認証は有効化されません。

手順 6.2 VM ゲスト 固有の VNC パスワードの設定

- 1 まずは下記のようにして、VNC をすべてのインターフェイスに対して (ローカルホストだけでなく) 公開し、接続を受け付けるように設定します。

```
vnc_listen = "0.0.0.0"
```

- 2 エディタで VM ゲスト の XML 設定ファイルを開きます。下記の例では *VM 名* の部分を VM ゲスト の名前に置き換えてください。使用されるエディタは、既定では \$EDITOR 変数で 指定されたものを使用します。この変数が設定されていない場合は、vi を使用します。

```
virsh edit VM 名
```

- 3 type='vnc' という属性を持った <graphics> の要素を探します。たとえば下記のような要素です:

```
<graphics type='vnc' port='-1' autoport='yes' />
```

- 4 passwd=パスワード という 属性を追加し、ファイルを保存しエディタを終了します。パスワードに設定 可能な最大の長さは 8 文字です。

```
<graphics type='vnc' port='-1' autoport='yes' passwd='PASSWORD' />
```

- 5 libvirtd を再起動します:

```
rc libvirtd restart
```

- 6 設定を変更する前から起動していたすべての VM ゲスト について、再起動を行ないます。VM ゲスト を再起動しない場合、VNC 接続時のパスワード 認証は有効化されません。

警告: セキュリティ

VNC プロトコルは安全なプロトコルとは考えられていません。パスワードは 暗号化されて送信されますが、攻撃者が暗号化されたパスワードと暗号鍵の 両方を傍受することができる場合、パスワードを解読できてしまいます。そのため、TLS/SSL を利用するか、もしくは SSH トンネルを経由するなどの 形で VNC を利用することをお勧めします。virt-viewer や Virtual Machine Manager のほか、バージョン 2.30 以降の vinagre で それら両方の方式に対応しています。

6.2 リモート接続の設定

libvirt の大きな利点のうちの 1 つに、複数の異なるリモートホスト上に ある VM ゲストを一括管理できるという機能があります。この章では、リモートから接続できるようにするための、サーバ側およびクライアント側の 設定方法を説明しています。

6.2.1 SSH を利用したリモートトンネル (qemu+ssh)

VM ホストサーバ 上で SSH のトンネルを利用したリモート接続を行なう場合は、SSH の接続を受け付ける設定だけを行なえば作業は完了します。それぞれ SSH デーモンが起動していること (rcsshd status) と、SSH サービスがファイアウォールで開かれていることを確認してください。

SSH 接続に対するユーザ認証は、従来通りのユーザ／グループと、パーミッションによるものです。詳しくは 6.1.1.1 項「パーミッションとグループ所有者を利用した UNIX ソケット向けのアクセス制御」(47 ページ) をお読みください。tux での接続 (qemu+ssh://tuxsIVname;/system) は即時に使用できる もので、libvirt 側では特に何も行なう必要はありません。

SSH 経由で qemu+ssh://ユーザ名@システム のように接続を行なった場合、ユーザ名 に対する パスワードを入力する必要があります。パスワード入力は、VM ホストサーバ 上の ~ユーザ名/.ssh/authorized_keys ディレクトリに公開鍵をコピーすることで避けることもできます。詳しくは 項「SSH 鍵のコピー」(第13章 SSH: 機密を保護する通信, ↑セキュリティガイド) をお読みください。マシン上で ssh-agent を使用するのも便利な方法です。こちらについては 項「ssh-agent コマンドの使用」(第13章 SSH: 機密を保護する通信, ↑セキュリティガイド) をお読みください。

6.2.2 x509 証明書を利用したリモートの TLS/SSL 接続 (qemu+tls)

TLS/SSL 暗号と x509 証明書による認証を行なう TCP 接続は、SSH を設定するよりも手順が複雑ですが、大規模な環境には適切な選択です。また、管理者が 変わる可能性のある環境で複数の VM ホストサーバ を管理するような場合には、この方法を選択してください。

6.2.2.1 基本的な考え方

基本的には、TLS (Transport Layer Security) は証明書を利用することで 2 台のコンピュータ間の通信を暗号化します。接続を開始するコンピュータは「クライアント証明書」を使用する「クライアント」として、接続を受け付けるコンピュータは「サーバ証明書」を使用する「サーバ」として扱われます。以下の説明では、中央にあるデスクトップ機からお使いの VM ホストサーバを管理する場合を例にしています。

なお、両方のコンピュータから接続が開始されることがある場合は、それぞれのコンピュータにクライアント証明書とサーバ証明書の両方を配置する必要があります。これはたとえば、一方のコンピュータから他方のコンピュータに VM ゲストを移設するような場合に該当します。

それぞれの x509 証明書には、それに対応する機密鍵ファイルが存在します。また、証明書自身の正当性を確認するには、証明書と機密鍵の両方の正しい組み合わせが必要です。さらに証明書が正しい所有者から発行されたものであることを確認するためには、証明機関 (CA) と呼ばれる中央機関から発行され署名されたものである必要があります。なお、サーバ証明書とクライアント証明書は同じ CA から発行されたものでなければなりません。

重要: ユーザ認証

リモートからの TLS/SSL 接続を利用する場合、基本的には特定の方向に対する 2 台のコンピュータ間のみの通信を許可することになります。この場合、クライアント側で証明書にアクセスできるユーザを制限することで、間接的にユーザ制限を行なうことができます。詳しくは 6.2.2.5 項「アクセス制限 (セキュリティ面の考慮事項)」(60 ページ) をお読みください。なお、libvirt では SASL を利用したユーザ認証にも対応しています。こちらについての詳細は 6.2.2.6 項「TLS ソケットを利用した SASL による中央集中型ユーザ認証」(62 ページ) をお読みください。

6.2.2.2 VM ホストサーバの設定

VM ホストサーバは接続を受け付ける側のマシンです。そのため、サーバ証明書をインストールする必要があります。証明機関の証明書についても、あわせてインストールしてください。証明書をインストールすると、libvirt は自動的に TLS サポートが有効になります。

- 1 A.2項「x509 クライアント／サーバ証明書の生成」(172 ページ) の手順に従って サーバ証明書を作成し、CA の証明書とともにエクスポートします。

- 2 VM ホストサーバ 上で下記のように入力してディレクトリを作成します:

```
mkdir -p /etc/pki/CA/ /etc/pki/libvirt/private/
```

下記のようにして証明書をインストールします:

```
/etc/pki/CA/cacert.pem  
/etc/pki/libvirt/servercert.pem  
/etc/pki/libvirt/private/serverkey.pem
```

重要: 証明書へのアクセス制限について

6.2.2.5項「アクセス制限 (セキュリティ面の考慮事項)」(60 ページ) で説明しているように、証明書へのアクセスが制限されていることを確認してください。

- 3 /etc/libvirt/libvirtd.conf ファイルを エディタで開き、listen_tls = 1 を設定します。設定が終わったら libvirtd を再起動します:

```
rclibvirtd restart
```

- 4 既定では libvirt は TLS の機密接続用に TCP ポート 16514 で待ち受け ます。必要であれば、ファイアウォールで左記のポートを開いてください。

重要: TLS を有効化した形での libvirtd の再起動

libvirt で TLS を有効にした場合、サーバ証明書をインストールする必要があります。サーバ証明書がインストールされていないと、libvirtd の再起動 が失敗します。また、証明書を変更した場合にも libvirtd の再起動が必要 となります。

6.2.2.3 クライアントの設定と設定テスト

クライアントは接続を行なう側のマシンです。そのため、クライアント 証明書をインストールする必要があります。証明機関の証明書についても、あわせてインストールしてください。

- 1 A.2項「x509 クライアント／サーバ証明書の生成」(172 ページ) の手順に従って クライアント証明書を作成し、CA の証明書とともにエクスポートします。
- 2 クライアント側で下記のように入力してディレクトリを作成します:

```
mkdir -p /etc/pki/CA/ /etc/pki/libvirt/private/
```

下記のようにして証明書をインストールします:

```
/etc/pki/CA/cacert.pem  
/etc/pki/libvirt/clientcert.pem  
/etc/pki/libvirt/private/clientkey.pem
```

重要: 証明書へのアクセス制限について

6.2.2.5項「アクセス制限 (セキュリティ面の考慮事項)」(60 ページ) で説明しているように、証明書へのアクセスが制限されていることを確認しておいてください。

- 3 下記のコマンドを入力することで、クライアントとサーバ間の接続テストを行なうことができます。下記の *mercury.example.com* には VM ホストサーバ の名前を指定してください。サーバ証明書を作成した時と同じ 完全修飾ドメイン名を入力します。

```
virsh -c qemu+tls://mercury.example.com/system list --all
```

設定が正しければ、VM ホストサーバ 上で libvirt を利用して作成した、すべての VM ゲスト の一覧が表示されます。

6.2.2.4 TLS/SSL を有効化した VNC 接続の有効化

現時点では TLS を利用した VNC 接続は、ごく少数のツールでしかサポート されていません。たとえばよく使われている *tightvnc* や *tigervnc* などのビューアでは TLS への対応は行なわれて いません。逆に動作が確認されているものとしては、Virtual Machine Manager (*virt-manager*), *virt-viewer*, GNOME VNC ビューア *vinagre* などがあります。

TLS/SSL を利用する VNC: VM ホストサーバ の設定

TLS/SSL を利用して VNC のグラフィカルコンソールにアクセスするには、VM ホストサーバ 側で下記の設定作業が必要です:

- 1 お買いのファイアウォールで、VNC サービスを開きます。
- 2 下記のように入力して */etc/pki/libvirt-vnc* ディレクトリを作成し、このディレクトリに対して証明書のリンクを作成 します:

```
mkdir -p /etc/pki/libvirt-vnc && cd /etc/pki/libvirt-vnc
```

```
ln -s /etc/pki/CA/cacert.pem ca-cert.pem
ln -s /etc/pki/libvirt/servercert.pem server-cert.pem
ln -s /etc/pki/libvirt/private/serverkey.pem server-key.pem
```

- 3** /etc/libvirt/qemu.conf ファイルを編集し、下記のようにパラメータを設定します:

```
vnc_listen = "0.0.0.0"
vnc_tls = 1
vnc_tls_x509_verify = 1
```

- 4** libvirtd を再起動します:

```
rclibvirtd restart
```

重要: VM ゲスト の再起動について

VNC の TLS 設定は VM ゲスト の起動時にのみ設定可能です。そのため、設定を変更する前から起動していたすべてのマシンについて、再起動を行なう必要があります。

TLS/SSL を利用する VNC: クライアントの設定

クライアント側で必要な唯一の作業は、選択したクライアントソフトウェアで 認識可能な場所に、x509 のクライアント証明書を配置することだけです。サポートされる各クライアント (Virtual Machine Manager, virt-viewer, vinagre) はそれぞれ別々の場所に証明書を配置する仕組み ですが、Virtual Machine Manager と vinagre については、全ユーザに適用 するシステム全体の証明書を配置する場所と、ユーザごとの証明書を配置する 場所が別々になっています。

Virtual Machine Manager (virt-manager)

リモートのホストに接続するには、Virtual Machine Manager に対して 6.2.2.3 項「クライアントの設定と設定テスト」(57 ページ) で説明している 設定手順を実施する必要があります。VNC のクライアント証明書については、 下記のディレクトリに配置します:

システム全体で使用する証明書の場合

```
/etc/pki/CA/cacert.pem
/etc/pki/libvirt-vnc/clientcert.pem
/etc/pki/libvirt-vnc/private/clientkey.pem
```

特定のユーザだけが使用する証明書の場合

```
/etc/pki/CA/cacert.pem
```

```
~/.pki/libvirt-vnc/clientcert.pem  
~/.pki/libvirt-vnc/private/clientkey.pem
```

virt-viewer

virt-viewer にはシステム全体で利用する証明書の 場所しか用意されていません:

```
/etc/pki/CA/cacert.pem  
/etc/pki/libvirt-vnc/clientcert.pem  
/etc/pki/libvirt-vnc/private/clientkey.pem
```

vinagre

システム全体で使用する証明書の場合

```
/etc/pki/CA/cacert.pem  
/etc/pki/vinagre/clientcert.pem  
/etc/pki/vinagre/private/clientkey.pem
```

特定のユーザだけが使用する証明書の場合

```
$HOME/.pki/CA/cacert.pem  
~/.pki/vinagre/clientcert.pem  
~/.pki/vinagre/private/clientkey.pem
```

重要: 証明書へのアクセス制限について

6.2.2.5項「アクセス制限 (セキュリティ面の考慮事項)」(60 ページ) で説明されている内容に従い、証明書へのアクセスを制限していることを ご確認ください。

6.2.2.5 アクセス制限 (セキュリティ面の考慮事項)

それぞれの x509 証明書は 2 種類の部品から構成されています。1 つは公開 されるべき証明書、もう 1 つは機密鍵です。クライアントは両方の部品を利用 することで認証を行なうことができます。そのため、クライアントの証明書と 機密鍵の両方を 読むことができるユーザであれば、誰でも VM ホストサーバに アクセスできるようになります。一方、サーバ証明書のうち両方の部品がそろって いれば、どんなマシンでも VM ホストサーバ になりすますことができます。このような ことを防ぐため、 少なくとも機密鍵についてはできるだけ不正にアクセスされない よう、制限を行なう 必要があります。最も簡単な解決方法としては、鍵ファイルに 対してパーミッションによるアクセス制限を行なう方法があります。

サーバ証明書

サーバ証明書は QEMU プロセスから読み込むことができます。openSUSE では、libvirt ツールから起動される QEMU のプロセスの所有者は root なっています。そのため、root が証明書をを読み込むことができます、それで十分ということになります:

```
chmod 700 /etc/pki/libvirt/private/  
chmod 600 /etc/pki/libvirt/private/serverkey.pem
```

/etc/libvirt/qemu.conf ファイルを編集して QEMU のプロセス所有者を変更した場合は、それにあわせて鍵ファイルの所有者を設定する必要があります。

システム全体で使用するクライアント証明書

システム全体で使用する鍵ファイルについてアクセスを制御するには、特定のグループに対して読み込みアクセスを許可するように設定し、そのメンバーだけが鍵ファイルを読み込むことができるようにします。下記の例では libvirt グループを作成し、clientkey.pem ファイルとその親ディレクトリに対して、libvirt の所有グループを設定しています。作業を行なうと、設定した所有者とグループにしかアクセスが許されなくなります。最後に tux を libvirt グループに追加することで、tux から鍵ファイルにアクセスできるようになります。

```
CERTPATH="/etc/pki/libvirt/"  
# libvirt グループの作成  
groupadd libvirt  
# ユーザ root とグループ libvirt に所有者を設定  
chown root.libvirt $CERTPATH/private $CERTPATH/clientkey.pem  
# パーMISSIONの設定  
chmod 750 $CERTPATH/private  
chmod 640 $CERTPATH/private/clientkey.pem  
# ユーザ tux をグループ libvirt に追加  
usermod -A libvirt tux
```

特定のユーザだけが使用する証明書

VM ゲストのグラフィカルコンソールに VNC 経由でアクセスする際、特定ユーザ向けのクライアント証明書は、ユーザのホームディレクトリ以下の ~/.pki に配置しておく必要があります。これらの証明書を使用する VNC ビューアなどでは、機密鍵のパーMISSIONを確認しない仕組みになっているため、ユーザ側の責任で機密鍵が他のユーザに読み取られないよう設定する必要があります。

サーバ側からのアクセス制限

既定では各クライアントには適切なクライアント証明書が配置され、TLS 接続が有効になっている VM ホストサーバに対して接続を行ないます。そのため、サー

バ側で SASL を利用した追加のサーバ側認証を行なうことができるようになっています。詳しくは 6.1.1.3 項「SASL を利用したユーザ名とパスワードでの認証」(49 ページ) をお読みください。

また、DN (識別名; Distinguished Names) のホワイトリスト機能を利用して、アクセス制限を行なうこともできます。この場合、DN の一覧に書かれた クライアントだけが接続できるようになります。

許可する DN を一覧に追加するには、`/etc/libvirt/libvirtd.conf` ファイル内の `tls_allowed_dn_list` を設定します。この値ではワイルドカードによる指定も可能です。なお、何も指定しない場合は すべての接続が拒否されることに注意してください。

```
tls_allowed_dn_list = [  
    "C=US,L=Provo,O=SUSE Linux Products GmbH,OU=*,CN=venus.example.com,EMAIL=*,  
    "C=DE,L=Nuremberg,O=SUSE Linux Products GmbH,OU=Documentation,CN=*"]
```

証明書に書かれている DN を表示するには、下記のコマンドを入力します:

```
certtool -i --infile /etc/pki/libvirt/clientcert.pem | grep "Subject:"
```

なお、設定を修正したあとは下記のようにして `libvirtd` を再起動します:

```
rclibvirtd restart
```

6.2.2.6 TLS ソケットを利用した SASL による中央集中型ユーザ認証

6.2.2.5 項「アクセス制限 (セキュリティ面の考慮事項)」(60 ページ) で書かれている手順でクライアント側のパーミッションを設定し、間接的にユーザ制限を行なうことはできますが、TLS で直接的なユーザ認証を行なうことはできません。しかしながら、`libvirt` でサーバ側でのユーザ認証を行ないたい場合、TLS に SASL (Simple Authentication and Security Layer) を追加することで、直接的なユーザ認証を実現することは可能です。詳しい設定方法については 6.1.1.3 項「SASL を利用したユーザ名とパスワードでの認証」(49 ページ) をお読みください。

6.2.2.7 トラブルシューティング

Virtual Machine Manager/virsh からサーバに接続できない

下記の順序で確認を行なってください:

ファイアウォールでポートを開いていますか？ (サーバ側で TCP ポート 16514 を開いておく必要があります)

Virtual Machine Manager や `virsh` を起動したユーザから、クライアント証明書 (証明書自身と機密鍵) を読み取ることができるように なっていますか？

サーバの証明書に書かれた完全修飾ドメイン名 (FQDN) と、接続時の完全修飾ドメイン名 (FQDN) が同じですか？

サーバ側で TLS が有効になっていますか？ (`listen_tls = 1`)

サーバ側で `libvirtd` を再起動しましたか？

VNC 接続が失敗する

まずは Virtual Machine Manager を利用してリモートのサーバに接続できるかどうかを確認してください。問題なく接続できる場合は、下記のようにしてサーバ上の仮想マシンで TLS が有効化されていることを確認します。下記の例は仮想マシンの名称が「sles11」である場合を示しています。

```
ps ax | grep qemu | grep ".*-name sles11" | awk -F" " -vnc " '{ print FS $2 }'
```

下記のように表示されていない場合は、そのマシンで TLS を有効化されていないということになります。TLS を有効化して再起動してください。

```
-vnc 0.0.0.0:0,tls,x509verify=/etc/pki/libvirt
```

6.3 VM ホストサーバ への接続

`libvirt` を利用してハイパーバイザに接続するには、URI (統一資源識別子; Uniform Resource Identifier) を指定する必要があります。この URI は `virsh` と `virt-viewer` を利用する場合 (ただし VM ホストサーバ 上で `root` として作業している場合を除きます) に必要となるほか、Virtual Machine Manager で必要に応じて使用します。`virt-viewer` の場合は接続パラメータとして指定することができます (例: `virt-manager -c qemu:///system`) が、接続 URI の作成をグラフィカルなインターフェイスで行なうこともできます。詳しくは 6.3.2 項「Virtual Machine Manager を利用した接続の管理」(66 ページ) をお読みください。

ハイパーバイザ^①+プロトコル^②::/ユーザ@リモート^③/接続種類^④

- ① ハイパーバイザを指定します。openSUSE では下記のハイパーバイザのみをサポートしています: `test` (テスト用のダミー)、`qemu` (KVM)、`xen` (Xen)。なお、このパラメータは必須です。
- ② リモートから接続を行なう場合、ここでプロトコルを指定します。プロトコルは以下のいずれかを指定します: `ssh` (SSH トンネル経由での接続)、`tcp`

(SASL/Kerberos 認証を利用した TCP 接続), `tls` (x509 証明書を利用した認証付きの TLS/SSL 暗号化接続)

- ③ リモートから接続を行なう場合、ここでユーザ名とリモートのホスト名を指定します。ユーザを指定しない場合は、コマンドを呼び出したユーザのユーザ名 (\$USER) が使われます。詳しくは下記をお読みください。TLS 接続の場合、ホスト名は x509 証明書と同じホスト名を指定しなければなりません。
- ④ QEMU のハイパーバイザに接続しようとしている場合、2 種類の接続タイプのうちからいずれかを選択します。完全なアクセス権を取得したい場合は `system` を、制限されたアクセスを取得したい場合は `session` を指定します。openSUSE では `session` アクセスをサポートしていないため、この文書内では `system` アクセスのみを説明しています。

ハイパーバイザの接続 URI 例

`test:///default`

ローカルのダミー用ハイパーバイザに接続します。テスト用に便利な方法です。

`qemu:///system`

ローカルホスト上の QEMU ハイパーバイザに接続し、完全なアクセス権 (`system` タイプ) を取得します。

`qemu+ssh://tux@mercury.example.com/system`

リモートの `mercury.example.com` ホスト上にある QEMU ハイパーバイザに接続します。接続は SSH トンネルを介して行ないます。

`qemu+tls://saturn.example.com/system`

リモートの `mercury.example.com` ホスト上にある QEMU ハイパーバイザに接続します。接続は TLS/SSL を介して行ないます。

詳細と指定例について、詳しくは <http://libvirt.org/uri.html> にある `libvirt` のドキュメンテーションをお読みください。

注記: URI 内でのユーザ名

Unix ソケット認証を使用する場合、ユーザ名／パスワード認証や PolKit を使用するかどうかに関わらず、ユーザ名を指定する必要があります。これはすべての SSH およびローカル接続に当てはまります。

SASL 認証を使用する場合 (TCP または TLS 接続) や、TLS 接続で追加のサーバ側認証を行なわない場合は、ユーザ名を指定する必要はありません。SASL の場合は、ここで入力したユーザ名は使用されず、ユーザ名とパスワードを別途に尋ねられることになります。

6.3.1 非特権ユーザに対する「システム」アクセス

上述のとおり、QEMU ハイパーバイザへの接続は、2 種類のプロトコル (セッション および システム) で実現できます。セッション 接続はクライアントプログラムと同じ権限で動作するもので、制限が存在する (USB/PCI デバイスの割り当てや 仮想ネットワークの設定ができなかったり、libvirtd に対するリモートアクセスが 制限されたりします) 都合上、主にデスクトップの仮想化に利用します。

システム 接続はサーバの仮想化に利用するもので、機能面での 制限はありませんが、root のみがアクセスできます。しかしながら、DAC (Discretionary Access Control; 任意アクセス制御) ドライバを libvirt に追加 することで、非特権ユーザに対して「システム」アクセスを許可できるようになります。「システム」アクセスを tux に許可するには、下記のようにします:

手順 6.3 一般ユーザに対する「システム」アクセスの許可

1 UNIX ソケットを利用したアクセス許可は、6.1.1.1 項「パーミッションとグループ所有者を利用した UNIX ソケット向けのアクセス制御」(47 ページ) に示されています。今回の例では、libvirt は libvirt グループ内のすべてのユーザに対してアクセスを許可し、tux ユーザをこのグループ のメンバーにしています。これにより、tux が virsh や Virtual Machine Manager を利用して接続できるようになります。

2 /etc/libvirt/qemu.conf を編集し、下記の設定項目を 変更します:

```
user = "tux"  
group = "libvirt"  
dynamic_ownership = 1
```

これにより、VM ゲスト は tux で起動されるようになり、ゲストに割り当てられたリソース (仮想ディスクなど) は、tux でアクセス／変更 できるようになります。

3 kvm グループに対して、tux を追加します:

```
usermod -A kvm tux
```

この手順は、VM ゲスト を起動するために必要なデバイス /dev/kvm にアクセスできるようにするために必要です。

4 最後に libvirtd を再起動します:

```
rclibvirtd restart
```

6.3.2 Virtual Machine Manager を利用した接続の管理

Virtual Machine Manager では管理対象のすべての VM ホストサーバ に対して 接続 を使用します。それぞれの接続には、そのホストで提供されているすべての VM ゲスト が含まれます。既定では、ローカルのホストに対する接続が自動で 設定 され、自動的に接続された状態になります。

すべての設定済みの接続は、Virtual Machine Manager のメインウィンドウ内に表示されます。有効な接続には小さな三角形マークが表示され、それを押すことで接続先にある VM ゲスト の一覧を展開したり折りたたんだりすることができます。

無効な接続はグレー (灰色) で表示され、Not Connected として印が付けられます。接続でダブルクリックを行なうか、もしくは右ボタンを 押すことで表示されるコンテキストメニューから *[Connect]* を選択することで、接続することができます。また、同じコンテキストメニュー から *[Delete]* を選択することで削除を行なうこともできます。

注記: 既存の接続の編集

既存の接続を編集することはできません。接続を変更したい場合は、新しい接続を作成して必要なパラメータを設定し、その後「古い」ほうの接続を 削除してください。

Virtual Machine Manager で新しい接続を作成するには、下記の手順で行ないます:

- 1 メニューから *[File]* > *[Add Connection]* を選択します。
- 2 ホストのハイパーバイザを *[Hypervisor]* で選択します (*[Xen]* または *[QEMU/KVM]*)。
- 3 次に接続のタイプを *[Connection]* で選択します。Virtual Machine Manager を起動しているホスト自身に接続する場合は *[Local]* を選択します。リモート接続の場合はいずれかを選択します (詳しくは 6.2 項「リモート接続の設定」(55 ページ) をお読みください)。
- 4 リモート接続の場合は、*[Hostname]* に接続先のホスト名を ユーザ名@リモートホスト の形式で指定します。SSH 接続やローカル接続の場合は、ユーザ名を指定しなければなりません。

重要: ユーザ名の指定について

TCP や TLS での接続の場合、ユーザ名を指定する必要はありません。指定を行っても無視されます。ユーザ名はローカル接続や SSH 接続の場合に指定しなければなりません。指定を行わない場合は、既定のユーザ root が使用されます。

- 5 Virtual Machine Manager の起動時に接続が自動的に有効化されないようにしたい場合は、[*Autoconnect*] のチェックを外します。
- 6 最後に [*Connect*] を押すと、設定完了です。

ストレージの管理

VM ホストサーバ 自身で VM ゲストを管理する場合、VM ゲストの仮想ハードディスクを追加したり既存のイメージを割り当てたりする際、VM ホストサーバのファイルシステム全体に対してアクセスして管理することができます。しかしながら、VM ゲストをリモートから管理する場合、これを行なうことはできません。このような理由から libvirt では、リモートからアクセスできる「ストレージプール」と呼ばれる仕組みが用意されています。

ヒント: CD/DVD ISO イメージ

リモートから VM ホストサーバ 上にある CD/DVD の iso イメージにアクセスできるようにするには、これらをストレージプール上に配置する必要があります。

libvirt ではそれぞれ、ストレージボリュームとストレージプールに対応しています:

ストレージボリューム

ストレージボリュームとは、ゲストに割り当てることのできるストレージデバイスで、仮想ディスクや CD/DVD/フロッピーディスクなどのイメージです。物理的な (VM ホストサーバ 上での) ブロックデバイス (パーティション、論理ボリュームなど) を割り当てることもできます。

ストレージプール

基本的にはストレージプールとは、ボリュームを保存する際に使用することのできる、VM ホストサーバ 上のストレージ資源を意味する用語です。デスクトップ機ではネットワークストレージの仕組みに似たもので、物理的には下記のような種類があります:

ファイルシステムのディレクトリ (*[dir]*)

イメージファイルを提供するディレクトリを指定する方法です。ファイルは 対応するディスク形式 (raw, qcow2, qed) のいずれか、もしくは ISO イメージ を利用することができます。

物理的なディスクデバイス (*[disk]*)

物理ディスクをストレージとして使用する方法です。パーティションは、プール に追加した各ボリュームに対して作成します。

事前フォーマット済みブロックデバイス (*[fs]*)

ファイルシステムのディレクトリ (イメージファイルを提供するディレクトリ) と 同じ方法でパーティションを指定する方法です。唯一の違いは、libvirt がデバイスのマウントを制御する、という点です。

iSCSI ターゲット (iscsi)

iSCSI のターゲット上にプールを作成する方法です。libvirt で利用できる ようにするには、事前に対象のボリュームにログインしておく必要があります。iSCSI のプールに対してパーティションを作成 する機能には対応していませんが、既存の論理ユニット番号 (LUN) がボリュームを 表わします。それらを使用する際、それぞれのボリューム (LUN) には、有効な (何も書かれていない) パーティションテーブルまたはディスクラベルが書かれている 必要があります。何も書かれていない場合は、fdisk を利用して追加します:

```
~ # fdisk -cu /dev/disk/by-path/ip-192.168.2.100:3260-iscsi-
iqn.2010-10.com.example:[...]-lun-2
Device contains neither a valid DOS partition table, nor Sun, SGI
or OSF disklabel
Building a new DOS disklabel with disk identifier 0xc15cdc4e.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

LVM ボリュームグループ (logical)

LVM のボリュームグループをプールとして使用する方法です。事前に設定された ボリュームグループを使用するか、もしくは使用するデバイスを指定して グループを作成します。ストレージボリュームは、ボリューム上のパーティション として作成されます。

警告: LVM ベースのプールに対する削除について

ストレージマネージャで LVM ベースのプールを削除すると、ボリュームグループ も削除されます。この削除は復元不可能な処理であるため、プール上に保存された全てのデータが失われることになります！

マルチパスデバイス ([*mpath*])

現時点ではマルチパスへの対応は、ゲストに対して既存のデバイスを割り当てる ことしかできません。libvirt からマルチパスデバイスを作成したり、設定したりすることには対応していません。

ネットワーク上の公開ディレクトリ ([*netfs*])

ファイルシステムのディレクトリ (イメージファイルを提供するディレクトリを指定する方法) と同じ方法で、ネットワーク上のディレクトリを指定します。ファイルシステムのディレクトリとの違いは、libvirt がデバイスのマウントを制御するという点です。プロトコルは NFS と glusterfs に対応しています。

SCSI ホストアダプタ ([*scsi*])

iSCSI ターゲットの場合と同様に、SCSI ホストアダプタを使用する方法です。なお、この方法を利用する場合は、`/dev/sdX` のような シンプルなデバイス名の形式ではなく、`/dev/disk/by-*` のような形式のデバイス名を使用するのがお勧めです。これは前者の方式の場合、デバイスの接続状況によって名前が変わってしまう可能性 (たとえばデバイスを追加したり削除したりした場合に、名前が変わってしまう可能性) があるためです。なお、iSCSI プール上にボリュームを作成する機能はサポートされていません。その代わりに、既存の論理ユニット番号 (LUN) がボリュームを表わします。

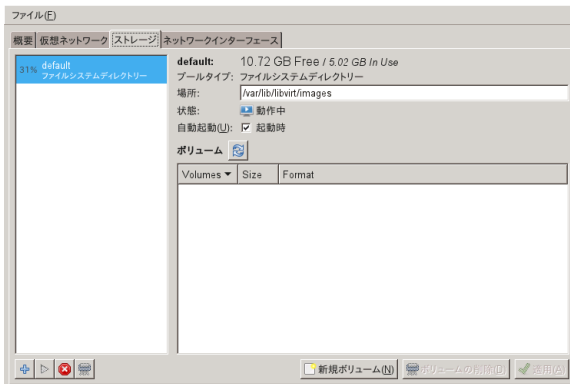
警告: セキュリティの考慮事項

データの損失や破壊を防ぐため、ストレージプールを構築する際に使用する LVM ボリュームグループや iSCSI ターゲットなどを VM ホストサーバ上で使用 (マウント) しないでください。VM ホストサーバ上からこれらの資源に接続したりマウントしたりする 必要はありません。libvirt がこれらの管理を行ないます。

また、VM ホストサーバ上のパーティションをラベルでマウントすることは避けてください。このような設定を行なっている場合、VM ゲスト内でも同じラベル名が使用される 可能性があるため、VM ホストサーバ側でのマウント時にトラブルを招く可能性があります。

7.1 Virtual Machine Manager を利用したストレージ管理

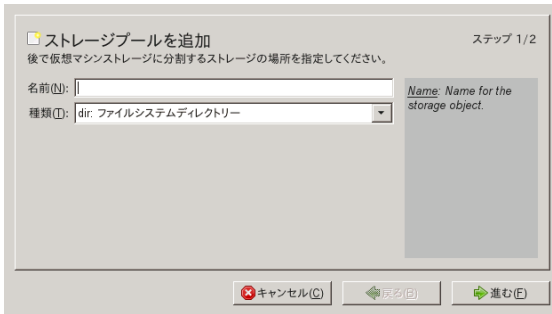
Virtual Machine Manager では、ストレージボリュームとプールを管理するのに、グラフィカルな インターフェイスであるストレージマネージャを提供しています。これを利用するには、接続を選んでからマウスの右ボタンを押して *[Details]* を選択するか、もしくは接続をハイライト表示させてから *[Edit]* > *[Connection Details]* を選択します。表示されたウィンドウで *[Storage]* タブを選択してください。



7.1.1 ストレージプールの追加

ストレージプールを追加するには、下記の手順で行ないます:

- 1 左下の隅にあるプラス型のシンボルを押し、*[Add a New Storage Pool Window]* を開きます。
- 2 まずは *[Name]* 欄で名前を指定します (半角英数字のほか、*[_.-]* の各記号を利用できます)。続いて *[Type]* を選び、*[Forward]* を押します。



□ ストレージプールを追加 ステップ 1/2

後で仮想マシンストレージに分割するストレージの場所を指定してください。

名前(N):

種類(T):

Name: Name for the storage object.

キャンセル(C) 戻る(B) 進む(F)

- 3 続いて表示されるウインドウでは、必要な詳細情報を指定します。入力すべきデータは、作成しようとしているプールの種類によって異なります。



□ ストレージプールを追加 ステップ 2/2

後で仮想マシンストレージに分割するストレージの場所を指定してください。

ターゲットパス(T): 参照(R)

フォーマット(O):

ホスト名(H):

ソースパス(S): 参照(R)

プールを構築(U): ☐

Source path: パスは、すでに存在するディスクデバイスでなければなりません。

キャンセル(C) 戻る(B) 完了(F)

[dir:] タイプの場合

- [Target Path]: 既存のディレクトリを選択します。

[disk:] タイプの場合

- [Target Path]: デバイスを提供するディレクトリを指定します。ほとんどの場合、既定値である /dev が適切な 値になります。
- [Format]: デバイスのパーティションテーブルの形式を指定します。ほとんどの場合は [auto] (自動) で問題なく動作するはずです。それ以外の場合は、parted -l を実行することで表示される一覧から形式を指定してください。
- [Source Path:] デバイスを指定します。たとえば ここでは /dev/sdX のような形式ではなく、/dev/disk/by-* の形式で 指定しておく

ことをお勧めします。これは前者の方式の場合、デバイスの 接続状況によって名前が変わってしまう可能性 (たとえばデバイスを追加 したり削除したりした場合に、名前が変わってしまう可能性) があるためです。また、ここではディスク上のパーティションではなく、ディスク全体を表わすパスを指定します。

- *[Build Pool:]* このオプションを選択すると、デバイスの フォーマット作業を行ないます。なお、このオプションを選択すると、対象の デバイスに存在する全てのデータが失われます！

[fs:] タイプの場合

- *[target Path:]* VM ホストサーバ のファイルシステム内で、マウントする位置 (マウントポイント) を指定します。
- *[Format:]* デバイスのファイルシステム形式を指定します。既定値 auto (自動) で問題なく動作するはずです。
- *[Source Path:]* デバイスファイルのパスを指定します。/dev/sdX のような 形式ではなく、/dev/disk/by-* のような形式で 指定してください。これは前者のように指定してしまうと、デバイスの 接続状況によって名前が変わってしまう可能性 (たとえばデバイスを追加 したり削除したりした場合に、名前が変わってしまう可能性) があるためです。

[iscsi:] タイプの場合

VM ホストサーバ 上で下記のコマンドを入力すると、必要なデータを得ることができます:

```
iscsiadm --mode node
```

上記を実行すると、下記のような書式で iSCSI ボリュームの一覧が出力されます。太字で表記している箇所が必要な情報です:

IP_アドレス:PORT,TPGT ターゲット名_(IQN)

- *[Target Path:]* デバイスファイルが含まれている ディレクトリを指定します。/dev/disk/by-path (既定値) または /dev/disk/by-id のいずれかを 指定します。
- *[Host Name:]* iSCSI サーバのホスト名または IP アドレスを 指定します。

- `[Source Path:]` iSCSI のターゲット名 (IQN) を指定します。

`[logical:]` タイプの場合

- `[Target Path:]` 既存のボリュームグループを使用する場合、ここに既存のデバイスパスを入力します。新しい LVM ボリュームグループを作成 する場合は、`/dev` ディレクトリ内のデバイスファイルのうち、その時点で存在していないデバイス名を入力します。
- `[Source Path:]` 既存のボリュームグループを使用する場合は 何も入力しません。新しいボリュームグループを作成する場合は、ここにデバイスを 指定します。
- `[Build Pool:]` 新しいボリュームグループを作成する場合のみチェックを入れます。

`[mpath:]` タイプの場合

- `[Target Path:]` マルチパスへの対応は、利用可能な全ての マルチパスデバイスを利用して作成する場合に限られています。そのため、ここでは任意の文字列を指定します (何も指定しないと、XML パーサーが エラーを返します)。後で使用するようなことはありません。

`[netfs:]` タイプの場合

- `[target Path:]` VM ホストサーバ のファイルシステム上での マウントポイントを指定します。
- `[Format:]` ネットワークファイルシステムのプロトコルを 指定します。
- `[Host Name:]` ネットワークファイルシステムを提供する サーバの IP アドレスまたはホスト名を入力します。
- `[Source Path:]` サーバ上で公開しているディレクトリを 指定します。

`[scsi:]` タイプの場合

- `[Target Path:]` デバイスファイルを含むディレクトリを 指定します。`/dev/disk/by-path` (既定値) または `/dev/disk/by-id` のいずれかを指定します。

- [Source Path:] SCSI アダプタの名称を入力します。

注記: ファイル参照

リモートから操作している場合、[Browse] ボタンを押す ことによるファイルブラウザは利用できません。

- 4 最後に [Finish] を押すと、ストレージプールへの追加が 行なわれます。

7.1.2 ストレージプールの管理

Virtual Machine Manager のストレージマネージャでは、プール内にボリュームを作成することができる ほか、プール内のボリュームを削除することができます。そのほか、一時的に既存の ストレージプールを無効化したり、恒久的に削除したりすることもできます。なお、プールの基本設定を変更する作業は、SUSE ではサポートしていません。

7.1.2.1 プールの開始と停止、および削除

ストレージプールの機能は、リモートから管理している場合に VM ホストサーバ 上に配置 されているブロックデバイスを追加できるようにするため、用意されている機能です。特定のプールをリモートから一時的にアクセスできないようにするには、ストレージ マネージャの左下隅にある停止マークを押して [Stop] (停止) させてください。停止されたプールは [State: Inactive] (状態: 無効) として印が付けられ、一覧表示では灰色で表示されます。既定では 新しく作成したプールは、VM ホストサーバ の起動時に自動で開始するように ([On Boot]) なっています。

無効になっているプールを開始 ([Start]) し、リモートから 再度アクセスできるようにするには、ストレージマネージャの左下隅にある再生ボタン を押してください。

注記: プールの状態と接続されたボリュームの影響関係

プールから VM ゲスト に割り当てられたボリュームは、そのプールの状態 ([Active] (有効) または [Inactive] (無効)) に関わらず、常に利用可能な状態になります。プールの状態は、リモートからの 管理で、VM ゲスト にボリュームを割り当てられるかどうかだけに影響します。

プールに対して恒久的にアクセスできないようにするには、ストレージマネージャの左下隅にあるシュレッダー型のボタンを押すことで [Delete] (削除) を行なうこと

ができます。削除は無効化されたプールに対してのみ実施できます。削除の作業では VM ホストサーバ上で物理的に削除されたり消去されたりすることではなく、プールの設定だけが削除されます。ですが、特に LVM ボリュームグループベースのツールを削除するような場合には注意して作業を行なってください:

警告: ストレージプールの削除

ローカルの ファイルシステムディレクトリや ローカルパーティション、ローカルディスクをベースにしたプールの 削除は、その時点で VM ゲスト に割り当てられているボリュームに対しては効果がありません (以前と変わらず利用できます)。

一方、iSCSI や SCSI、LVM グループやネットワークで公開されているディレクトリの場合は、プールが削除されるとアクセス不可能になります。ボリュームそれ自身が 削除されなくても、VM ホストサーバではそれらの資源に対してアクセスする権利がなくなる ためです。

iSCSI/SCSI ターゲット上やネットワークで公開されているディレクトリでは、新しいプールを同じ設定で作成したり、ホスト側のシステムからマウントしたり アクセスしたりすることで、再度アクセスできるようになります。

LVM グループをベースにしたプールの場合には、LVM グループの設定が消去されるため、ホストシステム上では存在が消去されてしまいます。設定は復元することはできず、プール上に存在していた全てのボリュームは失われてしまいます。

7.1.2.2 ストレージプールへのボリュームの追加

Virtual Machine Manager では、マルチパスや iSCSI、SCSI のプールを除く全てのストレージプールに対して、ボリュームを作成することができます。これらのプール上のボリュームは LUN と等価なもので、libvirt から変更することはできません。

- 1 新しいボリュームは、ストレージマネージャを利用して作成することができるほか、VM ゲスト に対して新しいデバイスを追加する際にも作成することができます。いずれの場合でも、*[Storage Pool]* を選択して *[New Volume]* を選択します。
- 2 まずはイメージの *[Name]* (名前) を指定し、イメージの形式を選択します (なお、SUSE では raw, qcow2, qed イメージ形式をサポートしています)。イメージの形式は、LVM グループベースのプールの場合には設定不要です。

さらに *[Max Capacity]* (最大容量) を指定し、初期状態で 割り当てる領域サイズを指定します。これらの値に対して異なる値を設定すると、それは *スパース イメージファイル*と呼ばれる形式になり、必要に応じてサイズを拡張するタイプのボリュームを作成します。

3 あとは *[Finish]* を押すと、ボリュームの作成が始まります。

7.1.2.3 ストレージプールからのボリュームの削除

ボリュームの削除は、ストレージマネージャでのみ実施することができます。削除したいボリュームを選んで *[Delete Volume]* を選んでください。あとは *[Yes]* を押して確認を行なうと、実際の 削除が行なわれます。なお、この機能を利用する際には特に注意を払ってください！

警告: ボリューム削除の際の確認について

ボリュームの削除は、有効な VM ゲスト や無効な VM ゲスト で使用されているかどうかに関わらず行なうことができます。削除されたボリュームを復元する手段はありません。

VM ゲスト でボリュームが使用されているかどうかは、ストレージマネージャ内の *[Used By]* 列で判断することができます。

7.2 virsh を利用したストレージの管理

virsh を利用することで、コマンドラインからもストレージを 管理することができます。なお、SUSE ではストレージプールの作成機能はサポートしていません。そのため、本章ではプールの開始や停止、削除のほか、ボリューム 管理の機能に限定して説明しています。

プールやボリュームの管理に使用する virsh の全サブコマンドは、それぞれ *virsh help pool* または *virsh help volume* のコマンドを実行することで表示することができます。

7.2.1 プールとボリュームの一覧表示

現在利用可能な全てのプールを表示するには、下記のコマンドを実行します。無効化されているプールも表示したい場合は、*--all* オプションを追加してください：


```
virsh pool-list --details
```

特定のプールに関する詳細を表示するには、pool-info サブコマンドを利用します:

```
virsh pool-info プール
```

既定では、プールごとにボリュームを一覧表示できます。あるプールから提供されている全てのボリュームを表示するには、下記のコマンドを実行します:

```
virsh vol-list --details プール
```

現時点の virsh コマンドでは、特定のボリュームがゲストで 使用されているかどうかを判断することはできません。下記の手順を利用することで、VM ゲスト で使用されている全てのプールと、そこから提供されている ボリュームを表示することができます。

手順 7.1 VM ホストサーバ で使用されている全ストレージボリュームの一覧表示

- 1 まずは XSLT スタイルシートを下記の内容で作成します。たとえば ~/libvirt/guest_storage_list.xml などのファイル名で保存します:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text"/>
  <xsl:template match="text()"/>
  <xsl:strip-space elements="*" />
  <xsl:template match="disk">
    <xsl:text> </xsl:text>
    <xsl:value-of select="(source/@file|source/@dev|source/@dir)[1]" />
    <xsl:text> &#10;</xsl:text>
  </xsl:template>
</xsl:stylesheet>
```

- 2 下記のコマンドをシェル内で実行します。なお以下の例は、全てのゲストの XML 定義が既定の場所 (/etc/libvirt/qemu) に存在 している場合の例です。また、xsltproc プログラムは libxslt パッケージ内に あります。

```
SSHEET="$HOME/libvirt/guest_storage_list.xml"
cd /etc/libvirt/qemu
for FILE in *.xml; do
  basename $FILE .xml
  xsltproc $SSHEET $FILE
done
```

7.2.2 プールの開始と停止、および削除

virsh のサブコマンドを利用することで、プールの開始や 停止、削除などを行なうことができます。下記の例では、それぞれ *POOL* の部分をプール名または UUID に置き換えて 実行してください:

プールの停止

```
virsh pool-destroy POOL
```

注記: プールの状態と接続されたボリュームの影響関係

プールから VM ゲスト に割り当てられたボリュームは、そのプールの状態 (*[Active]* (有効) または *[Inactive]* (無効)) に関わらず、常に利用可能な状態になります。プールの状態は、リモートからの 管理で、VM ゲスト にボリュームを割り当てられるかどうかだけに影響します。

プールの削除

```
virsh pool-delete POOL
```

警告: ストレージプールの削除

ローカルの ファイルシステムディレクトリや ローカルパーティション、ローカルディスクをベースにしたプールの 削除は、その時点で VM ゲスト に割り当てられているボリュームに 対しては効果がありません (以前と変わらず利用できます)。

一方、iSCSI や SCSI、LVM グループやネットワークで公開されている ディレクトリの場合は、プールが削除されるとアクセス不可能になります。ボリュームそれ自身が削除されなくても、VM ホストサーバ ではそれらの資源に 対してアクセスする権利がなくなるためです。

iSCSI/SCSI ターゲット上やネットワークで公開されているディレクトリでは、新しいプールを同じ設定で作成したり、ホスト側のシステムからマウントしたり アクセスしたりすることで、再度アクセスできるようになります。

LVM グループをベースにしたプールの場合は、LVM グループの設定が消去されるため、ホストシステム上では存在が消去されてしまいます。設定は復元することはできず、プール上に存在していた全てのボリュームは失われてしまいます。

プールの開始

```
virsh pool-start POOL
```

プールの自動開始設定

```
virsh pool-autostart POOL
```

VM ホストサーバ を再起動した際、自動的に開始するようにマークしたプールだけを自動的に開始します。

プールの自動開始解除

```
virsh pool-autostart POOL --disable
```

7.2.3 ストレージプールへのボリュームの追加

virsh では 2 種類の方法でストレージプールを作成することができます。1 つは XML の定義から作成する方法で、vol-create と vol-create-from を利用して行ないます。もう 1 つは vol-create-as を利用してコマンドラインパラメータから作成する方法です。前者の方法は SUSE でサポートしていませんので、ここでは vol-create-as のサブコマンドについて説明します。

既存のプールにボリュームを追加するには、下記のコマンドを実行します:

```
virsh vol-create-as POOL ❶NAME❷ 12G --format❸raw/qcow2/qed❹ --allocation 4G❺
```

- ❶ ボリュームを追加すべき先のプール名を指定します。
- ❷ ボリューム名を指定します。
- ❸ イメージのサイズを指定します。この例では 12 ギガバイトを指定しています。それぞれ k,M,G,T の接尾辞を付けることで、キロバイト、メガバイト、ギガバイト、テラバイトの意味になります。
- ❹ ボリュームの形式を指定します。SUSE では現在、raw, qcow2, qed の各形式をサポートしています。
- ❺ 残りは任意指定のパラメータです。virsh の既定では、必要に応じてサイズの大きくなるスパースイメージファイルを作成します。このパラメータでは、割り当てるべき領域を指定しています (この例では 4 ギガバイトを指定しています)。それぞれ k,M,G,T の接尾辞を付けることで、キロバイト、メガバイト、ギガバイト、テラバイトの意味になります。

このパラメータを指定しないと、初期状態では領域を割り当てないスパースイメージファイルを作成することになります。また、非スパース型のボリュームを作成したい場合は、このパラメータをボリュームのサイズと一致 (この例では 12G) させてください。

7.2.3.1 既存のボリュームの複製

プールにボリュームを追加する際のもう 1 つの方法として、既存のボリュームを複製する、という方法があります。複製したボリュームは、元のボリュームと同じプール内に作成されます。

```
vol-clone 既存のボリュームの名前①新しいボリュームの名前② --pool POOL③
```

- ❶ 複製元の既存のボリューム名を指定します。
- ❷ 新しいボリューム名を指定します。
- ❸ 任意指定のパラメータです。libvirt ではプールを自動的に判断しますが、これがうまく行かない場合は、このパラメータを指定してください。

7.2.4 ストレージプールからのボリュームの削除

プールからボリュームを恒久的に削除するには、vol-delete のサブコマンドを利用します：

```
virsh vol-delete NAME --pool POOL
```

--pool は任意指定です。libvirt ではプールを自動的に判断しますが、これがうまく行かない場合は、このパラメータを指定してください。

警告: No Checks Upon Volume Deletion

ボリュームの削除は、有効な VM ゲスト や無効な VM ゲスト で使用されているかどうかに関わらず行なうことができます。削除されたボリュームを復元する手段はありません。

VM ゲスト でボリュームが使用されているかどうかを判断するには、手順 7.1「VM ホストサーバ で使用されている全ストレージボリュームの一覧表示」(79 ページ) をご覧ください。

7.3 virtlockd を利用したディスクファイルとブロックデバイスの施錠

ブロックデバイスやディスクファイルに対する施錠 (ロック) を設定することで、複数の VM ゲストからブロックデバイスやディスクファイルに書き込まれるような事態を避けることができます。それ以外にも、同じ VM ゲスト を 2 度起動してしまうような

事態や、別々の仮想マシンに対して同じディスクを追加してしまうような事態を避けることもできます。この仕組みにより、誤った設定によって 仮想マシンのディスクイメージを破壊するリスクを減らすことができます。

施錠処理は `virtlockd` と呼ばれる デーモンが制御します。このデーモンは `libvirtd` デーモンとは別に動作する ため、`libvirtd` がクラッシュした場合や、`libvirtd` を再起動したような 場合であっても、施錠を維持することができます。また、`virtlockd` 自身を更新したような場合にも、自分 自身を再実行することができますので、この場合にも施錠を維持することができます。このような仕組みが備わっていますので、`virtlockd` の更新を適用した場合であっても、VM ゲスト を再起動する必要は *ありません*。

7.3.1 施錠の有効化

仮想ディスクの施錠は、openSUSE の既定では有効化されていません。有効化して、再起動時にも自動的に開始するように設定するには、下記の手順を 実施します：

- 1 `/etc/libvirt/qemu.conf` ファイルを編集し、下記のように値を設定します：

```
lock_manager = "lockd"
```

- 2 下記のコマンドを実行し、`virtlockd` デーモンを起動します：

```
rcvirtlockd start
```

- 3 下記のコマンドを実行して、`libvirtd` を再起動します：

```
rclibvirtd restart
```

- 4 システムの起動時に自動的に `virtlockd` を開始するように設定します：

```
insserv virtlockd
```

7.3.2 施錠の設定

既定では、`virtlockd` はお使いの VM ゲスト に設定されたすべてのディスクを、自動的に施錠するように設定されています。また、既定の設定では "直接的な" 施錠領域を使用するように設定されています。これは、VM ゲストの `<disk>` デバイスに設定された、実際のファイルパスに対して施錠を行なう、という意味です。たとえば、VM ゲスト の `<disk>` デバイス内に下記のような設定が書かれている場合、`/var/lib/libvirt/images/my-server/disk0.raw` ファイルに対して `flock(2)` が直接呼び出されます：

```
<disk type='file' device='disk'>
  <driver name='qemu' type='raw' />
  <source file='/var/lib/libvirt/images/my-server/disk0.raw' />
  <target dev='vda' bus='virtio' />
</disk>
```

virtlockd の設定は、`/etc/libvirt/qemu-lockd.conf` ファイルを編集することで変更することができます。このファイルには、コメント欄に 詳しい説明が書いてありますので、こちらをお読みください。なお、変更した設定を反映するには、下記のようにして virtlockd を再読み込みさせる必要があります：

```
rcvirtlockd reload
```

注記: ディスク施錠の設定範囲について

SUSE Linux Enterprise 11 SP3 では、施錠は全体 (全ての仮想ディスク) に対して設定されます。個別のディスクに対して施錠の可否を設定する機能は、将来リリースされる 予定です。

7.3.2.1 間接施錠領域の有効化

virtlockd の既定の設定では、「直接」施錠領域を使用するようになっています。これは仮想 マシンの `<disk>` デバイス内に設定された、実際のファイルパスに対して 施錠を行なう、という設定です。実際のファイルパスが、必ずしもすべての ホストからアクセスできるような構成でないような場合は、virtlockd を「間接」施錠領域 を使用するよう設定することもできます。この設定では、間接施錠領域用のディレクトリ内に、ディスクファイルのパスから作られたハッシュファイルを作成して施錠を確立します。そのため、実際のディスクファイルのパスではなく、ハッシュファイルに対して施錠を設定することになります。間接 施錠領域は、ディスクファイルを含むファイルシステムが、`fcntl()` による施錠に対応していないような場合にも 有用な仕組みです。間接施錠領域は、`file_lockspace_dir` という設定で 指定することができます：

```
file_lockspace_dir = "/MY_LOCKSPACE_DIRECTORY"
```

7.3.2.2 LVM や iSCSI ボリュームに対する施錠の有効化

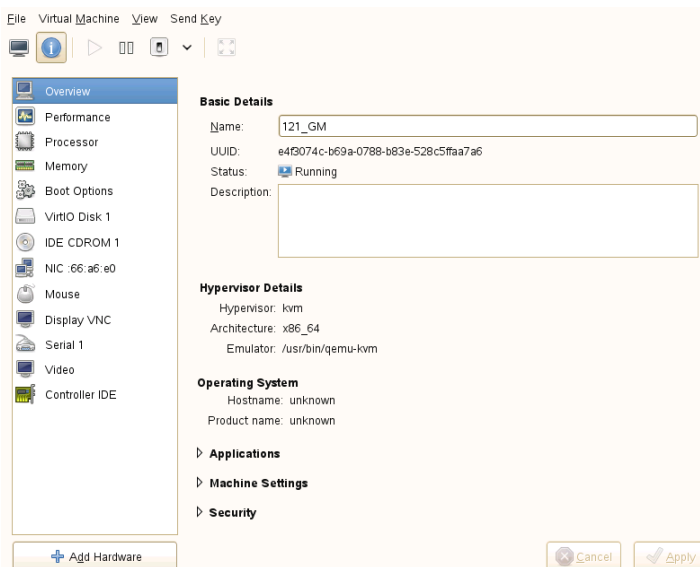
LVM や iSCSI ボリューム上に配置されていて、複数のホストで共有されている 仮想ディスクに対して施錠を実施したい場合は、パス (既定値) ではなく UUID で施錠を実施する必要があります。また、施錠領域のディレクトリは、ボリュームを共有する全てのホストからアクセス可能な、共有ファイルシステム 上に配置する必要があります。LVM や iSCSI をお使いの場合は、下記の オプションを指定してください：

```
lvm_lockspace_dir = "MY_LOCKSPACE_DIRECTORY"  
iscsi_lockspace_dir = "MY_LOCKSPACE_DIRECTORY"
```


仮想マシンの設定

Virtual Machine Manager の **〔詳細〕** ビューでは、VM ゲスト のすべての設定項目と ハードウェアの設定について、より詳しい情報が提供されています。このビューを利用することでゲストの設定を変更することができるほか、仮想ハードウェアを追加したり変更したりすることもできます。このビューにアクセスするには、Virtual Machine Manager でゲストのコンソールを開いてメニューから **〔ビュー〕** > **〔詳細〕** を選択するか、もしくはツールバー内の 青い情報アイコンを押します。

図 8.1 VM ゲストの [詳細] ビュー



8.1 シームレスなカーソル移動とカーソル移動の同期

マウスで VM ゲスト のコンソールを押すと、カーソルはコンソールウィンドウ内にキャプチャされ (閉じこめられ)、明示的に解放 (Alt + Ctrl を押すと解放することができます) されるまで外側に移動できなくなります。キー入力とマウスの移動をキャプチャ されないようにし、ホストとゲストの間を自由に移動できるようにするには、VM ゲスト にタブレットを追加してください。

タブレットを追加すると、ゲスト側でグラフィカル環境を使用している場合、VM ホストサーバと VM ゲスト の間でカーソルを同期することもできます。ゲスト側で タブレットを設定しない場合、ホストとゲストのカーソルが常に 1 つずつ 表示されます。

- 1 Virtual Machine Manager 内で VM ゲスト の項目をダブルクリックしてコンソールを開き、[View] > [Details] で [Details] ビューに切り替えます。
- 2 [ハードウェアの追加] を押してから [入力] を選択し、表示されたポップアップウィンドウ から [EvTouch USB Graphics Tablet] を選択 します。選択したあとは [Finish] で続行します。

- 3 タブレットの追加時にゲストが動作中であった場合は、次の再起動でタブレットを有効化するかどうかの確認が表示されます。この場合は **[Yes]** で続行してください。
- 4 VM ゲスト が動作中であった場合は再起動を行なうことで、動作中で なかった場合はそのまま起動することで、タブレットが利用できるようになります。

8.2 Virtual Machine Manager を利用した CD/DVD-ROM デバイスの追加

KVM では、VM ゲスト 内から CD または DVD-ROM を利用することができます。VM ゲスト から VM ホストサーバ 上にあるデバイスそのものに直接アクセスさせることができるほか、ISO イメージにアクセスさせることもできます。既存の CD や DVD から ISO イメージを作成したい場合は、dd コマンドを利用して 下記のように実行します:

```
dd if=/dev/CD_または_DVD_デバイス of=my_distro.iso bs=2048
```

VM ゲスト に CD/DVD-ROM デバイスを追加するには、下記の手順で行ないます:

- 1 Virtual Machine Manager 内から設定したい VM ゲスト を選択してダブルクリックし、コンソールを開いてから **[View]** > **[Details]** を選択し、**[Details]** 画面に移動します。
- 2 **[Add Hardware]** を押し、表示されたポップアップ ウィンドウから **[Storage]** を選択します。表示された画面は **[Forward]** で先に進めます。
- 3 **[Device Type]** では **[IDE CDROM]** を選択します。
- 4 **[Select Managed or Other Existing Storage]** を選択します。
 - 4a VM ホストサーバ 上の実際のデバイスを割り当てる場合は、**[Browse]** ボタンの隣にある項目に、VM ホストサーバ の CD/DVD-ROM デバイスパス (たとえば `/dev/cdrom`) を入力します。この項目では **[Browse]** ボタンを利用することでファイルブラウザを開き、**[Browse Local]** からデバイスを選択することもできます。なお、Virtual Machine Manager を VM ホストサーバ 上で起動した場合にのみ実際のデバイスを 割り当てることができます。

- 4b** 既存のイメージファイルをデバイスに割り当てる場合は、`[Browse]` を押して表示される画面で、ストレージプールからイメージを選択します。Virtual Machine Manager を VM ホストサーバ 上で起動している場合は、`[Browse Local]` を押すことで、ファイルシステム内からイメージの場所を選択することもできます。イメージを選択してファイルブラウザを閉じるには、`[Choose Volume]` を押します。
- 5** `[Forward]` を押して先に進み、設定内容を確認します。あとは `[Finish]`, `[Yes]`, `[Apply]` と押していくと、設定を適用することができます。
- 6** 新しく作成したデバイスを利用できるようにするには、VM ゲスト を再起動する必要があります。さらに詳しい情報は、8.4項「Virtual Machine Manager を利用したフロッピーディスクまたは CD/DVD-ROM メディアの取り出しと交換」(91 ページ) をお読みください。

8.3 Virtual Machine Manager を利用したフロッピーデバイスの追加

現時点では KVM ではフロッピーディスクイメージのみをサポートし、実際の フロッピーディスクドライブを利用する機能には対応していません。既存の フロッピーディスクからディスクイメージを作成するには、`dd` を使用して行ないます:

```
dd if=/dev/fd0 of=/var/lib/libvirt/images/floppy.img
```

中に何も入っていない空のフロッピーディスクイメージを作成するには、以下を実行します:

```
# RAW イメージの作成
dd if=/dev/zero of=/var/lib/libvirt/images/floppy.img bs=512 count=2880

# FAT でフォーマットする
mkfs.msdos -C /var/lib/libvirt/images/floppy.img 1440
```

お使いの VM ゲスト にフロッピーデバイスを追加するには、下記の手順で行ないます:

- 1** Virtual Machine Manager 内から設定したい VM ゲスト を選択してダブルクリックし、コンソールを開いてから `[View]` > `[Details]` を選択し、`[Details]` 画面に移動します。

- 2 [Add Hardware] を押し、表示されたポップアップ ウィンドウから [Storage] を選択します。表示された画面は [Forward] で先に進めます。
- 3 [Device Type] では [Floppy Disk] を選択します。
- 4 [Select Managed or Other Existing Storage] を選択して から [Browse] ボタンを押し、ストレージプールから既存の イメージを選択します。Virtual Machine Manager を VM ホストサーバ 上で起動している場合は、[Browse Local] を押すことで、ファイルシステム内から イメージの場所を選択することもできます。イメージを選択してファイル ブラウザを閉じるには、[Choose Volume] を押します。
- 5 [Forward] を押して先に進み、設定内容を確認します。あとは [Finish], [Yes], [Apply] と押していくと、設定を適用することができます。
- 6 新しく作成したデバイスを利用できるようにするには、VM ゲスト を再起動する 必要があります。さらに詳しい情報は、8.4項「Virtual Machine Manager を利用したフロッピーディスクまたは CD/DVD-ROM メディアの取り出しと交換」(91 ページ) をお読みください。

8.4 Virtual Machine Manager を利用したフロッピーディスクまたは CD/DVD-ROM メディアの取り出しと交換

VM ホストサーバ 側の実在する CD/DVD-ROM デバイスを利用するか、もしくは ISO イメージやフロッピーイメージを利用するかどうかに関わらず、VM ゲスト 側では メディアやイメージの交換を行なうことができます。交換を行なう場合には、ゲスト側であらかじめ disconnect (切断) をする必要があります。

- 1 Virtual Machine Manager 内から設定したい VM ゲスト を選択してダブルクリックし、コンソールを開いてから [View] > [Details] を選択し、[Details] 画面に移動します。
- 2 フロッピーまたは CD/DVD-ROM デバイスを選択し、[Disconnect] を押してメディアの「取り出し」を行ないます。
- 3 新しいメディアを「挿入」するには、[Connect] を押します。

- 3a VM ホストサーバ 側の実在する CD/DVD-ROM デバイスを利用している場合は、まず デバイス内のメディアを交換します (この場合、メディアを取り出す前に VM ホストサーバ 側でマウント解除が必要になります)。その後 *[CD-ROM or DVD]* を選択して、表示されたドロップダウン リストからデバイスを選択します。
- 3b ISO イメージを利用している場合は、*[ISO image Location]* を押してから *[Browse]* を押し、イメージを選択します。リモートから接続している場合は、既存のストレージプールからのみ選択することができます。
- 4 最後に *[OK]* を押して完了します。これで VM ゲスト から 新しいメディアにアクセスできるようになります。

8.5 Virtual Machine Manager を利用した PCI デバイスの追加

グラフィカルなツールである virt-manager を利用することで、ゲストに PCI デバイスを追加することができます。PCI デバイスを特定の VM ゲスト に割り当てると、再割り当てを行なわない限り、他の VM ゲスト では利用できなくなります。下記の手順では、USB コントローラを仮想化ゲストに割り当てするための 手順を説明しています。

- 1 Virtual Machine Manager で対象の VM ゲスト をダブルクリックしてコンソールを開き、*[View]* > *[Details]* を選択して *[Details]* ビューを表示させます。
- 2 *[Add Hardware]* を選択して、左側のペインで *[PCI Host Device]* の分類を選択します。すると、ウィンドウの 右側に、ホスト側の PCI デバイスの一覧が表示されます。
- 3 利用可能な PCI デバイスの一覧から、VM ゲスト に割り当てる USB コントローラを選択します。たとえば USB2 Enhanced Host Controller のような名前で掲載されているはずです。選択を行なったら *[Finish]* を押します。

ヒント

VM ゲスト が動作中の場合は、PCI デバイスを割り当てることはできません。動作中のマシンに対して PCI デバイスを割り当てようとすると、Virtual Machine Manager は 次回の VM ゲスト のシャットダウン時に、対象のデバ

イスを追加するかどうかを 尋ねます。ここで [Yes] (はい) を選択すると、次の VM ゲスト の再起動時にデバイスが割り当てられるようになります。

8.6 virsh を利用した PCI デバイスの追加

virsh を利用して VM ゲスト に対し、PCI デバイスを 占有させて割り当てるには、下記の手順を実施します:

1 まずは PCI デバイスを識別します。

VM ゲスト に提供する PCI デバイスを、virsh nodedev-list コマンドまたは lspci -n コマンドで識別します。

下記のコマンドを実行すると、利用可能な PCI デバイスのみを表示します:

```
virsh nodedev-list | grep pci
```

なお、PCI デバイスは下記のような文字列形式で区別されます (下記の例で、8086 は Intel アーキテクチャであることを 示す値で、**** は各デバイスに割り当てられる 4 桁の 16 進数値です):

```
pci_8086_****
```

表示された PCI のデバイス番号を覚えておきます。この番号は、下記の手順で必要となります。

2 ドメイン、バス、機能に対する情報収集を行ないます:

```
# virsh nodedev-dumpxml pci_8086_1d26
<device>
  <name>pci_8086_1d26</name>
  <parent>computer</parent>
  <driver>
    <name>ehci_hcd</name>
  </driver>
  <capability type='pci'>
    <domain>0</domain>
    <bus>0</bus>
    <slot>29</slot>
    <function>0</function>
    <product id='0x1d26'>Patsburg USB2 Enhanced Host Controller #1</product>
    <vendor id='0x8086'>Intel Corporation</vendor>
    <capability type='virt_functions'>
      </capability>
    </capability>
  </capability>
</device>
```

```
</capability>
</device>
```

- 3 VM ゲスト に割り当てる前に、ホストシステム側でデバイスの切り離しを行いません:

```
# virsh nodedev-detach pci_8086_1d26
Device pci_8086_1d26 detached
```

- 4 domain, bus, slot, function の値をそれぞれ 10 進数から 16 進数に変換します。このとき、'0x' という文字列を頭に付けて、16 進数である旨を表します。下記は bus = 0, slot = 29, function = 0 の場合での例です。それぞれの 16 進数値は、下記のようになります:

```
# printf %x 0
0
# printf %x 29
1d
```

上記の手順で判明した通り、bus と function の値は '0x00', slot の値は '0x1d' になります。

- 5 対象の VM ゲスト に対して virsh edit を実行し、下記のデバイス項目を <devices> セクション内に 追記します:

```
<hostdev mode='subsystem' type='pci' managed='no'>
  <source>
    <address domain='0x0000' bus='0x00' slot='0x1d' function='0x00' />
  </source>
</hostdev>
```

ヒント: 'managed' と 'unmanaged' の違いについて

libvirt は PCI デバイスを扱うにあたって、2 つの モード (それぞれ 'managed' or 'unmanaged' と呼ばれます) が用意されています。'managed' では、必要に応じて libvirt 側で、ドメインの起動時などに既存のドライバからの切り離しとデバイスのリセット、そして pci-stub への割り当てを行いません。ドメインが停止した場合やドメインからデバイスが切り離された際、libvirt は pci-stub からの切り離しと元のドライバへの再割り当てを行いません。それに対して 'unmanaged' では、ドメインへの割り当て前に実施すべきすべての作業から、割り当て完了後の作業までを、ユーザ側で実施しなければなりません。

今回の例では managed='no' オプションが設定されていますが、これは 'unmanaged' と同じ意味になります。つまり、自分自身で virsh nodedev-detach や virsh nodedev-attach コマンドを実行して、関連するドライバの

状態を制御する必要があります。デバイスを 'managed' に切り替える には、上記の例を managed=' yes' にして、残りの手順 (VM ゲスト の起動以外のすべて) を実施しないようにしてください。

- 6 VM ゲスト システム側で PCI デバイスを利用できるようになったら、ホストに対して利用を停止させます。まずはホスト側のシステムで、どのドライバが PCI デバイスを利用しているのかを調べます。

```
# readlink /sys/bus/pci/devices/0000¥:00¥:1d.0/driver
../../../../bus/pci/drivers/pci-stub
```

- 7 上記の例では、pci-stub ドライバが読み込まれています。このように表示されていれば、VM ゲスト を起動できるようになります。これで PCI デバイスが VM ゲスト に割り当てられます。

```
# virsh start sles11
```

ヒント

FLR (function level reset; 機能レベルリセット) や PM (power management; 電源管理) リセットに対応していない多機能型の PCI デバイスを利用している 場合は、VM ホストサーバ 側ですべての機能を切り離す必要があります。また、FLR や PM リセットに対応していない場合、セキュリティ上の理由でデバイス 全体をリセットしなければなりません。libvirt では、VM ホストサーバ やその他の VM ゲスト でデバイスの機能が利用されている場合、処理を拒否するようになっています。

VM ゲスト からデバイスの機能を安全に切り離すには、virsh nodedev-detach コマンドを利用します。

ヒント

お使いの PCI デバイスが 'managed' ではなく、PCI デバイスを制御しているドライバが pci-stub ではない場合、まずはデバイスから 切り離さなければなりません:

```
virsh nodedev-detach pci_8086_1d26
```

ヒント

また、お使いのホストで SELinux を動作させている場合、下記のように実行して 無効化する必要があります:

```
# setsebool -P virt_use_sysfs 1
```

処理が完了したら、仮想マシンを起動してください。

8.7 時計の設定

VM ゲスト で正しい時刻を設定することは、仮想化を利用する際の難題の 1 つ です。正しい時刻の設定は、ネットワークアプリケーションを利用する際、特に要求されるもので、VM ゲスト に対してライブマイグレーションを行なう 際にも、事前に行なっておかなければならない項目です。

ヒント: VM ホストサーバ における時刻同期

VM ホストサーバ 側でも、たとえば NTP などを使用 (詳しくは 第17章 *NTP を利用した時刻同期* (↑リファレンス)) をお読みください) して、正しい時刻への同期を強くお勧めします。

8.7.1 kvm_clock の使用

KVM では擬似仮想化クロックが提供されています。これは `kvm_clock` と呼ばれるドライバで、openSUSE のほか、SUSE Linux Enterprise Server 10 SP3 またはそれ以降のバージョン、もしくは RedHat Enterprise Linux 5.4 またはそれ以降のバージョンでサポートされています。可能であれば `kvm_clock` を使っておくことを強くお勧めします。

実行中の VM ゲスト で `kvm_clock` ドライバが読み込まれているかどうか、VM ゲスト 内から確認を行なうには下記の コマンドを実行します:

```
~ # dmesg | grep kvm-clock
[ 0.000000] kvm-clock: cpu 0, msr 0:7d3a81, boot clock
[ 0.000000] kvm-clock: cpu 0, msr 0:1206a81, primary cpu clock
[ 0.012000] kvm-clock: cpu 1, msr 0:1306a81, secondary cpu clock
[ 0.160082] Switching to clocksource kvm-clock
```

現在使用されている時刻源を確認するには、VM ゲスト 内で下記のコマンドを実行します。`kvm-clock` と入力されるべき項目です:

```
echo /sys/devices/system/clocksource/clocksource0/current_clocksource
```

重要: kvm-clock と NTP について

VM ゲスト 内で kvm-clock を使用している場合は、NTP による時刻同期を併用することはお勧めできません。VM ホストサーバ 側での NTP は、VM ゲスト 内での kvm-clock の使用に 関わらず、使用をお勧めします。

8.7.2 その他の時刻維持方法

擬似仮想化 kvm-clock ドライバは、SUSE Linux Enterprise Server 9 や Windows オペレーティングシステムに対しては提供されていません。Windows の場合は Windows 時刻サービスツール を利用して同期を行なってください (詳しくは <http://technet.microsoft.com/ja-jp/library/cc773263%28WS.10%29.aspx> をお読みください)。

SUSE Linux Enterprise Server 9 SP4 で時刻維持を行ないたい場合は、システムの起動時に特殊な 起動パラメータを設定することで実現できます:

32 ビットカーネルの場合:clock=pmtmr

64 ビットカーネルの場合:ignore_lost_ticks

VM ゲスト の管理

9.1 VM ゲスト の移行

仮想化を利用する大きな利点のうちの 1 つに、VM ゲスト の可搬性があげられます。VM ホストサーバ をメンテナンスなどの理由でダウンさせなければならないような場合や、ホストの負荷が大きすぎるような場合、ゲストを他の VM ホストサーバ に対して簡単に 移動することができます。KVM や Xen では「ライブ」マイグレーションにも対応していて、VM ゲスト を常に提供しながら移行作業を 実施することができます。

VM ゲスト を他の VM ホストサーバ に正しく移行できるようにするためには、下記の要件が満たされていることを確認してください：

- 現行のホストと移行先のホストの両方で、プロセッサの製造元が同じである こと (Intel または AMD)。
- VM ゲスト が 64 ビット版のオペレーティングシステムを利用している場合、移行先のホストでも 64 ビット版のオペレーティングシステムが動作している こと (32 ビット版の VM ゲスト であれば、32 ビットと 64 ビットの両方の ホストに移行できます)。
- 両方のホストから (たとえば NFS や SCSI を介して) ストレージデバイスに アクセスできること、および両方のマシンでストレージプールが設定されている こと (詳しくは 第7章 *ストレージの管理* (69 ページ) を参照)。これは CD-ROM や フロッピーディスクのイメージについても同様に、移動する 時点で必要なものであれば、必ず両方のホストからアクセスできなければ なりません (不要であれば

移動前に切断することもできます。詳しくは 8.4項「Virtual Machine Manager を利用したフロッピーディスクまたは CD/DVD-ROM メディアの取り出しと交換」(91 ページ) をお読みください。

- libvirtd が両方の VM ホストサーバ で動作していて、互いに相手方の libvirt に対してリモート接続を行なうことができること。詳しくは 6.2項「リモート接続の設定」(55 ページ) をお読みください。
- 移行先のホストでファイアウォールが動作している場合は、移行中に必要な ポートを開く必要があります。移行処理時に特にポートを指定しない場合、libvirt は 49152 から 49215 までのいずれかのポートを利用します。この 範囲のポート全体 (推奨) または移行中に指定したポートについて、**移行先のホスト** のファイアウォールで開かれていることを 確認してください。
- 移行元と移行先のホストが同じサブネット内に存在していること。異なる サブネット内に存在していると、移行処理が正しく動作しない場合があります。
- 移行先のホストで、同じ名前で稼働しているか、もしくは一時停止している VM ゲスト が存在しないこと。同じ名前でシャットダウン済みのゲストが 存在した場合は、設定が上書きされます。
- VM ゲスト を移行する場合、既定の CPU モデル (qemu64) のみを使用することがあります。
- SATA ディスクのデバイスタイプは、移行することができません。
- ファイルシステムのパススルー機能は、移行処理には対応していません。

9.1.1 virt-manager を利用した移行

Virtual Machine Manager を利用した VM ゲスト を移行する場合、どのマシンで起動しても移行 作業を行なうことができます。Virtual Machine Manager を移行元や移行先のホストで動作させて 移行することができるほか、別のホストから実行してもかまいません。別の ホストから実行する場合は、移行元と移行先の両方に対して、リモートからの 接続を行なえる必要があります。

- 1 Virtual Machine Manager を起動して移行元と移行先の両方のホストに対して、接続を確立します。Virtual Machine Manager を移行元や移行先のホストで起動しない場合は両方のホストに対して 接続を行ないます。
- 2 移行対象の VM ゲスト を選んでマウスの右ボタンを押し、[*Migrate*] (移行) を選択します。なお、対象のゲストが 実行中か一時停止の状態になっていることを

確認してください。シャットダウン されている場合、移行作業を行なうことができません。

- 3 VM ゲスト の移行先を [New Host] (新しいホスト) を選びます。移行先として選択したいホストが表示されない場合は、そのホストに対する接続が 確立されているかどうかをご確認ください。

既定では「ライブ」マイグレーションが実行されます。VM ゲスト を移行中に一時停止させる「オフライン」マイグレーションを行ないたい場合は、[*Migrate offline*] (オフライン状態で マイグレーションを行なう) を選択します。

- 4 最後に [Migrate] を押すと、既定のポートと帯域幅を利用して 移行処理が始まります。

これらの既定値を変更するには、[*Advanced Options*] (高度なオプション) の三角形を押して、高度なオプションを設定してください。ここでは移行先のホストの [Address] (アドレス。IP アドレス またはホスト名) を設定することができるほか、ポート番号 ([Port]) や帯域幅 ([Bandwidth]; メガビット毎秒 [Mbps]) を設定することができます。ポートを設定した場合、[Address] にも 値を設定しなければなりません。帯域幅の指定は任意です。

- 5 移行作業が完了すると [Migrate] (移行) ウィンドウが閉じ、VM ゲスト が Virtual Machine Manager 内の新しい移行先のホスト内に表示されるようになります。元の VM ゲスト についても移行元のホスト内に残ります (シャットダウン状態に なります)。

9.1.2 virsh を利用した移行

virsh migrate を利用して VM ゲスト を 移行するには、VM ホストサーバ に対する直接的または間接的なシェルアクセスが必要です。これは、このコマンドを実行できるのはホスト側だけであるためです。基本的に移行作業は下記のようなコマンドを入力して行ないます:

```
virsh migrate [オプション] 仮想マシンの ID または名前 接続 URI [--migrateuri tcp://ホスト:ポート]
```

下記では元も重要なオプションについて説明しています。完全な一覧を取得したい場合は、virsh help migrate を実行してください。

--live

ライブマイグレーションを実行します。これを指定しない場合、VM ゲスト は移行作業中に一時停止します。

--suspend

オフラインによるマイグレーションを実行し、移行先のホストでは VM ゲストの再起動を行ないません。

--persistent

既定では移行された VM ゲストは一時的な存在で、それらの設定はシャットダウンされたときにホスト上から削除されます。このスイッチを指定することで、移行作業を恒久化させることができますようになります。

--undefinesource

これを指定した場合、移行が問題なく完了したあとで移行元の VM ゲストの設定を削除します (ただし、削除するゲストで利用していた仮想ディスクは *削除されません*)。

下記は mercury.example.com を移行元、jupiter.example.com を移行先のシステムとして利用し、移行する VM ゲストは ID が 37、名称が opensuse11 である場合の例です。

既定のパラメータを利用したオフラインマイグレーション

```
virsh migrate 37 qemu+ssh://tux@jupiter.example.com/system
```

既定のパラメータを利用した一時的なライブマイグレーション

```
virsh migrate --live opensuse11 qemu+ssh://tux@jupiter.example.com/system
```

恒久的なライブマイグレーション; 移行元の VM 設定を削除する場合

```
virsh migrate --live --persistent --undefinesource 37 ¥  
qemu+tls://tux@jupiter.example.com/system
```

ポート 49152 を利用するオフラインマイグレーション

```
virsh migrate opensuse11 qemu+ssh://tux@jupiter.example.com/system ¥  
--migrateuri tcp://@jupiter.example.com:49152
```

注記: 一時的なマイグレーションと恒久的なマイグレーション

既定では `virsh migrate` を実行すると、宛先のホストで VM ゲストに対する一時的なコピーを作成します。元のゲスト設定のシャットダウン版は、元のホスト上に残ります。一時的なマイグレーションの場合、サーバがシャットダウンすると削除されます。

宛先のホストで恒久的なコピーを作成するには、`--persistent` オプションを使用します。この場合も元のゲスト設定に対するシャットダウン版が、元のホスト上

に残ります。--persistent と共に --undefinesource を指定すると、宛先のホスト上で恒久的な コピーを作成し、元のホスト上にあった設定を削除します。

--persistent 無しで --undefinesource を指定することはお勧めできません。これは宛先のホストでゲストをシャット ダウンしてしまうと、双方で VM ゲスト の設定が消えてしまうためです。

9.2 監視

9.2.1 Virtual Machine Manager を利用した監視

Virtual Machine Manager を起動して VM ホストサーバ に接続したあとは、すべてのゲストに対して CPU の使用率が表示されるようになります。

CPU 以外にも、このツールではディスクやネットワークの使用率を取得することができます。なお、これらを取得する際には *[Preferences]* (設定) を有効にしなければなりません:

- 1 *virt-manager* を起動します。
- 2 *[Edit]* > *[Preferences]* を選択します。
- 3 *[General]* から *[Stats]* にタブを切り替えます。
- 4 *[Disk I/O]* と *[Network I/O]* にそれぞれチェックを入れます。
- 5 必要であれば、更新間隔や履歴内に保持するサンプル数などを指定します。
- 6 *[Preferences]* ダイアログを閉じます。
- 7 *[View]* > *[Graph]* を選択すると、グラフを表示することができます。

上記を行なうことで、ディスクとネットワークの統計情報が Virtual Machine Manager のメイン ウィンドウ内にも表示されるようになります。

より正確なデータは、VNC ウィンドウから取得することができます。VNC ウィンドウの開き方については、5.2項「グラフィカルコンソールの表示」(36 ページ)をお読みください。ツールバーから *[Details]* を選択するか、*[View]* メニューを選択してください。統計情報は 左側のツリーメニュー内の *[Performance]* 項目から表示することができます。

9.2.2 kvm_stat を利用した監視

KVM の性能イベントを追跡するには、kvm_stat を 利用することができます。このコマンドは /sys/kernel/debug/kvm を監視するため、debugfs をマウントしておく必要があります。openSUSE では特に設定しない限りマウントされます。マウントされていない場合は、下記のコマンドを実行してマウントしてください：

```
mount -t debugfs none /sys/kernel/debug
```

kvm_stat は 3 種類のモードを利用することができます：

```
kvm_stat                # 1 秒ごとに更新
kvm_stat -l             # 1 秒分のスナップショット
kvm_stat -l > kvmstats.log # ログ形式で 1 秒ごとに更新する。
                        # このログは表計算プログラムなどに取り込むことができる
```

例 9.1 kvm_stat の出力例

```
kvm statistics
```

efer_reload	0	0
exits	11378946	218130
fpu_reload	62144	152
halt_exits	414866	100
halt_wakeup	260358	50
host_state_reload	539650	249
hypercalls	0	0
insn_emulation	6227331	173067
insn_emulation_fail	0	0
invlpg	227281	47
io_exits	113148	18
irq_exits	168474	127
irq_injections	482804	123
irq_window	51270	18
largepages	0	0
mmio_exits	6925	0
mmu_cache_miss	71820	19
mmu_flooded	35420	9
mmu_pde_zapped	64763	20
mmu_pte_updated	0	0
mmu_pte_write	213782	29
mmu_recycled	0	0

mmu_shadow_zapped	128690	17
mmu_unsync	46	-1
nmi_injections	0	0
nmi_window	0	0
pf_fixed	1553821	857
pf_guest	1018832	562
remote_tlb_flush	174007	37
request_irq	0	0
signal_exits	0	0
tlb_flush	394182	148

これらの値の解釈方法について、詳しくは <http://clalance.blogspot.com/2009/01/kvm-performance-tools.html> をお読みください。

パート III. QEMU を利 用した仮想マシンの管理

QEMU の概要

QEMU は高速な複数プラットフォーム対応のオープンソース型エミュレータで、多くのハードウェアアーキテクチャを擬似することができます。QEMU は お使いのシステム (VM ホストサーバ) 上で、何も修正を施さずにオペレーティング システム (VM ゲスト) を稼働させることができます。

QEMU はデバッグ目的でも利用することができます。仮想マシンは必要なタイミングで 停止させることができるほか、状態の検査や保存、復元を行なうこともできます。

QEMU は下記のような部品から構成されています:

- プロセッサのエミュレータ (x86, s390x, PowerPC, Sparc など)
- デバイスのエミュレータ (グラフィックカード、ネットワークカード、ハードディスクドライブ、マウスなど)
- エミュレートされたデバイスから関連するホスト側のデバイスに接続する際に使用する、汎用デバイス
- エミュレートされたマシンの説明 (PC, Power Mac など)
- デバッグ
- エミュレータとやりとりを行なうためのユーザインターフェイス

仮想化ソリューションという意味では、QEMU は KVM のカーネルモジュールと 合わせて動作させることができます。VM ホストサーバ と VM ゲスト が同じアーキテ

クチャの ハードウェアであった場合、QEMU は KVM による高速化によるメリットを生かすこと ができるようになります。

ゲストのインストール

仮想マシンは、データと仮想マシンを定義するオペレーティングシステムファイルから構成されています。仮想マシンは VM ホストサーバ内で動作し、制御されます。この章では、仮想マシンのインストールについて一般的な手順を説明しています。

仮想マシンを作成する前に、下記の内容にご注意ください：

- 自動インストールファイル (AutoYaST, NetWare® レスポンスファイル, RedHat Kickstart など) を利用してインストールしたい場合、ホストマシン側のサーバ内で該当のファイルを、ホストマシンのサーバ上にダウンロードしておくか、もしくはネットワークファイルシステム経由でアクセスできるようにしておく必要があります。
- スパースイメージファイル (表示上の容量ではなく、実際に使用している部分の容量で領域を確保するタイプのファイル) を作成する場合は、作成するパーティションに十分な容量があるかどうか、よくご確認ください。ゲスト側のシステムではホスト側のディスク領域を確認したりすることはありませんので、ホスト側のパーティションに容量不足が発生すると、ゲスト側では書き込みエラーとして報告され、ゲストシステム側で保存していたデータを失ってしまうことになります。
- NetWare および OES Linux 仮想マシンを作成する場合は、作成するそれぞれの仮想マシンに対して固定の IP アドレスが必要となります。
- Open Enterprise Server (OES) 2 Linux をインストールしようとしている場合は、OES 2 Linux ソフトウェアのネットワークインストールソースが必要となります。

さらなる要件については、各オペレーティングシステムのインストール関連のマニュアルをお読みください。

11.1 qemu-kvm を利用した基本インストール

virt-manager や vm-install など、libvirt ベースのツールでは仮想マシンを設定したり管理したりするのに便利なインターフェイスを提供しています。これらはいずれも qemu-kvm に対するラッパー (ラッピングするもの) として動作します。ただし、libvirt ベースのツールを全く使用せず、qemu-kvm を直接使用することも可能です。

警告

qemu-kvm で作成した仮想マシンは、libvirt ベースのツールでは表示することができません。

下記の例では、qemu-kvm を使用することで 例4.1「vm-install を使用したコマンドラインからの対話型セットアップ」(28 ページ) での例と同じ パラメータで仮想マシンを設定します。コマンドについての詳しい説明は、関連する マニュアルページをお読みください。

仮想化された環境下で動作させたいシステムについて、そのイメージを作成していない 場合は、インストールメディアからイメージを作成する必要があります。このような 場合は、ハードディスクイメージを作成し、インストールメディアやメディアそれ自身の イメージを用意する必要があります。

まずは qemu-img でハードディスクを作成します。

```
qemu-img create① -f raw② /images/sles11/hda③ 8G④
```

- ① サブコマンド create では、qemu-img に対して 新しいイメージを作成するように指示しています。
- ② -f パラメータでディスクの形式を指定しています。
- ③ イメージファイルのフルパスを指定しています。
- ④ イメージのサイズ、ここでは 8 GB を指定しています。イメージはスパースファイルと呼ばれる形で作成され、ディスク内にデータが蓄えられていくたびに実容量が拡大していく形になっています。ここで指定するサイズは、イメージファイルの最大サイズを 指定しています。

1 つ以上のハードディスクイメージを作成したら、qemu-kvm を利用して仮想マシンの設定を行ない、インストールシステムを起動します。

```
qemu-kvm -name "sles11" ❶ -M pc-0.12 ❷ -m 768 ❸ ¥  
-smp 2 ❹ -boot d ❺ ¥  
-drive file=/images/sles11/hda,if=virtio,index=0,media=disk,format=raw ❻ ¥  
-drive file=/isos/SLES-11-SP1-DVD-x86_64-GM-DVD1.iso,index=1,media=cdrom ❼ ¥  
-net nic,model=virtio,macaddr=52:54:00:05:11:11 ❽ ¥  
-vga cirrus ❾ -balloon virtio ❿
```

- ❶ ウインドウキャプションに表示されるほか、VNC サーバの名前としても使用される 仮想マシン名です。名前はシステム内で唯一のものでなければなりません。
- ❷ マシンの種類 ([*Standard PC*], [*ISA-only PC*], [*Intel-Mac*] など) を指定しています。ここに指定できるパラメータの一覧を得るには、`qemu-kvm -M ?` を実行してください。pc-0.12 は既定の [*Standard PC*] の意味です。
- ❸ 仮想マシンに割り当てる最大メモリ量を指定しています。
- ❹ 2 プロセッサの SMP システムであることを指定しています。
- ❺ 起動順序を指定しています。指定できる値はそれぞれ、a, b (フロッピーディスク 1 または 2), c (最初のハードディスク), d (最初の CDROM), n から p まで (ネットワークアダプタ 1 から 3 に対応する イーサネット起動) です。既定値は c です。
- ❻ 最初の (index=0) ハードディスクを指定しています。raw 形式で、擬似仮想化 (if=virtio) ドライブとしてアクセスします。
- ❼ 2 番目の (index=1) イメージドライブは、CD-ROM として動作するものを指定しています。
- ❽ 擬似仮想化 (model=virtio) のネットワークアダプタを、MAC アドレス 52:54:00:05:11:11 で指定しています。ここで指定する値はネットワーク内で唯一のものを設定してください。そうでないと、ネットワーク上での通信に問題が発生する可能性があります。
- ❾ グラフィックカードを指定しています。*none* を指定した場合はグラフィックカードが無効化されます。
- ❿ 動的にメモリ量を変更できるようにするため、擬似仮想化バルーンデバイスを設定しています (最大メモリ量については -m パラメータで 指定します)。

ゲスト側のオペレーティングシステムのインストールが終わったら、CD-ROM デバイスの 指定を行なう必要はなくなり、簡単にシステムを起動できるようになります:

```
qemu-kvm -name "sles11" -M pc-0.12 -m 768 ¥  
-smp 2 -boot c ¥  
-drive file=/images/sles11/hda,if=virtio,index=0,media=disk,format=raw ¥  
-net nic,model=virtio,macaddr=52:54:00:05:11:11 ¥  
-vga cirrus -balloon virtio
```

11.2 qemu-img を利用したディスク管理

前の章 (11.1 項「qemu-kvm を利用した基本インストール」(112 ページ)) では、ハードディスクのイメージを作成するのに `qemu-img` コマンドを使用してきましたが、本コマンドは汎用的なディスクイメージ操作にも使用することができます。この章では、ディスクイメージを柔軟に管理する際に便利な、各種の `qemu-img` サブコマンドを紹介しています。

11.2.1 qemu-img 実行時の一般的な情報

`qemu-img` は (zypper のように) サブコマンドを使用します。各サブコマンドは異なる種類のオプションを認識する仕組みになっていて、それらのうちのいくつかは複数のサブコマンドで使用できる 汎用オプション、それ以外はサブコマンド内で独自のものになっています。利用可能なオプションを確認するには、`qemu-img` のマニュアルページ (`man 1 qemu-img`) をご覧ください。`qemu-img` は一般に、下記のような書式で記述します:

`qemu-img` サブコマンド [オプション]

サブコマンドには下記のようなものがあります:

`create`

ファイルシステム上に新しいディスクイメージを作成します。

`check`

既存のディスクイメージにエラーがないかどうか、確認を行ないます。

`convert`

既存のディスクイメージを、異なる形式に変換します。

`info`

指定したディスクイメージに関する情報を表示します。

`snapshot`

既存のディスクイメージに対するスナップショットを管理します。

`commit`

既存のディスクイメージに対して、変更点を適用します。

rebase

既存のイメージから、新しいベースイメージを作成します。

resize

既存のイメージのサイズを増やしたり、減らしたりします。

11.2.2 ディスクイメージの作成や変換、チェック

この章では、ディスクイメージの作成方法や状態の確認のほか、ディスクイメージの形式を一方から他方に変換する手順、および特定のディスクイメージについて 詳細な情報を取得する手順について説明しています。

11.2.2.1 qemu-img create

VM ゲスト 側のオペレーティングシステムで使用する新しいディスクイメージを 作成するには、qemu-img create コマンドを使用します。下記のような書式で実行します:

qemu-img create -f 種類① -o オプション② ファイル名③ サイズ④

- ① 作成するイメージの種類を指定します。対応しているイメージの種類を 表示するには、qemu-img -h を 実行し、出力の最後の行をお読みください。
- ② イメージの種類に対して追加のオプションをコマンドラインで指定することができます。このようなオプションは -o で指定してください。raw イメージの場合は size オプション のみがオプションとして存在していて、たとえば -o size=8G のように記述すると、コマンドの終わりにサイズを追加することなくサイズを設定することができます。
- ③ 作成するディスクイメージのパスを指定します。
- ④ 作成するディスクイメージのサイズを指定します (-o size=<サイズ> オプションで指定していない場合)。ここでは必要に応じて接尾辞を付けることができ、それぞれ K (キロバイト), M (メガバイト), G (ギガバイト), T (テラバイト) を指定することができます。

たとえば sles11sp1.raw という新しいディスクイメージを、/images ディレクトリ以下に最大 4 GB として作成する場合、下記のようなコマンドになります:

```
tux@venus:~> qemu-img create -f raw -o size=4G /images/sles11sp1.raw
Formatting '/images/sles11sp1.raw', fmt=raw size=4294967296
```

```
tux@venus:~> ls -l /images/sles11sp1.raw
-rw-r--r-- 1 tux users 4294967296 Nov 15 15:56 /images/sles11sp1.raw
```

```
tux@venus:~> qemu-img info /images/sles11sp1.raw
image: /images/sles11sp1.raw
file format: raw
virtual size: 4.0G (4294967296 bytes)
disk size: 0
```

上記のとおり、新しく作成したイメージの *見た目* (virtual) 上のサイズは 4 GB ですが、実際にディスク領域を占有しているサイズ (disk size) は 0 になっています。これは、まだ何もデータが書き込まれていない ためです。

11.2.2.2 qemu-img convert

ディスクのイメージ形式を変換するには、`qemu-img convert` コマンドを使用します。QEMU でサポートしているイメージ形式について、一覧を取得するには、`qemu-img -h` を実行し、出力の最後の行をお読みください。コマンドは下記のように記述します:

```
qemu-img convert -c❶ -f 入力種類❷ -O 出力種類❸ -o オプション❹ 入力ファイル名❺ 出力ファイル名❻
```

- ❶ 目的のディスクイメージに対して、圧縮機能を適用します。qcow と qcow2 形式のみが圧縮をサポートしています。
- ❷ 変換元のディスクイメージの形式を指定します。多くの場合自動検出が行なわれるため、省略が可能です。
- ❸ 変換先のディスクイメージの形式を指定します。
- ❹ 変換先のイメージ形式に対して、オプションを指定する項目です。-o ? を指定すると、変換先のイメージ形式で対応している オプションの一覧を表示することができます。
- ❺ 変換元のディスクイメージファイルを指定します。
- ❻ 変換先のディスクイメージファイルを指定します。

```
tux@venus:~> qemu-img convert -O vmdk /images/sles11sp1.raw ¥
/images/sles11sp1.vmdk
```

```
tux@venus:~> ls -l /images/
-rw-r--r-- 1 tux users 4294967296 16. 10.50 sles11sp1.raw
-rw-r--r-- 1 tux users 2574450688 16. 10.18 sles11sp1.vmdk
```

選択した変換先のイメージ形式について、これに対応するオプションの一覧を表示するには、下記のコマンドを入力します (なお vmdk をお使いの イメージ形式に合わせてください):

```
tux@venus:~> qemu-img convert -O vmdk /images/sles11sp1.raw ¥
/images/sles11sp1.vmdk -o ?
Supported options:
size                Virtual disk size
```

backing_file	File name of a base image
compat6	VMDK version 6 image
subformat	VMDK flat extent format, can be one of {monolithicSparse ¥ (default) monolithicFlat twoGbMaxExtentSparse twoGbMaxExtentFlat}
scsi	SCSI image

11.2.2.3 qemu-img check

既存のディスクイメージについて、エラーがないかどうかを確認するには、`qemu-img check` コマンドを使用します。なお、全ての ディスクイメージの形式で、この機能に対応しているというわけではないことに ご注意ください。下記のような書式で実行します:

`qemu-img check -f 形式① ファイル名②`

- ❶ チェック対象のディスクイメージの形式を指定します。多くの場合自動検出が行なわれるため、省略が可能です。
- ❷ チェック対象のディスクイメージファイルを指定します。

何もエラーが検出されない場合、コマンドは何も出力を返しません。それ以外の場合、エラーの種類とエラー数がそれぞれ表示されます。

```
tux@venus:~> qemu-img check -f qcow2 /images/sles11sp1.qcow2
ERROR: invalid cluster offset=0x2af0000
[...]
ERROR: invalid cluster offset=0x34ab0000
378 errors were found on the image.
```

11.2.2.4 既存のディスクイメージのサイズ拡張

新しいイメージを作成する場合は、イメージを作成する前に、その最大サイズを指定しなければなりません (詳しくは 11.2.2.1 項「`qemu-img create`」(115 ページ)をお読みください)。また、VM ゲスト をインストールし実行したあとで、初期 設定したサイズでは不足し、さらなる領域を割り当てる必要が発生する場合があります。

既存のディスクイメージのサイズを 2 ギガバイトだけ拡張するには、下記の手順を行ないます:

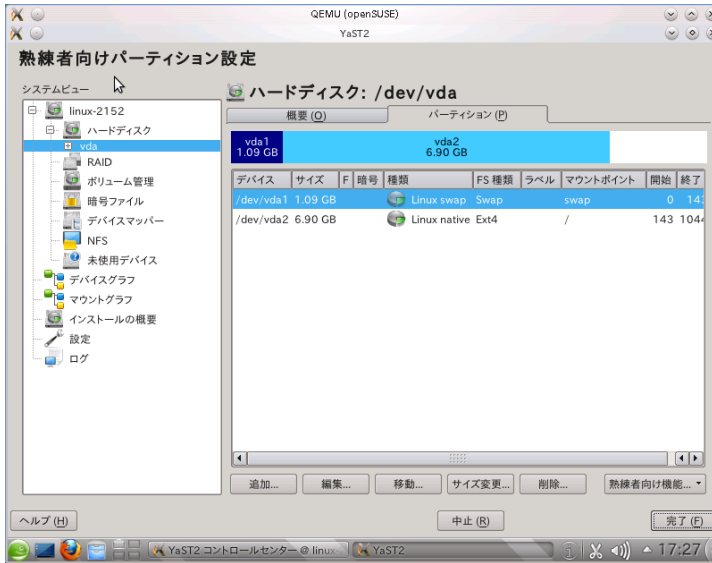
```
qemu-img resize /images/sles11sp1.raw +2GB
```

注記

サイズ変更は raw ディスクイメージに対してのみ実施することができます。その他のイメージ形式のサイズを変更するには、サイズ変更の前に `qemu-img convert` で raw 形式に 変換する必要があります。

これにより、ディスク内には最後のパーティション以降に 2 ギガバイトほどの 空き領域が生まれます。あとは既存のパーティションのサイズ変更を行ったり、新しいパーティションを追加したりすることができます。

図 11.1 ゲスト側の YaST パーティション設定における新規 2GB パーティション



11.2.3 qemu-img による仮想マシンのスナップショット管理

仮想マシンのスナップショットとは、VM ゲスト が実行中の状態について、それを 保存するための機能です。スナップショットにはプロセッサ (CPU) の状態のほか、メモリ (RAM) やデバイス、全ての書き込み可能なディスクについての情報が 含まれます。

スナップショットは、お使いの仮想マシンを特定の状態で保存しておきたい場合に 便利な 機能です。たとえば仮想サーバ内でネットワークサービスを設定したような 場合、その 状態で仮想マシンをすぐ開始して、その時点の状態に戻したいような 場合 が考えられます。それ以外にも、仮想マシンの電源を落としたタイミングでスナップ ショットを作成して それをバックアップとし、VM ゲスト に対して実験的な作業を行 なう前の準備とする 場合にも便利なものです。この章では、特に後者のような 場合 を想定して説明しています。前者については 第13章 *QEMU モニタを利用した仮 想マシンの管理* (157 ページ) で説明しています。

スナップショットを使用するには、お使いの VM ゲスト に少なくとも 1 台以上の 書き込み可能なハードディスクイメージが、qcow2 形式が必要です。このデバイスは通常、1 台目の仮想ハードディスクとして使用します。

仮想マシンのスナップショットは savevm コマンドで作成するもので、これは対話的な QEMU モニタから実行することができます。また、それぞれの スナップショット には 'タグ' を設定することができ、スナップショットの識別が 簡単に行なえるようになっています。QEMU モニタについて、詳しくは 第13章 *QEMU モニタを利用した仮想マシンの管理* (157 ページ) をお読みください。

qcow2 ディスクイメージにスナップショットを保存すると、qemu-img snapshot コマンドで内容を確認することができます。

警告

仮想マシンが動作している間は、qemu-img snapshot コマンドで 仮想マシンのスナップショットを作成したり削除したりすることは行なわないでください。これを行なってしまうと、仮想マシンの状態を保持しているディスクイメージが 壊れてしまう場合があります。

11.2.3.1 既存のスナップショットの表示

ディスクイメージで指定するディスクイメージ内に保存されている 全てのスナップショットを表示するには、qemu-img snapshot -l ディスクイメージを使用します。こちらのコマンドは、VM ゲスト が動作中でも実行できます。

```
tux@venus:~> qemu-img snapshot -l /images/sles11sp1.qcow2
```

Snapshot list:

ID ^❶	TAG ^❷	VM SIZE ^❸	DATE ^❹	VM CLOCK ^❺
1	booting	4.4M	2010-11-22 10:51:10	00:00:20.476
2	booted	184M	2010-11-22 10:53:03	00:02:05.394
3	logged_in	273M	2010-11-22 11:00:25	00:04:34.843
4	ff_and_term_running	372M	2010-11-22 11:12:27	00:08:44.965

- ❶ スナップショットの識別番号です。通常は自動的に番号が増えていきます。
- ❷ スナップショットの識別文字列です。これは識別番号を人間にとって 読みやすい形式にしたものです。
- ❸ スナップショットが占有するディスク領域です。実行中のアプリケーションで 多くのメモリが使用されている場合、スナップショットのサイズも大きくなります。
- ❹ スナップショットを作成した日時です。
- ❺ 仮想マシンのクロックについて、現時点での状態を示しています。

11.2.3.2 電源の切られた仮想マシンに対するスナップショット作成

電源の切られた仮想マシンについて、このスナップショットを作成するには、`qemu-img snapshot -c` スナップショット名 ディスクイメージを実行します。

```
tux@venus:~> qemu-img snapshot -c backup_snapshot /images/sles11sp1.qcow2
```

```
tux@venus:~> qemu-img snapshot -l /images/sles11sp1.qcow2
```

Snapshot list:

ID	TAG	VM SIZE	DATE	VM CLOCK
1	booting	4.4M	2010-11-22 10:51:10	00:00:20.476
2	booted	184M	2010-11-22 10:53:03	00:02:05.394
3	logged_in	273M	2010-11-22 11:00:25	00:04:34.843
4	ff_and_term_running	372M	2010-11-22 11:12:27	00:08:44.965
5	backup_snapshot	0	2010-11-22 14:14:00	00:00:00.000

VM ゲスト で復元したいような状況が発生し、保存したスナップショット (下記の 例では 5 番のスナップショット) に戻りたい場合は、VM ゲスト の電源を切って から下記のコマンドを実行します:

```
tux@venus:~> qemu-img snapshot -a 5 /images/sles11sp1.qcow2
```

実行完了後、`qemu-kvm` で仮想マシンを起動すると、5 番のスナップショットの状態に戻ることができます。

注記

`qemu-img snapshot -c` コマンドは、QEMU モニタの `savevm` コマンド (第13章 *QEMU モニタを利用した仮想マシンの管理* (157 ページ) をお読みください) とは無関係です。たとえば QEMU のモニタから `savevm` でスナップショットを作成した場合、このスナップショットを `qemu-img snapshot -a` コマンドで適用させることはできません。

11.2.3.3 スナップショットの削除

仮想マシンのスナップショットのうち、古いものや不要なものを削除したい場合は、`qemu-img snapshot -d` スナップショット ID ディスクイメージ コマンドを実行します。このコマンドを実行すると、`qcow2` のディスクイメージ内で スナップショット用に確保されていたディスク領域が開放されます:

```
tux@venus:~> qemu-img snapshot -d 2 /images/sles11sp1.qcow2
```

11.2.4 効果的なディスクイメージの作業

実際の活用例を考えてみます: あなたはサーバの管理者で、数多くの仮想化されたオペレーティングシステムを実行し、管理しているものとします。これらのうちの あるグループは、ある特定のディストリビューションをベースにした構成で、その他のグループはプラットフォームやディストリビューションが異なる (そしておそらくは Unix 以外の) オペレーティングシステムをベースにした構成であるものとします。さらに複雑なことに、それぞれの仮想化されたゲストシステムは、それぞれの部署や 配置環境に合わせて同じディストリビューションを異なる設定で使用しているものとします。一方はファイルサーバとして異なる設定を、他方は Web サーバなどの サービスを提供していて、いずれも SUSE Linux Enterprise Server 11 SP1 をベースにしているものとします。

QEMU では「ベース」と呼ばれるディスクイメージを作成することができます。これは仮想マシンの雛型として利用できるもので、これらのベースイメージを利用することで、オペレーティングシステムを何回もインストールしたりすることなく、手間を省くことができるようになっています。

11.2.4.1 ベースイメージと派生イメージ

まずは通常の手順でディスクイメージを作成し、目的となるシステムをインストールします。それぞれ詳しくは 11.1 項「qemu-kvm を利用した基本インストール」(112 ページ) と 11.2.2 項「ディスクイメージの作成や変換、チェック」(115 ページ) をお読みください。その後、ここまでで作成したイメージを元に、新しいイメージを構築します。ベースとなるイメージは '支援' ファイルとも呼ばれ、必要な '派生' イメージをここから作成し、ベースイメージを再作成するような手間を省いて派生イメージを直接起動できる 仕組みになっています。また、複数の派生イメージが 1 つのベースイメージを参照することもあります。そのため、ベースイメージを変更してしまうと、依存関係に問題が発生することになるので、QEMU では派生イメージにのみ変更点を書き込み、ベース イメージは読み込み専用としてアクセスします。

ベースイメージは新規にインストールした (必要であればユーザ登録済みの) オペレーティングシステムを利用し、一切の修正や追加のアプリケーションを適用 またはインストール／削除していない状態であることをお勧めします。ベースイメージに対して作成した派生イメージで、必要な最新の修正をインストールしておくのが便利です。

11.2.4.2 派生イメージの作成

注記

ベースイメージに対しては raw 形式を使用することができますが、raw 形式は派生イメージに必要な `backing_file` オプションに対応していないため、この形式を利用することができません。派生イメージに対しては、たとえば `qcow2` 形式をお使いください。

たとえば `/images/sles11sp1_base.raw` ファイルが、新規インストールしたイメージを含むベースイメージとします。

```
tux@venus:~> qemu-img info /images/sles11sp1_base.raw
image: /images/sles11sp1_base.raw
file format: raw
virtual size: 4.0G (4294967296 bytes)
disk size: 2.4G
```

イメージの予約済みサイズは 4 GB で、実際のサイズは 2.4 GB です。また、形式は `raw` になっています。この `/images/sles11sp1_base.raw` ファイルから 派生イメージを作成するには、下記のように実行します:

```
tux@venus:~> qemu-img create -f qcow2 /images/sles11sp1_derived.qcow2 \
-o backing_file=/images/sles11sp1_base.raw
Formatting '/images/sles11sp1_derived.qcow2', fmt=qcow2 size=4294967296 \
backing_file='/images/sles11sp1_base.raw' encryption=off cluster_size=0
```

作成された派生イメージの情報は、下記のようになります:

```
tux@venus:~> qemu-img info /images/sles11sp1_derived.qcow2
image: /images/sles11sp1_derived.qcow2
file format: qcow2
virtual size: 4.0G (4294967296 bytes)
disk size: 140K
cluster_size: 65536
backing file: /images/sles11sp1_base.raw \
(actual path: /images/sles11sp1_base.raw)
```

派生イメージが予約するサイズはベースイメージのものと同じ (4 GB) ですが、実際のサイズは 140 KB しかありません。これは派生イメージ側で行なわれた変更点だけが 派生イメージに書き込まれるためです。あとは派生イメージ側で仮想マシンを起動して、必要であれば登録を行なったり、最新の修正をインストールしたりしてください。また、不要なソフトウェアパッケージをアンインストールしたり、必要なものをインストールしたりすることもできます。その後 VM ゲスト をシャットダウンして、再度派生 イメージを確認すると、下記のようになります:

```
tux@venus:~> qemu-img info /images/sles11sp1_derived.qcow2
image: /images/sles11sp1_derived.qcow2
```

```
file format: qcow2
virtual size: 4.0G (4294967296 bytes)
disk size: 1.1G
cluster_size: 65536
backing file: /images/sles11sp1_base.raw ¥
(actual path: /images/sles11sp1_base.raw)
```

disk size の欄が 1.1 GB まで増えています。これは ベースイメージと比較し、ファイルシステム上に施された変更点が書き込まれている ためです。

11.2.4.3 派生イメージの再ベース

いったん派生イメージを修正する (修正の適用やアプリケーションのインストール、環境設定の変更など) を行なうと、派生イメージのサイズはそれなりのサイズになってしまいます。場合によっては、そこから既存のベースイメージと派生イメージを利用した '合成' イメージを作成したい場合が考えられます。たとえば最初の ベースイメージを新規にインストールしたシステムとして作成し、そこから セキュリティの修正や更新をインストールした状態を追加のベースイメージとしたりしたい場合が考えられます。あとは後者のほうのベースイメージを雛型にして 派生イメージを作成したりすることができます。後者のほうのベースイメージは、前者のベースイメージとは独立した存在となります。派生イメージから新しい ベースイメージを作成する作業を、'再ベース' と呼びます:

```
tux@venus:~> qemu-img convert /images/sles11sp1_derived.qcow2 ¥
-O raw /images/sles11sp1_base2.raw
```

このコマンドでは、新しいベースイメージ /images/sles11sp1_base2.raw を、raw 形式で作成します。

```
tux@venus:~> qemu-img info /images/sles11sp1_base2.raw
image: /images/sles11sp1_base2.raw
file format: raw
virtual size: 4.0G (4294967296 bytes)
disk size: 2.8G
```

新しいイメージは元々のベースイメージと比較して 0.4 GB ほど大きくなっています。また、このファイルには backing file (ベースイメージ) の表示がありませんので、このイメージファイルをベースにして、新しい派生イメージを作成することができます。これにより、お使いの環境に合わせて洗練された仮想ディスクイメージの構造を作成することができるため、作業の手間を大幅に省くことができます。

11.2.4.4 VM ホストサーバ におけるイメージのマウント

作業上の都合によっては、仮想ディスクイメージをホストシステム側でマウントしたほうが便利な場合があります。たとえば VM ホストサーバ 側にネットワークサポート

が存在しない場合、マウント作業は VM ゲスト とファイルの転送を行なう唯一の手段となります。

Linux システムでは、'ループバック' デバイスを利用することで raw 形式のディスクイメージ内のパーティションをマウントすることができます。最初の例は少し複雑ですが詳しい説明が、もう 1 つの例は簡単に わかりやすい方式です：

手順 11.1 パーティションオフセットを計算してディスクイメージをマウントする方法

- 1 まずは *loop* (ループ) デバイスを設定し、マウントしたいパーティションを含むディスクイメージを指定します。

```
tux@venus:~> losetup /dev/loop0 /images/sles11sp1_base.raw
```

- 2 次にマウントしたいパーティションの **セクタサイズ** と **セクタ番号** を確認します。

```
tux@venus:~> fdisk -lu /dev/loop0
```

```
Disk /dev/loop0: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders, total 8388608 sectors
Units = sectors of 1 * 512 = 512❶ bytes
Disk identifier: 0x000ceca8
```

	Device	Boot	Start	End	Blocks	Id	System
	/dev/loop0p1		63	1542239	771088+	82	Linux swap
	/dev/loop0p2	*	1542240 ^❷	8385929	3421845	83	Linux

- ❶ ディスクのセクタサイズを示しています。
- ❷ パーティションの開始セクタ番号を示しています。

- 3 下記のようにしてパーティションのオフセット値を計算します：

セクタサイズ * 開始セクタ番号 = 512 * 1542240 = 789626880

- 4 ループデバイスを削除し、計算結果のオフセット値を指定して、パーティションを用意しておいたディレクトリにマウントします。

```
tux@venus:~> losetup -d /dev/loop0
tux@venus:~> mount -o loop,offset=789626880 ¥
/images/sles11sp1_base.raw /mnt/sles11sp1/
tux@venus:~> ls -l /mnt/sles11sp1/
total 112
drwxr-xr-x  2 root root 4096 Nov 16 10:02 bin
drwxr-xr-x  3 root root 4096 Nov 16 10:27 boot
drwxr-xr-x  5 root root 4096 Nov 16 09:11 dev
[...]
drwxrwxrwt 14 root root 4096 Nov 24 09:50 tmp
```

```
drwxr-xr-x 12 root root 4096 Nov 16 09:16 usr
drwxr-xr-x 15 root root 4096 Nov 16 09:22 var
```

- 5 マウントできたパーティションに対して、ファイルをコピーしたりします。作業が終わったらマウントを解除します。

```
tux@venus:~> cp /etc/X11/xorg.conf /mnt/sles11sp1/root/tmp
tux@venus:~> ls -l /mnt/sles11sp1/root/tmp
tux@venus:~> umount /mnt/sles11sp1/
```

手順 11.2 kpartx を利用したディスクイメージのマウント

- 1 まずは *loop* (ループ) デバイスを設定し、マウントしたいパーティションを含むディスクイメージを指定します。

```
tux@venus:~> losetup /dev/loop0 /images/sles11sp1_base.raw
```

- 2 ディスクイメージのパーティション情報から、デバイスマップを作成します。

```
tux@venus:~> kpartx -a /dev/loop0
```

- 3 事前に用意しておいたディレクトリに、ディスクイメージ内のパーティションをマウントします。

```
tux@venus:~> mount /dev/mapper/loop0p1 /mnt/p1
```

なお、*loop0p1* の項目には、マウントしたいパーティションの 番号を指定してください。たとえば *loop0p3* と指定すると、ディスクイメージ内の 3 番目のパーティションをマウントすることになります。

- 4 あとは必要に応じてマウントされたパーティションに対してファイルやディレクトリの コピーや移動を行ないます。作業が終わったらマウントを解除し、ループデバイスを 削除します。

```
tux@venus:~> umount /mnt/p1
tux@venus:~> losetup -d /dev/loop0
```

警告

仮想マシンが実行中の場合は、それらのイメージに対して 読み書き可能な 形で マウントを行なってはなりません。これを行なってしまうと、パーティションの情報を破壊してしまうほか、VM ゲスト 全体を破壊してしまう場合があります。

qemu-kvm を利用した仮想マシンの実行

12

仮想ディスクイメージを作成 (詳しくは 11.2 項「qemu-img を利用したディスク管理」(114 ページ) を参照) したあとは、仮想マシンを起動することができるようになります。11.1 項「qemu-kvm を利用した基本インストール」(112 ページ) では、VM ゲスト を インストールして起動するまでの簡単なコマンド例を紹介してきましたが、この章では qemu-kvm の使用方法についてより細かい説明を 行ない、個別の要件についての解決方法を示しています。qemu-kvm のオプションについて完全な一覧をお読みになりたい場合は、マニュアルページ (man 1 qemu-kvm) をお読みください。

12.1 基本的な qemu-kvm の実行

qemu-kvm コマンドは、下記のような書式でコマンドを記述します:

qemu-kvm オプション❶ ディスクイメージ❷

- ❶ qemu-kvm には数多くのオプションが用意されています。これらのうちの多くはハードウェアのエミュレーションを設定するもので、残りは一般的なエミュレーション動作を指定するものです。特にオプションを 指定しない場合は既定値が使用され、実行すべきディスクイメージのパスを 指定することになります。
- ❷ 仮想化に使用したいゲストシステムについて、そのディスクイメージのパスを指定します。qemu-kvm では数多くのイメージ形式に対応しています。対応しているイメージ形式の一覧を取得するには、qemu-img --help を実行してください。ディスクイメージの パスは、ここでの指定以外にも -drive file= オプションでも 指定することができます。

12.2 一般的な qemu-kvm オプション

この章では、qemu-kvm で利用できる一般的なオプションのほか、仮想マシンのプロセッサ、メモリ、モデルタイプ、時刻の処理方法を指定する基本ハードウェアのエミュレーション関係のオプションを紹介します。

-name *ゲストの名前*

ゲストシステムの名前を指定します。ウインドウのタイトル (キャプション) 内に表示されるほか、VNC サーバでも利用されます。

-boot *オプション*

設定したドライブの起動順序を指定します。ドライブはそれぞれ文字で指定し、'a' と 'b' がフロッピーディスクドライブの 1 と 2 を、'c' が 1 台目のハードディスクを、'd' が 1 台目の CD-ROM ドライブをそれぞれ示します。また、'n' から 'p' はネットワーク起動のネットワークアダプタを示します。

たとえば `qemu-kvm [...] -boot order=ndc` のように指定すると、まずネットワークからの起動を試し、その後 CD-ROM ドライブの 1 台目、ハードディスクの 1 台目からそれぞれ起動を試します。

-pidfile *ファイル名*

QEMU のプロセス識別番号 (PID) の保存先ファイル名を指定します。これは QEMU をスクリプトから実行するような場合に便利です。

-nodefaults

既定では QEMU は、コマンドラインで特に何も指定しない場合でも、基本的な仮想デバイスについては作成を行ないます。このオプションは、このような機能を無効にするためのもので、グラフィックカードやネットワークカード、パラレルポートやシリアルポート、仮想コンソールなど、それぞれのデバイスを個別に指定しなければならなくなります。また、QEMU モニタについても、既定では接続されなくなります。

-daemonize

'daemonize' とは "デーモン化" の意味で、このオプションを指定すると、QEMU のプロセスがバックグラウンドで動作するようになります。また、接続を受けられる状態になった段階で、標準入出力との接続も切られるようになります。

12.2.1 基本的な仮想ハードウェア

-M マシンの種類

擬似するマシンの種類を指定します。利用可能なマシンの種類について、一覧を取得するには `qemu-kvm -M help` を実行してください。

```
tux@venus:~> qemu-kvm -M help
Supported machines are:
q35                Standard PC (Q35 + ICH9, 2009) (alias of pc-q35-1.4)
pc-q35-1.4         Standard PC (Q35 + ICH9, 2009)
pc                 Standard PC (i440FX + PIIX, 1996)
pc-i440fx-1.4      Standard PC (i440FX + PIIX, 1996) (default)
pc-1.3             Standard PC
pc-1.2             Standard PC
pc-1.1             Standard PC
pc-1.0             Standard PC
pc-0.15            Standard PC
pc-0.14            Standard PC
pc-0.13            Standard PC
pc-0.12            Standard PC
pc-0.11            Standard PC, qemu 0.11
pc-0.10            Standard PC, qemu 0.10
isapc              ISA-only PC
none              empty machine
```

-m メガバイト

仮想化環境に割り当てる RAM のサイズをメガバイト単位で指定します。既定は 512 MB です。

-balloon virtio

動的に VM ゲストに割り当てる RAM 容量を変化させるための、擬似仮想化デバイスを指定します。割り当ての上限値は、`-m` オプションで指定します。

-cpu CPU_モデル

プロセッサ (CPU) のモデルタイプを指定します。対応する CPU モデルの一覧を表示するには、`qemu-kvm -cpu ?` を実行してください。

```
tux@venus:~> qemu-kvm -cpu help
x86      qemu64   QEMU Virtual CPU version 1.4.0
x86      phenom   AMD Phenom(tm) 9550 Quad-Core Processor
x86      core2duo Intel(R) Core(TM)2 Duo CPU    T7700  @ 2.40GHz
x86      kvm64    Common KVM processor
x86      qemu32   QEMU Virtual CPU version 1.4.0
x86      kvm32    Common 32-bit KVM processor
x86      coreduo  Genuine Intel(R) CPU          T2600  @ 2.16GHz
x86      486
x86      pentium
x86      pentium2
```

x86	pentium3	
x86	athlon	QEMU Virtual CPU version 1.4.0
x86	n270	Intel(R) Atom(TM) CPU N270 @ 1.60GHz
x86	Conroe	Intel Celeron 4x0 (Conroe/Merom Class Core 2)
x86	Penryn	Intel Core 2 Duo P9xxx (Penryn Class Core 2)
x86	Nehalem	Intel Core i7 9xx (Nehalem Class Core i7)
x86	Westmere	Westmere E56xx/L56xx/X56xx (Nehalem-C)
x86	SandyBridge	Intel Xeon E312xx (Sandy Bridge)
x86	Haswell	Intel Core Processor (Haswell)
x86	Opteron_G1	AMD Opteron 240 (Gen 1 Class Opteron)
x86	Opteron_G2	AMD Opteron 22xx (Gen 2 Class Opteron)
x86	Opteron_G3	AMD Opteron 23xx (Gen 3 Class Opteron)
x86	Opteron_G4	AMD Opteron 62xx class CPU
x86	Opteron_G5	AMD Opteron 63xx class CPU

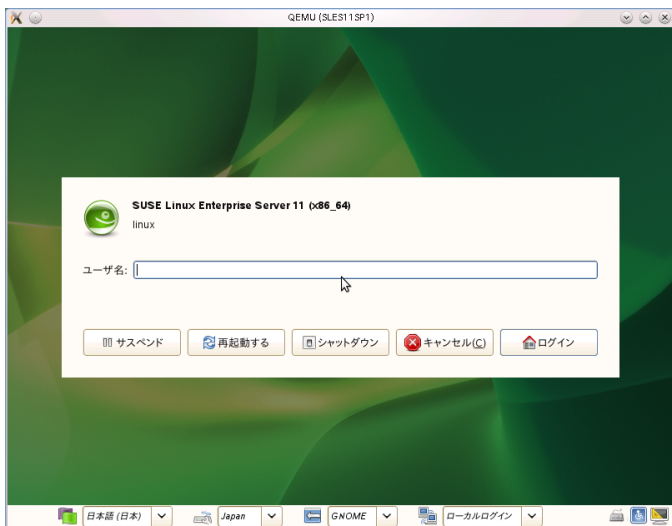
-smp CPU 数

仮想化環境での CPU 数を指定します。PC プラットフォームの場合、QEMU では最大で 255 個まで (KVM のアクセラレーション機能を使用した場合は 64 個まで) の CPU に対応します。このオプションは、ソケット の数やコア の数、およびコアごとの スレッド 数など、その他の CPU 関連パラメータも受け付けます。

下記は qemu-kvm の実行例です:

```
qemu-kvm -name "SLES 11 SP1" -M pc-0.12 -m 512 -cpu kvm64 \
-smp 2 /images/sles11sp1.raw
```

図 12.1 VM ゲストとして SLES 11 SP1 を利用した場合の QEMU ウィンドウ



`-no-acpi`

ACPI サポートを無効にします。VM ゲスト が ACPI インターフェイス まわりで問題を起こしているような場合、これを無効に設定してみてください。

`-S`

QEMU を CPU が停止した状態で起動します。CPU を動作させるには、QEMU モニタで `c` を入力してください。詳しくは 第13章 *QEMU モニタを利用した仮想マシンの管理* (157 ページ) をお読みください。

12.2.2 仮想デバイス設定の保存と読み込み

`-readconfig` 設定ファイル

`qemu-kvm` では、VM ゲスト の起動時に毎回コマンドラインで デバイスの設定情報を指定する代わりにファイルから設定を読み込むことができます。設定ファイルは `-writeconfig` で作成すること ができるほか、手作業でもファイルを作成することができます。

`-writeconfig` 設定ファイル

現在の仮想マシンのデバイス設定について、これをテキストファイルに書き出します。ここで保存したファイルは、`-readconfig` を利用すること で読み込ませることができます。

```
tux@venus:~> qemu-kvm -name "SLES 11 SP1" -M pc-0.12 -m 512 -cpu kvm64 ¥
-smp 2 /images/sles11sp1.raw -writeconfig /images/sles11sp1.cfg
(exited)
tux@venus:~> more /images/sles11sp1.cfg
# qemu config file
```

```
[drive]
  index = "0"
  media = "disk"
  file = "/images/sles11sp1_base.raw"
```

このオプションを利用することで、お使いの仮想マシンに関する設定を便利に管理することができますようになります。

12.2.3 ゲスト側のリアルタイムクロック

`-rtc` オプション

VM ゲスト での RTC (リアルタイムクロック (時計)) の処理方法を指定します。既定では、ゲストのリアルタイムクロックはホストシステムのものを利用します。

そのため、ホストシステムの時計は正確な外部クロック (たとえば NTP サービスなど) と同期させることをお勧めします。

VM ゲスト とホストとの時計を別途に管理する必要がある場合は、`clock=host` の既定値の代わりに、`clock=vm` を指定してください。

また `base` オプションを利用することで、VM ゲスト のクロックについて '開始時点の日時' を指定することもできます:

```
qemu-kvm [...] -rtc clock=vm,base=2010-12-03T01:02:00
```

タイムスタンプの代わりに `utc` や `localtime` を指定することもできます。前者は VM ゲスト のクロックを現在の UTC 時刻 (協定世界時; Coordinated Universal Time <http://ja.wikipedia.org/wiki/%E5%8D%94%E5%AE%9A%E4%B8%96%E7%95%8C%E6%99%82>) の現在時刻に、後者はローカルの時刻設定での現在時刻にそれぞれ合わせます。

12.3 QEMU における仮想デバイスの使用

QEMU の仮想マシンは、VM ゲスト を動作させるのに必要なすべてのデバイスを擬似します。たとえば QEMU は複数種類のネットワークカードやブロック デバイス (ハードディスクドライブ、リムーバブルドライブ)、USB デバイスや キャラクタデバイス (シリアルポート、パラレルポート)、マルチメディア デバイス (グラフィックカード、サウンドカード) に対応しています。また、仮想マシンに対して十分な操作性と性能を確保するため、それらのデバイスの うちの一部 (または全部) は正しく設定されていなければなりません。この 章では、対応するデバイスについて各種の設定方法を紹介しています。

ヒント

`-drive` などで指定するデバイスが特殊なドライバを必要とするもので、ドライバの属性を設定する必要がある場合は、それらを `-device` オプションで設定し、`drive=` サブオプションで識別することができます。たとえば下記のようになります:

```
qemu [...] -drive if=none,id=drive0,format=raw \
-device virtio-blk-pci,drive=drive0,scsi=off ...
```

利用可能なドライバと属性の一覧について、詳しくは `-device ?` および `-device ドライバ, ?` を実行してください。

12.3.1 ブロックデバイス

ブロックデバイスは仮想マシンにとって必要不可欠なものです。一般的にこれらは固定またはリムーバブルのストレージメディアで、一般には 'ドライブ' と呼ばれます。通常は 1 台だけハードディスクドライブが接続され、そこにゲスト側のオペレーティングシステムが保存されます。

仮想マシンのドライブは `-drive` で設定します。この オプションには数多くのサブオプションが存在していて、本章ではそれらのうちの 一部のみを紹介しています。完全な一覧をご希望の場合は、マニュアルページを 参照してください (`man 1 qemu-kvm`)。

`-drive` オプション向けのサブオプション

`file`=イメージファイル名

このドライブで使用するディスクイメージのパスを指定します。何も指定しない場合は、空の (リムーバブル) ドライブが指定されたものと 見なします。

`if`=ドライブインターフェイス

ドライブが接続されるインターフェイスの種類を指定します。現時点では SUSE では `floppy`, `ide`, `virtio` をサポートしています。`virtio` は擬似仮想化ディスクドライブの意味です。既定値は `ide` です。

`index`=コネクタ番号

ディスクインターフェイス上でのドライブの接続先コネクタ番号を指定します (`if` オプションをお読みください)。何も指定しない 場合は、コネクタ番号が自動で 設定 (順に割り当て) されます。

`media`=種類

メディアの種類を指定します。`disk` を指定すると ハードディスク、`cdrom` を指定するとリムーバブルな CD-ROM ドライブの意味になります。

`format`=イメージ形式

接続されるディスクイメージの形式を指定します。何も指定しない場合は、形式を自動で検出します。現時点では SUSE は `qcow2`, `qed`, `raw` の各形式をサポートしています。

`cache`=方法

ドライブのキャッシュ方法を指定します。`unsafe`, `writethrough`, `writeback`, `directsync`, `none` のいずれかを 指定します。なお、`qcow2` のイメージ形式の場合は、性能確保を目的としたい場合には `writeback` を指定 してください。また

none はホスト側でのページ キャッシュが無効化されるため、もっとも安全なオプションとなります。既定値は writeback です。

ヒント

ブロックデバイスの設定を簡単にするため、QEMU ではいくつかの ショートカット機能が用意されています。これを利用することで、qemu-kvm のコマンドライン入力をより便利に利用することができます。

たとえば下記のような書き方があります：

```
qemu-kvm -cdrom /images/cdrom.iso
```

上記を正しい (長い) 形式に直すと、下記のようにになります：

```
qemu-kvm -drive file=/images/cdrom.iso, index=2, media=cdrom
```

また、下記のような書き方もあります：

```
qemu-kvm -hda /images/image1.raw -hdb /images/image2.raw -hdc ¥  
/images/image3.raw -hdd /images/image4.raw
```

上記を正しい形式に直すと下記のようにになります：

```
qemu-kvm -drive file=/images/image1.raw, index=0, media=disk ¥  
-drive file=/images/image2.raw, index=1, media=disk ¥  
-drive file=/images/image3.raw, index=2, media=disk ¥  
-drive file=/images/image4.raw, index=3, media=disk
```

ヒント: イメージの代替としてのホスト側ドライブ使用

通常は、仮想マシンのディスクドライブにはディスクイメージを使用します (11.2項「qemu-img を利用したディスク管理」(114 ページ) を参照) が、VM ホストサーバ 側に接続されているディスクを使用して VM ゲスト 側からアクセスさせることもできます。この場合は、ディスクイメージのファイル名の代わりに、ホスト側のディスクデバイスを直接指定してください。

ホスト側の CD-ROM ドライブにアクセスするには、下記のように指定します：

```
qemu-kvm [...] -drive file=/dev/cdrom, media=cdrom
```

ホスト側のハードディスクにアクセスするには、下記のように指定します：

```
qemu-kvm [...] -drive file=/dev/hdb, media=disk
```

VM ゲスト からホスト側のハードディスクにアクセスする場合、そのアクセスが *読み込み専用* になっていることをお確かめください。読み込み専用かどうかは、ホスト側のデバイスのパーミッションで確認できます。

12.3.1.1 virtio-blk-data-plane

virtio-blk-data-plane は、KVM 向けの新しい性能改善 機能です。VM ゲストからやってくる I/O 要求に対して、高い性能を誇る コードを実行することができます。より詳しく書くと、この機能は専用の スレッドを (仮想ブロックデバイスごとに) 用意し、*virtio-blk* ドライバに対して I/O 要求を処理させる仕組みです。これにより、VM ホストサーバ 側の Linux カーネルで提供されている AIO (asynchronous I/O interface; 非同期 I/O インターフェイス) が直接利用できることになります。これは QEMU のブロック レイヤを介する必要をなくすものでもあるため、ストレージ設定で非常に高い I/O 速度を維持することができます。

virtio-blk-data-plane 機能は、VM ゲスト の起動時に *qemu* コマンドのオプションとして、*x-data-plane=on|off* を 付けることで有効化／無効化することができます。

```
qemu [...] -drive if=none,id=drive0,cache=none,aio=native,¥  
format=raw,file=filename -device virtio-blk-pci,drive=drive0,scsi=off,¥  
config-wce=off,x-data-plane=on [...]
```

なお、現時点では *virtio-blk-data-plane* に対し、下記の制限があります：

- raw イメージ形式のみに対応しています。
- ライブマイグレーションには対応していません。
- ブロックジョブやホットアンプラグ (活線抜去) は、-EBUSY で失敗します。
- I/O のスロットリング制限が無視されます。
- Linux AIO を使用するため、Linux の VM ホストサーバ のみに対応しています。ただし、VM ゲスト 側については Linux 以外でも対応しています。

重要

virtio-blk-data-plane 機能は、openSUSE ではまだサポート対象外 です。これは技術プレビュー状態という位置づけになります。

12.3.2 グラフィックデバイスとディスプレイオプション

この章では、ビデオカードの擬似に関する QEMU のオプションと、VM ゲスト のグラフィカル出力方法を説明しています。

12.3.2.1 ビデオカードの設定

QEMU では `-vga` オプションを利用することで VM ゲスト のグラフィカル表示出力用のビデオカードを設定することができます。`-vga` オプションでは下記のような値を利用することができます:

`none`

VM ゲスト でビデオカードを無効にします (ビデオカードの擬似を行ないません)。動作中の VM ゲスト にアクセスするには、QEMU モニタや シリアルコンソールを利用することができます。

`std`

標準 VESA 2.0 VBE ビデオカードを擬似します。VM ゲスト で高い表示 解像度を利用したい場合に指定してください。

`cirrus`

Cirrus Logic GD5446 を擬似します。擬似環境で高い互換性を重視するような場合に指定してください。多くのオペレーティングシステム (Windows 95 できれい) で、この種類のグラフィックカードに対応しています。

ヒント

`cirrus` を選択した際には、もっとも良い性能を引き出すため、VM ホストサーバと VM ゲスト の両方で 16 ビット色をご利用ください。

12.3.2.2 ディスプレイオプション

下記のオプションは、VM ゲスト のグラフィカル出力に対して詳細な 指定を行なうためのものです。

`-nographic`

QEMU のグラフィカル出力を無効化します。コンソールの出力は、擬似されたシリアルポート宛に行なわれます。

仮想マシンを `-nographic` オプション付きで 起動した場合は、仮想コンソール内で `Ctrl + AH` を押すと、コンソール と QEMU モニタの切り替えなどの便利なショートカットについて、一覧 表示が行なわれます。

```
tux@venus:~> qemu-kvm -hda /images/sles11sp1_base.raw -nographic
```

```
C-a h    print this help
```

```
C-a x    exit emulator
C-a s    save disk data back to file (if -snapshot)
C-a t    toggle console timestamps
C-a b    send break (magic sysrq)
C-a c    switch between console and monitor
C-a C-a  sends C-a
(pressed C-a c)

QEMU 0.12.5 monitor - type 'help' for more information
(qemu)
```

-no-frame

QEMU のウィンドウ装飾を無効化します。専用のデスクトップワークスペースを利用したい場合に便利です。

-full-screen

QEMU のグラフィカル出力をフルスクリーンで行なうようにします。

-no-quit

QEMU ウィンドウの '閉じる' ボタンを無効にし、強制的に閉じられてしまうことを防ぎます。

-alt-grab, -ctrl-grab

既定では QEMU のウィンドウは、Ctrl + Alt を押すことでマウスの 'キャプチャ' モードを解放します。このキーの組み合わせを Ctrl + Alt + Shift (-alt-grab の場合) または 右 Ctrl (-ctrl-grab) に変更することができます。

12.3.3 USB デバイス

KVM の VM ゲストで利用できる USB デバイスを作成するには、2 つの方法があります。1 つは VM ゲスト内で新しい USB デバイスを擬似 (エミュレーション) する方法、もう 1 つはホスト側にある既存の USB デバイスを VM ゲストに割り当てる方法です。QEMU で USB デバイスを使用するには、まず -usb オプションを指定して、汎用 USB ドライバを有効にする必要があります。このドライバを有効にしたあと、個別のデバイスを -usbdevice オプションで指定します。

12.3.3.1 VM ゲスト における USB デバイスのエミュレーション

SUSE では、下記の種類の USB デバイスをサポートしています: ディスク, ホスト, シリアル, ブライユ点字, ネットワーク, マウス, タブレット。

-usbdevice オプションで使用する USB デバイスの種類

disk

ファイルをベースにして、マストレージデバイスを擬似します。任意指定のオプション `format` を付けることで、手動で形式を指定することもできます (何も指定しなければ自動検出 になります)。

```
qemu-kvm [...] -usbdevice  
disk:format=raw:/virt/usb_disk.raw
```

host

ホストデバイスにパススルーします (バス.アドレス の形式で識別します)。

serial

ホストのキャラクタデバイスに接続される、シリアルコンバータです。

braille

ブライユ点字を表示するための、BrAPI を使用する点字デバイスを 擬似します。

net

CDC イーサネットと RNDIS プロトコルに対応するネットワークアダプタ を擬似します。

mouse

仮想的な USB マウスを擬似します。このオプションは既定で設定されている PS/2 マウスの擬似を無効にします。下記の例では、VM ゲストにおけるマウスの ハードウェア状態を、`qemu-kvm [...] -usbdevice mouse` で表示しています:

```
tux@venus:~> hwinfo --mouse  
20: USB 00.0: 10503 USB Mouse  
[Created at usb.122]  
UDI: /org/freedesktop/Hal/devices/usb_device_627_1_1_if0  
[...]  
Hardware Class: mouse  
Model: "Adomax QEMU USB Mouse"  
Hotplug: USB  
Vendor: usb 0x0627 "Adomax Technology Co., Ltd"  
Device: usb 0x0001 "QEMU USB Mouse"  
[...]
```

tablet

たとえばタッチスクリーンのように、絶対座標を使用するポインタデバイスを 擬似します。このオプションは既定の PS/2 マウスの擬似を無効にします。タブレットデバイスは、VNC プロトコルを介して VM ゲストを閲覧している ような

場合に便利な機能です。詳しくは 12.5 項「VNC を利用した VM ゲスト の閲覧」(150 ページ)をお読みください。

12.3.3.2 USB パススルー

ホスト側に接続された既存の USB デバイスを VM ゲスト 側に割り当てるには、まずホスト側でバス ID とデバイス ID を見つける必要があります。

```
tux@vmhost:~> lsusb
[...]
Bus 002 Device 005: ID 12d1:1406 Huawei Technologies Co., Ltd. E1750
[...]
```

上記の例は、USB デバイスがホスト側のバス番号 2、デバイス番号 5 に接続されている場合の例です。VM ゲスト 側では下記のような追加パラメータを指定して実行します:

```
qemu-kvm [...] -usb -device usb-host,hostbus=2,hostaddr=5
```

ゲストが起動したあとは、割り当てた USB デバイスが接続されているかどうかを確認してください。

```
tux@vmguest:~> lsusb
[...]
Bus 001 Device 002: ID 12d1:1406 Huawei Technologies Co., Ltd. E1750
[...]
```

注記

割り当てた USB デバイスのマウントについては、ゲスト側のオペレーティング システムで管理しなければならないことに注意してください。

12.3.4 PCI パススルー

PCI パススルー は、PCI デバイスに対して VM ゲスト からの排他的なアクセス 機能を提供する技術です。

注記

PCI パススルー を利用できるようにするには、まずお使いのコンピュータのマザー ボードのチップセットと BIOS、そして CPU のすべてが AMD の IOMMU (または Intel で言うところの VT-d) 仮想化技術に対応していなければ なりませ

ん。お使いのコンピュータが対応しているものかどうかを確認するには、お使いのシステムを提供した提供元に PCI パススルー をお尋ねください。

注記

なお SUSE では、グラフィックカードの割り当てはサポートしていません。

手順 12.1 PCI パススルー の設定

- 1 ホスト側で動作しているカーネルについて、CONFIG_DMAR_DEFAULT_ON が設定されていることを確認します:

```
grep CONFIG_DMAR_DEFAULT_ON /boot/config-`uname -r`
```

このオプションが設定されていない場合は、お使いのブートローダの設定を編集し、`intel_iommu=on` (Intel CPU の場合) または `iommu=pt iommu=1` (AMD CPU の場合) をそれぞれ追加してください。追加後は再起動を行なうことで、設定を反映することができます。

- 2 次にホスト側で IOMMU が有効化され、認識されていることを確認します。Intel CPU の場合は `dmesg | grep -e DMAR -e IOMMU` を、AMD CPU の場合は `dmesg | grep AMD-Vi` を実行します。何も出力されない場合は、お使いのハードウェアで IOMMU (VT-d) に対応しているかどうか、および BIOS 内で有効に設定されているかどうかをご確認ください。

- 3 次にゲストに割り当てる PCI デバイスを判断します。

```
tux@vmhost:~> lspci -nn
[...] 00:1b.0 Audio device [0403]: Intel Corporation 82801H (ICH8 Family) ¥
HD Audio Controller [8086:284b] (rev 02) [...]
```

ここで表示されたデバイス ID (上記の例では 00:1b.0) と製造元 ID (上記の例では 8086:284b) を記憶しておきます。

- 4 さらに、ホスト側のカーネルドライバに対してデバイスの切断要求を送信し、PCI スタブドライバに接続します。

```
tux@vmhost:~> modprobe pci_stub
tux@vmhost:~> echo "8086 284b" > /sys/bus/pci/drivers/pci-stub/new_id
tux@vmhost:~> echo "0000:00:1b.0" > /sys/bus/pci/devices/0000:00:1b.0/driver/unbind
tux@vmhost:~> echo "0000:00:1b.0" > /sys/bus/pci/drivers/pci-stub/bind
```

- 5 以上の作業で、PCI デバイスを割り当てた VM ゲスト を起動することができます。

```
qemu-kvm [...] -device pci-assign,host=00:1b.0
```

注記

該当の PCI デバイスが他のデバイスと IRQ を共有している場合、VM ゲストに対して割り当てることはできません。

KVM では VM ゲストに対して、PCI デバイスのホットプラグ (活性挿抜) を行なうこともできます。この機能を利用するには、QEMU モニタに切り替えて (詳しくは 第 13 章 *QEMU モニタを利用した仮想マシンの管理* (157 ページ) をお読みください) から、下記のコマンドを実行します:

- 接続:

```
device_add pci-assign,host=00:1b.0,id=new_pci_device
```

- 取り外し:

```
device_del new_pci_device
```

12.3.5 キャラクタデバイス

新しいキャラクタデバイスを作成するには、`-chardev` オプションをお使いください。このオプションでは下記のような汎用書式を使用します:

```
qemu-kvm [...] -chardev バックエンドのタイプ,id=ID 文字列
```

ここで、*バックエンドのタイプ* には `null`, `socket`, `udp`, `msmouse`, `vc`, `file`, `pipe`, `console`, `serial`, `pty`, `stdio`, `braille`, `tty`, `parport` のいずれかを指定します。また、すべてのキャラクタデバイスには最大で 127 文字までの識別子 (ID) を付与しなければなりません。これは、このデバイスを他のデバイスと区別するために使用します。バックエンドのタイプごとのサブオプションについて、詳しくはマニュアルページ (`man 1 qemu-kvm`) をお読みください。下記にはおおまかな説明を示します:

`null`

データを何も出力せず、受け取ったデータはすべて廃棄するだけの空のデバイスを作成します。

`stdio`

QEMU プロセスの標準入力または標準出力に接続します。

`socket`

双方向のストリームソケットを作成します。*path* を指定すると、Unix ソケットを作成します:

```
qemu-kvm [...] -chardev ¥
```

```
socket, id=unix_socket1, path=/tmp/unix_socket1, server
```

なお、*server* サブオプションを指定すると、ソケットは待ち受け側として動作します。

port を指定した場合は、TCP ソケットが 作成されます:

```
qemu-kvm [...] -chardev ¥  
socket, id=tcp_socket1, host=localhost, port=7777, server, nowait
```

上記のコマンドは、ローカル接続の待ち受け (*server*) TCP ソケットを作成します。また、QEMU ではクライアントの接続について 待機を行ないません (*nowait*)。

udp

VM ゲスト からリモートのホストに送信するネットワークトラフィック について、UDP プロトコルを利用するようにします。

```
qemu-kvm [...] -chardev udp, id=udp_fwd, host=mercury.example.com, port=7777
```

上記のコマンドは、リモートのホスト *mercury.example.com* との接続を行なう際に ポート 7777 を開き、このポートを利用して VM ゲスト に対する通信を行ないます。

vc

新しい QEMU テキストコンソールを作成します。オプションで仮想 コンソールのサイズを設定することができます:

```
qemu-kvm [...] -chardev vc, id=vc1, width=640, height=480 -mon chardev=vc1
```

上記のコマンドは *vc1* と呼ばれる仮想コンソールを 指定のサイズで作成します。作成した後は QEMU モニタから接続を行ないます。

file

VM ゲスト が出力するデータを VM ホストサーバ 上のファイルに保存します。*path* パラメータで出力先のファイルを設定する必要があります。なお、指定したファイルが存在しない場合は作成されます。

```
qemu-kvm [...] -chardev file, id=qemu_log1, path=/var/log/qemu/guest1.log
```

既定では QEMU はシリアルポートとパラレルポート、そして QEMU モニタ向けのキャラクタデバイスを作成します。それ以外にも、独自のキャラクタデバイスを 作成して使用することができます。下記のオプションもお読みください:

-serial キャラクタデバイス

VM ゲスト の仮想シリアルポートを VM ホストサーバ 上のキャラクタデバイス (キャラクタデバイス で指定するデバイス) に転送します。既定では、この値は

グラフィカルモードの場合には仮想コンソール (vc) を、グラフィカルモードでない場合には stdio の意味になります。また `-serial` には多くのサブオプションが用意されています。完全な一覧を読みたい場合は `man 1 qemu-kvm` を参照してください。

最大で 4 ポートまでのシリアルポートを擬似できます。すべての シリアルポートを無効にするには、`-serial none` を指定してください。

`-parallel` デバイス

VM ゲスト のパラレルポートをホスト側のデバイス デバイス に転送します。このオプションでは、`-serial` と同じ種類のデバイスに対応しています。

ヒント

VM ホストサーバ が openSUSE または SUSE Linux Enterprise Server の場合、`/dev/parportN` (N はポート番号) で示されるような、実際のパラレルポートを使用することができます。

最大で 3 ポートまでのパラレルポートを擬似できます。すべての パラレルポートを無効にするには、`-parallel none` を指定してください。

`-monitor` キャラクタデバイス

QEMU のモニタを VM ホストサーバ 上のキャラクタデバイス (キャラクタデバイス で指定するデバイス) に転送します。このオプションでは `-serial` と同じ種類のデバイスに対応しています。既定では、この値はグラフィカル モードの場合には仮想コンソール (vc) を、グラフィカル モードでない場合には stdio の意味になります。

利用可能なキャラクタデバイスのバックエンドについて、詳しくはマニュアル ページ (`man 1 qemu-kvm`) をお読みください。

12.4 QEMU でのネットワーキング

`-net` オプションを利用すると、VM ゲスト におけるネットワーク インターフェイスの種類とネットワークの種類を設定することができます。現時点では SUSE は `none`, `nic`, `user`, `bridge`, `tap` の各オプションをサポートしています。`-net` オプションのサブオプションについて、完全な一覧はマニュアルページ (`man 1 qemu-kvm`) をお読みください。

サポートされる `-net` のサブオプション

none

VM ゲスト 上でのネットワークカードの擬似機能を無効にします。ループバックデバイスである `lo` ネットワーク インターフェイスのみを利用することができるようになります。

bridge

TAP インターフェイスを設定するために指定したネットワークヘルパーを使用し、指定したブリッジに接続します。詳しくは 12.4.3 項「ブリッジ設定を利用するネットワークング」(146 ページ) をお読みください。

nic

新しいネットワークインターフェイスカード (NIC) を作成し、指定した 仮想ローカルエリアネットワーク (VLAN) に接続します。詳しくは 12.4.1 項「ネットワークインターフェイスカードの設定」(144 ページ) をお読みください。

user

ユーザモードネットワークングを指定します。詳しくは 12.4.2 項「ユーザモードネットワークング」(145 ページ) をお読みください。

tap

ブリッジ設定またはルーティング設定を利用し、ネットワーク機能を利用します。詳しくは 12.4.3 項「ブリッジ設定を利用するネットワークング」(146 ページ) をお読みください。

12.4.1 ネットワークインターフェイスカードの設定

新しい擬似ネットワークカードを追加するには、`-net nic` を使用します:

```
qemu-kvm [...] -net nic,vlan=1①,macaddr=00:16:35:AF:94:4B②,  
model=virtio③,name=ncard1④
```

- ① ネットワークインターフェイスを VLAN 番号 1 に接続します。ここで指定する番号は独自のもので、主に識別用に使用する値です。このサブオプションを省略すると、QEMU では既定値である 0 を使用します。
- ② ネットワークカードに対するメディアアクセス制御 (MAC) アドレスを指定しています。これは唯一となるべき番号で、常に指定しておくことをお勧めします。指定しない場合は、QEMU は既定の MAC アドレスを割り当て、接続される VLAN で矛盾が発生しないようにします。

- ③ ネットワークカードのモデルを指定します。`-net nic,model=?` と入力すると、お使いのプラットフォーム上の QEMU に対応するすべてのネットワークカードの一覧を表示します。

現時点では、SUSE は `rtl8139` と `virtio` のモデルをサポートしています。

12.4.2 ユーザモードネットワーキング

`-net user` オプションを指定すると、QEMU が提供する ユーザモードのネットワーク機能を利用することができます。このオプションは、特にネットワークモードが指定されない場合の既定値でもあります。そのため、下記のコマンドラインはいずれも同じ意味になります：

```
qemu-kvm -hda /images/sles11sp1_base.raw
qemu-kvm -hda /images/sles11sp1_base.raw -net nic -net user
```

このモードは、VM ゲスト からインターネットなどの外部のネットワーク資源に アクセスさせたい場合に便利なモードです。既定では一切のゲスト宛の通信が 許可されない仕組みになっているため、VM ゲスト はネットワーク上の他の ホストからは参照できません。また、このネットワーキングモードでは、管理者 権限も不要です。また、ユーザモードは VM ホストサーバ 上にあるローカルディレクトリ から VM ゲスト を 'ネットワーク起動' したい場合にも便利です。

VM ゲスト は仮想の DHCP サーバから IP アドレスの割り当てを受けます。VM ホストサーバ (DHCP サーバ) は 10.0.2.2 としてアクセスできるようになっていて、ゲスト側に割り当てられるアドレスは 10.0.2.15 以降のアドレスになっています。ssh コマンドを利用すれば 10.0.2.2 にある VM ホストサーバ に アクセスすることができ、scp ではファイルを ホストからゲスト、またはゲストからホストにコピーすることもできます。

12.4.2.1 コマンドライン例

この章では、QEMU のユーザモードネットワーキングを設定するための 方法について、いくつかの例示を含めて示しています。

例 12.1 制限付きユーザモードネットワーキング

```
qemu-kvm [...] -net user①,vlan=1②,name=user_net1③,restrict=yes④
```

- ① ユーザモードネットワーキングの指定をしています。
- ② VLAN 番号 1 に接続する指定です。何も指定しない場合は 0 になります。
- ③ ネットワークスタックに対するわかりやすい名前を指定しています。QEMU モニタ内で識別する際に便利な機能です。

- ④ VM ゲスト を孤立させるための指定です。これにより VM ホストサーバ への通信が 行なえなくなるほか、外部のネットワークとの通信も遮断されます。

例 12.2 独自の IP アドレス範囲を指定するユーザモードネットワーキング

```
qemu-kvm [...] -net user,net=10.0.0.0/8①,host=10.0.0.6②,dhcpstart=10.0.0.20③,¥  
hostname=tux_kvm_guest④
```

- ① VM ゲスト に対して割り当てる IP アドレス範囲を指定しています。任意でネットマスクも指定できます。既定値は 10.0.2.0/8 です。
- ② VM ゲスト から見た VM ホストサーバ のアドレスを指定しています。既定値は 10.0.2.2 です。
- ③ 内蔵の DHCP サーバが VM ゲスト に対して割り当てる最初の 16 個の IP アドレスを指定します。既定値は 10.0.2.15 です。
- ④ 内蔵の DHCP サーバが VM ゲスト に割り当てるホスト名を指定しています。

例 12.3 ネットワーク起動と TFTP を使用するユーザモードネットワーキング

```
qemu-kvm [...] -net user,tftp=/images/tftp_dir①,bootfile=/images/boot/pxelinux.0②
```

- ① 内蔵の TFTP (とても基本的な機能だけを実装するファイル転送プロトコル) サーバを有効にしています。VM ゲスト 側では、TFTP サーバのルート ディレクトリとして指定したディレクトリ内のファイルが見える形に なっています。
- ② 指定したファイルを BOOTP (IP アドレスの付与と起動イメージの場所を 通知するプロトコル。ディスク装置のないワークステーションでよく使用 されるもの) ファイルとしてブロードキャストする指定です。tftp とともに使用した場合は、ホスト上のローカル ディレクトリから起動できるようになります。

例 12.4 ホスト側でのポート転送機能を利用するユーザモードネットワーキング

```
qemu-kvm [...] -net user,hostfwd=tcp::2222-:22
```

ホスト側のポート 2222 に到達した TCP の接続を、VM ゲスト 側のポート 22 (SSH) に転送する指定です。VM ゲスト 上で sshd が動作 していれば、

```
ssh qemu_host -p 2222
```

と入力することで、VM ゲスト 内の SSH に接続できるようになります。ここで qemu_host は、ホスト側のホスト名または IP アドレスとします。

12.4.3 ブリッジ設定を利用するネットワーキング

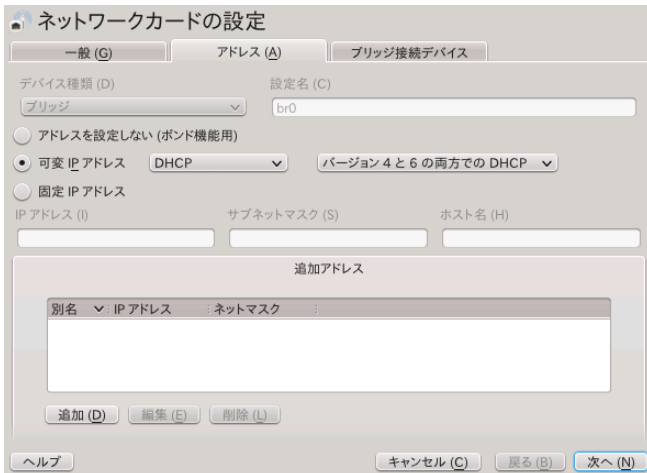
-net tap オプションを指定すると、QEMU はホスト側の TAP ネットワークデバイス と VM ゲスト 側の指定した VLAN を接続し、ネットワークブリッジを作成します。こ

のネットワークインターフェイスは、ネットワーク内の他のホストから参照できます。また、この方法は既定では動作できず、特殊な設定作業が必要です。

まずはネットワークブリッジを作成し、それを物理的なネットワーク インターフェイス (通常は `eth0`) に追加します:

- 1 YaST コントロールセンター を起動し、[ネットワークデバイス] > [ネットワークの設定] を選択します。
- 2 [追加] を押して [ハードウェアダイアログ] ウィンドウを開きます。ウィンドウを開いたら [デバイス種類] のドロップダウンリストで [ブリッジ] を選択します。選択を行ったら [次へ] を押します。
- 3 動的に IP アドレスを割り当てるか、もしくは固定で割り当てるかを選択します。また、必要であれば各種のネットワーク設定を行ないます。
- 4 [ブリッジ接続デバイス] のタブでは、ブリッジに追加したい イーサネットデバイスを選択します。

図 12.2 YaST を利用したネットワークブリッジの設定



さらに [次へ] を押します。既に設定済みのデバイスをブリッジに追加しようとしている場合は、その旨の確認メッセージが表示されるので、[続行] を押します。

- 5 [OK] を押して変更点を保存します。下記のように入力すると、ブリッジが作成されたことを確認することができます:

```
tux@venus:~> brctl show
bridge name bridge id          STP enabled  interfaces
br0          8000.001676d670e4  no          eth0
```

12.4.3.1 手作業によるブリッジ接続

下記のサンプルスクリプトを使用することで、新しく作成したブリッジ インターフェイス `br0` に VM ゲスト を接続することができます。下記のうちのいくつかは、`root` の権限が必要となるため、`sudo` などの仕組みを利用して実行しています。

注記

なお、VM ホストサーバ 側に `tunctl` と `bridge-utils` のパッケージがインストールされていることをご確認ください。インストールされていない場合は、`zypper in tunctl bridge-utils` コマンドでインストールを行なうことができません。

```
#!/bin/bash
bridge=br0❶
tap=$(sudo tunctl -u $(whoami) -b)❷
sudo ip link set $tap up❸
sleep 1s❹
sudo brctl addif $bridge $tap❺
qemu-kvm -m 512 -hda /images/sles11sp1_base.raw ¥
-net nic,vlan=0,model=virtio,macaddr=00:16:35:AF:94:4B ¥
-net tap,vlan=0,ifname=$tap❻,script=no❼,downscript=no
sudo brctl delif $bridge $tap❸
sudo ip link set $tap down❾
sudo tunctl -d $tap❿
```

- ❶ ブリッジデバイスの名前を指定しています。
- ❷ 新しい TAP デバイスを作成し、スクリプトを実行しているユーザに 割り当てています。TAP デバイスは仮想化やエミュレーション (擬似環境) でよく使用される仮想ネットワークデバイスです。
- ❸ 新しく作成した TAP ネットワークインターフェイスを有効にしています。
- ❹ 新しい TAP ネットワークインターフェイスが有効になるまで 1 秒間の 待機を行なっています。
- ❺ 新しい TAP デバイスを、ネットワークブリッジ `br0` に追加しています。
- ❻ `ifname=` サブオプションで、ブリッジに使用する TAP ネットワークインターフェイスの名前を指定しています。
- ❼ `qemu-kvm` がネットワークブリッジに接続する前に、`script` と `downscript` の値を確認します。VM ホストサーバ のファイルシステム内に指定したスクリプトが存在すると、ネットワークブリッジへの接続が行なわれる前に `script` で

指定したスクリプトを、ネットワーク環境が 終了した後に `downscript` で指定したスクリプトを、それぞれ実行します。これらのスクリプトを利用することで、ブリッジ型の ネットワークデバイスを設定したり設定を消したりすることができます。 になっています。既定では `/etc/qemu-ifup` と `/etc/qemu-ifdown` のスクリプトを実行します。また、`script=no` や `downscript=no` を指定するとスクリプトは実行されなくなり、ブリッジ周りの作業は別途に 行なうことになります。

- ⑧ ネットワークブリッジ `br0` から TAP インターフェイスを 削除しています。
- ⑨ TAP デバイスを 'ダウン' 状態に移行させています。
- ⑩ TAP デバイスの設定を消去しています。

12.4.3.2 qemu-bridge-helper を利用したブリッジへの接続

VM ゲスト をブリッジを介してネットワークに接続するもう 1 つの方法として、`qemu-bridge-helper` と呼ばれるヘルパープログラムを 利用する方法があります。これは TAP インターフェイスを設定して、指定した ブリッジに接続するまでの処理を行います。既定のヘルパー実行ファイルは `/usr/lib64/qemu-bridge-helper` で、ヘルパー実行 ファイルは `setuid root` の設定が為されていて、かつ仮想化グループ (`kvm`) のメンバーのみが実行できるように設定されます。そのため、`qemu-kvm` コマンドそれ自身は、`root` の権限無しに実行することができます。

ヘルパープログラムを実行するには、下記のように実行します:

```
qemu-kvm [...] -net nic,vlan=0,model=virtio -net bridge,vlan=0,br=br0
```

TAP デバイスの設定や設定解除を実施する、独自のヘルパースクリプトを 呼び出すように設定することもできます。この場合は `helper=/path/to/your/helper` の形式で指定します:

```
qemu-kvm [...] -net bridge,vlan=0,br=br1,helper=/path/to/bridge-helper
```

ヒント

`qemu-bridge-helper` に対するアクセス権限を設定するには、`/etc/qemu-kvm/bridge.conf` ファイルをご覧ください。たとえば下記のような設定の場合、`qemu-kvm` コマンドは VM ゲスト をネットワークブリッジ `br0` に接続できるようになります:

```
allow br0
```

12.4.4 vhost-net を利用したネットワークのアクセラレーション

vhost-net モジュールを利用することで、KVM の 擬似仮想化ネットワークドライバの性能を上げることができます。これにより、ネットワークの遅延低下とスループットの増大を実現できます。

このモジュールを利用するには、ホスト側で動作しているカーネルについて、CONFIG_VHOST_NET が設定されていて、カーネル内蔵 もしくはモジュールになっていることを確認します：

```
grep CONFIG_VHOST_NET /boot/config-`uname -r`
```

また、同様にゲスト側のカーネルでも、CONFIG_PCI_MSI が設定されていることを確認します：

```
grep CONFIG_PCI_MSI /boot/config-`uname -r`
```

両方の条件が満たされたら、あとは下記のコマンドラインのように、vhost-net ドライバを利用してゲストを起動します：

```
qemu-kvm [...] -netdev tap,id=guest0,vhost=on,script=no  
-net nic,model=virtio,netdev=guest0,macaddr=00:16:35:AF:94:4B
```

ここで、guest0 は vhost 制御のデバイスに対する 識別文字列を指定します。

12.5 VNC を利用した VM ゲスト の 関 覧

QEMU は通常、SDL (複数プラットフォームに対応したマルチメディアライブラリ) のウィンドウ機能を利用して VM ゲスト のグラフィカルな出力を表示します。-vnc オプションを指定すると、QEMU に対して指定した VNC ディスプレイを待ち受けるように指定することができます。これにより、グラフィカルな 出力を VNC セッション内に転送させることができます。

ヒント

QEMU の仮想マシンを VNC セッションを介して作業している場合、-usbdevice tablet オプションを指定して作業を行なったほうが 操作しやすくなります。

また、既定の en-us (英語) 以外のキーボードレイアウトを 使用したい場合は、-k オプションを利用して指定してください。

-vnc の最初のサブオプションは ディスプレイ の値でなければなりません。-vnc オプションでは、下記のような ディスプレイ値を指定することができます：

host:ディスプレイ

host で指定したホストの ディスプレイ で 指定したディスプレイ番号からの接続のみを受け付けます。VNC セッションで利用する TCP ポートは通常、ディスプレイ で指定した番号に 5900 を足した 値となります。host を指定しない場合は、任意のホストからの 接続を受け付けます。

unix:パス

VNC サーバは Unix ドメインソケットからの接続だけを受け付けます。unix オプションでは、Unix ソケットの場所を指定することができます。

none

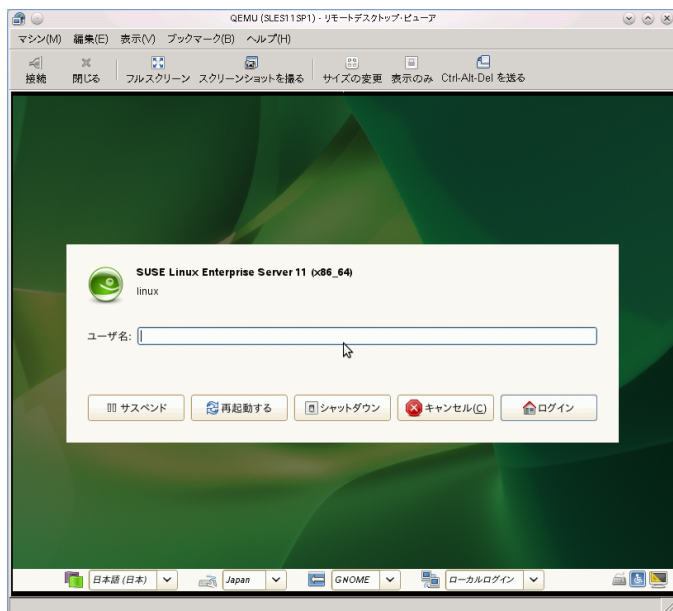
VNC サーバ機能は初期化されますが、サーバ自身は起動しないようになります。起動後に QEMU モニタから VNC サーバを起動することができます。詳しくは 第13章 *QEMU モニタを利用した仮想マシンの管理* (157 ページ) をお読みください。

```
tux@venus:~> qemu-kvm [...] -vnc :5
```

(クライアント側で:)

```
wilber@jupiter:~> vinagre venus:5905 &
```

図 12.3 QEMU VNC セッション



12.5.1 機密を保持する VNC 接続

既定の VNC サーバ設定は認証を全く行ないません。上述の例ではネットワーク上にある任意のホストから、任意のユーザが接続して QEMU の VNC セッションを閲覧することができます。

VNC のクライアント／サーバ間の接続では、様々なレベルのセキュリティを設定することができます。QEMU の 1 つのコマンドを入力するだけでパスワードでの接続保護を設定することができるほか、x509 証明書を設定したり SASL による認証を行ったり、複数の認証方法を組み合わせたりすることができます。

x509 の証明書生成について、詳しくは A.2 項「x509 クライアント／サーバ証明書の生成」(172 ページ)をお読みください。また、VM ホストサーバとクライアント間での x509 証明書の設定方法については、6.2.2 項「x509 証明書を利用したリモートの TLS/SSL 接続 (qemu+tls)」(55 ページ)と 6.2.2.3 項「クライアントの設定と設定テスト」(57 ページ)をお読みください。

Vinagre VNC ビューアでは高度な認証メカニズムに対応しています。そのため、下記のような形で VM ゲストのグラフィカル出力を閲覧することができます。この例では、サーバ証明書である `ca-cert.pem`, `server-cert.pem`, `server-key.pem` がホスト側の `/etc/pki/qemu` ディレクトリに配置されていて、クライアント側の証明書はクライアント内の下記の場所に配置されているものとします：

```
/etc/pki/CA/cacert.pem
/etc/pki/libvirt-vnc/clientcert.pem
/etc/pki/libvirt-vnc/private/clientkey.pem
```

例 12.5 パスワード認証

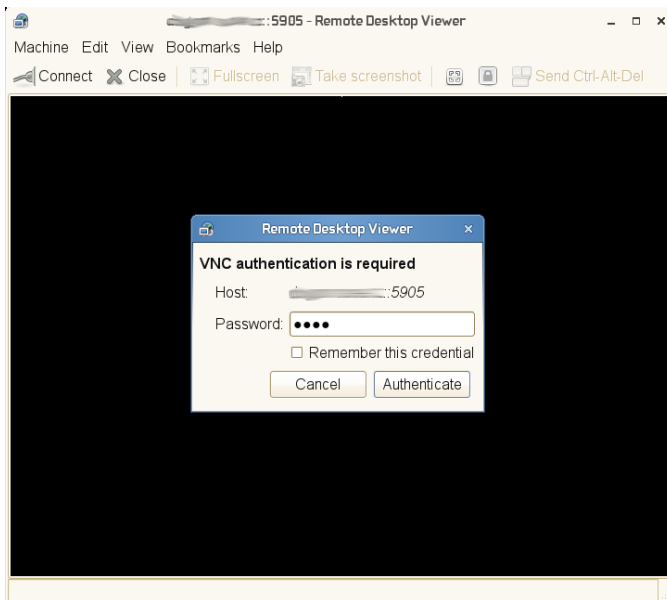
```
qemu-kvm [...] -vnc :5,password -monitor stdio
```

VM ゲストのグラフィカル出力を VNC ディスプレイ番号 5 番 (通常は 5905 番ポートを使用) に設定して起動します。password サブオプションではシンプルなパスワードベースの認証方法を指定しています。既定ではパスワードは何も設定されていないため、QEMU モニタから `change vnc password` コマンドで設定を行ないます：

```
QEMU 0.12.5 monitor - type 'help' for more information
(qemu) change vnc password
Password: ****
```

なお、ここでは `-monitor stdio` オプションを指定していますが、これは QEMU モニタを管理するため、入出力を転送しなければならないためです。

図 12.4 Vinagre での認証ダイアログ



例 12.6 x509 証明書認証

QEMU VNC サーバでは、セッションに対して TLS の暗号化を設定し、x509 証明書による認証を設定することができます。サーバはクライアントに対して証明書を要求し、その証明書を受け取ると証明機関 (CA) の証明書と照らし合わせて検証を行いません。この方法は、社内用の証明機関が用意されているような企業環境で利用してください。

```
qemu-kvm [...] -vnc :5,tls,x509verify=/etc/pki/qemu
```

例 12.7 x509 証明書とパスワード認証

クライアントに対する 2 階層の認証モデルを構築する目的で、TLS 暗号と x509 証明書認証に加えて、パスワードによる認証を組み合わせることができます。下記のコマンドを入力した後、QEMU モニタで忘れずにパスワードを設定してください:

```
qemu-kvm [...] -vnc :5,password,tls,x509verify=/etc/pki/qemu -monitor stdio
```

例 12.8 SASL 認証

簡易認証セキュリティ層 (Simple Authentication and Security Layer (SASL)) は、インターネットプロトコルにおける認証とデータセキュリティのフレームワークで

す。PAM, Kerberos LDAP など、複数の認証メカニズムを組み合わせて統合します。SASL では独自のデータベースを持っている仕組みであるため、接続を行なうユーザ アカウントは、VM ホストサーバ上に (Unix の) アカウントとして存在する必要はありません。

なお、セキュリティ上の理由から、SASL 認証を利用する際は TLS 暗号と x509 証明書認証を組み合わせて使用してください:

```
qemu-kvm [...] -vnc :5,tls,x509,sasl -monitor stdio
```

12.6 VirtFS: ホストとゲストの間でのフォルダ共有

VM ゲスト は通常、分離された環境で動作します。メモリ領域は独自の範囲で動作しますし、CPU やファイルシステムについても専用に割り当てられます。ですが、VM ホストサーバ のファイルシステムの一部分を共有することで、容易に相互 のデータ交換を実現できるため、仮想化環境をより柔軟に運用できるようになります。CIFS や NFS などのようなネットワークファイルシステムを利用して、従来のようにフォルダを共有することもできますが、これらの機能は 仮想化用に作られているものではなく、性能面や機能面に大きな問題があります。

KVM では *VirtFS* と呼ばれる (場合によっては「ファイルシステムパススルー」とも呼ばれます)、新しくかつ 最適化されたツールが用意されています。VirtFS は擬似仮想化されたファイル システムのドライバを使用するため、ゲスト側のアプリケーションのファイル システムをブロックデバイスの操作に変換したり、その結果を逆にゲスト側に 戻したりするような処理を省くことができます。VirtFS は Plan-9 のネットワーク プロトコルを利用し、ゲスト-ホスト間で通信します。

VirtFS の使用例としては、下記のようなものがあります。

- 複数のゲストから単一の共有フォルダにアクセスしたり、ゲスト間で共有する ファイルシステムを提供したりする目的。
- ルートファイルシステムとして割り当てていた仮想ディスクを、ゲスト側の 起動時に接続する RAM ディスクに置き換える目的。
- クラウド環境などの形で、単一のホスト側のファイルシステムを複数の顧客 に提供するようなストレージサービス。

12.6.1 実装

QEMU では、VirtFS の実装は 2 種類のデバイスとして提供されています:

- ホストとゲストの間でプロトコルメッセージやデータをやりとりする virtio-9p-pci デバイス
- ファイルシステムの種類やセキュリティモデルなど、ファイルシステムの プロパティ情報を開示するための fsdev デバイス

例 12.9 VirtFS を利用したホスト側のファイルシステムの公開

```
qemu-kvm [...] -fsdev local,id=exp1❶,path=/tmp/❷,security_model=mapped❸  
-device virtio-9p-pci,fsdev=exp1❹,mount_tag=v_tmp❺
```

- ❶ 公開対象のファイルシステムに割り当てる識別子。
- ❷ 公開対象のファイルシステムのホスト側パス
- ❸ 使用するセキュリティモデル。mapped は、ゲスト側の ファイルシステムのモードとパーミッションをホスト側と区別して処理します。none は「パススルー」セキュリティモデル と呼ばれ、ゲスト側のファイルに対して実施したパーミッション変更を、ホスト側にも反映します。
- ❹ -fsdev id= で指定した、公開対象のファイルシステム 識別子。
- ❺ ゲスト側でファイルシステムをマウントする際に使用する、マウントタグ。

公開されたファイルシステムは、ゲスト側で下記のようにしてマウントすることができ
ます:

```
mount -t 9p -o trans=virtio v_tmp /mnt
```

ここで、v_tmp には -device mount_tag= で指定したマウントタグを、/mnt には公開されている ファイルシステムのマウント先を指定します。

12.7 KSM: ゲスト間でのメモリページ共有

カーネル同一ページ合成 (Kernel SamePage Merging (KSM)) は、複数のプロセスで全く同一のページが存在した場合に、それらを共有するための Linux カーネルの機能です。KVM のゲストは Linux のプロセスとして動作する仕組みであるため、KSM はオーバーコミットの機能を利用してより効率的なメモリ管理を行なう

ことができます。そのため、限られたメモリ内で複数の仮想マシンを動作 させる必要があるような場合に、KSM はベストな解決方法となります。

KSM を利用するには、下記の手順を実施します。

1 まずはお使いのカーネルで、KSM が有効化されているかどうかを確認します：

```
grep KSM /boot/config-`uname -r`  
CONFIG_KSM=y
```

お使いのカーネルで KSM が有効化されている場合、`/sys/kernel/mm/ksm` ディレクトリ以下に下記のような ファイルが存在しているはずです：

```
ls -l /sys/kernel/mm/ksm  
total 0  
drwxr-xr-x 2 root root  0 Nov  9 07:10 ./  
drwxr-xr-x 6 root root  0 Nov  9 07:10 ../  
-r--r--r-- 1 root root 4096 Nov  9 07:10 full_scans  
-r--r--r-- 1 root root 4096 Nov  9 07:10 pages_shared  
-r--r--r-- 1 root root 4096 Nov  9 07:10 pages_sharing  
-rw-r--r-- 1 root root 4096 Nov  9 07:10 pages_to_scan  
-r--r--r-- 1 root root 4096 Nov  9 07:10 pages_unshared  
-r--r--r-- 1 root root 4096 Nov  9 07:10 pages_volatile  
-rw-r--r-- 1 root root 4096 Nov  9 07:10 run  
-rw-r--r-- 1 root root 4096 Nov  9 07:10 sleep_millisecs
```

2 KSM の機能が有効化されているかどうかを確認します。

```
cat /sys/kernel/mm/ksm/run
```

上記のコマンドの実行結果が `0` であった場合は、下記のコマンドで有効化することができます：

```
echo 1 > /sys/kernel/mm/ksm/run
```

3 あとは KVM を利用して、複数の VM ゲスト を実行し、`pages_sharing` と `pages_shared` の内容を確認します。たとえば下記のように なります：

```
while [ 1 ]; do cat /sys/kernel/mm/ksm/pages_shared; sleep 1; done  
13522  
13523  
13519  
13518  
13520  
13520  
13528
```

`/sys/kernel/mm/ksm/*` ファイルの意味について、詳しくは `/usr/src/linux/Documentation/vm/ksm.txt` をお読みください (kernel-source パッケージ内にあります)。

QEMU モニタを利用した仮想マシンの管理

13

QEMU が実行中であればモニタコンソールが提供され、ユーザとの対話操作が可能となります。モニタコンソールでのコマンドを利用すると、動作中のオペレーティングシステムの調査やリムーバブルメディアの交換を行ったり、スクリーンショットやサウンドデータを採取したり、その他仮想マシンに関するいくつかの制御を行なうことができます。

13.1 モニタコンソールへのアクセス

QEMU からモニタコンソールにアクセスするには、Ctrl + Alt + 2 を押します。QEMU の モニタコンソールから抜けたい場合は、Ctrl + Alt + 1 を押します。

コンソールの使用中にヘルプを表示させたい場合は、help か ? コマンドを利用します。また、特定のコマンドに対してヘルプを表示するには、help コマンドのように入力します。

13.2 ゲストシステムについての情報取得

ゲストシステムについての情報を取得するには info オプション コマンドを利用します。何もオプションを指定せずに実行すると、利用可能なオプションの一覧が表示されます。オプションでは分析したい項目を指定します：

`info version`

QEMU のバージョンを表示します。

`info commands`

利用可能な QMP コマンドを表示します。

`info network`

ネットワークの状態を表示します。

`info chardev`

キャラクタデバイスを表示します。

`info block`

ハードディスクドライブやフロッピーディスクドライブ、CD-ROM ドライブ など、ブロックデバイスについての情報を表示します。

`info blockstats`

ブロックデバイスについての読み込み／書き込み統計情報を表示します。

`info registers`

CPU レジスタを表示します。

`info cpus`

利用可能な CPU についての情報を表示します。

`info history`

コマンドラインの履歴を表示します。

`info irq`

割り込みに関する統計情報を表示します。

`info pic`

i8259 (PIC) の状態を表示します。

`info pci`

PCI に関する情報を表示します。

`info tlb`

仮想メモリと物理メモリのマッピング情報を表示します。

`info mem`

有効な仮想メモリのマッピング情報を表示します。

`info jit`

ダイナミックコンパイラの情報を表示します。

`info kvm`

KVM の情報を表示します。

`info numa`

NUMA の情報を表示します。

`info usb`

ゲスト側の USB デバイスの情報を表示します。

`info usbhost`

ホスト側の USB デバイスの情報を表示します。

`info profile`

プロファイル情報を表示します。

`info capture`

キャプチャ (オーディオ録音) 情報を表示します。

`info snapshots`

現在保存されている仮想マシンのスナップショットについて、情報を表示します。

`info status`

現在の仮想マシンの情報を表示します。

`info pcmcia`

PCMCIA の状態を表示します。

`info mice`

ゲスト側のどのマウスがイベントを受信しているかを 表示します。

`info vnc`

VNC サーバ情報を表示します。

`info name`

現在の仮想マシン名を表示します。

`info uuid`

現在の仮想マシンの UUID を表示します。

`info usernet`
ユーザネットワークのスタック接続状態を表示します。

`info migrate`
マイグレーション状態を表示します。

`info balloon`
バルーンデバイスの情報を表示します。

`info qtree`
デバイスの木構造を表示します。

`info qdm`
`qdev` デバイスモデルリストを表示します。

`info roms`
ROM の情報を表示します。

`info migrate_cache_sizes`
現在のマイグレーション `xbzrl` (=Xor Based Zero Run Length Encoding; XOR ベースのゼロ長エンコーディング) のキャッシュサイズを表示します。

`info migrate_capabilities`
`xbzrl` 圧縮などの様々なマイグレーション機能の状態を表示します。

`info mtree`
VM ゲスト のメモリ階層構造を表示します。

`info trace-events`
利用可能なトレースイベントとその状態を表示します。

13.3 VNC パスワードの変更

VNC のパスワードを変更するには、`change vnc password` コマンドを入力し、新しいパスワードを入力します:

```
(qemu) change vnc password
Password: *****
(qemu)
```

13.4 デバイスの管理

リムーバブルメディアデバイスに接続されているデバイスやファイルを開放 (取り出す) には、`eject デバイス名` コマンドを利用します。`-f` オプションを指定すると、強制的な取り出しを実行します。

リムーバブルメディア (たとえば CD-ROM など) のメディア交換を行なうには、`change デバイス名` コマンドを利用します。リムーバブルメディアのデバイス名は、`info block` コマンドで取得できます:

```
(qemu) info block
ide1-cd0: type=cdrom removable=1 locked=0 file=/dev/sr0 ro=1 drv=host_device
(qemu) change ide1-cd0 /path/to/image
```

13.5 キーボードとマウスの操作

モニタコンソールから、キーボードやマウスの入力を擬似させることができます。たとえば お使いのグラフィカルユーザインターフェイス (GUI) が低レベルなキー入力の組み合わせを奪い取ってしまうような場合 (たとえば X Window での `Ctrl + Alt + F1` など) は、`sendkey キー指定` コマンドでキー入力を送信することができます:

```
sendkey ctrl-alt-f1
```

`キー指定` オプションで使用するキー名の一覧を表示するには、`sendkey` コマンドを入力して `Tab` を押します。

マウスを制御したい場合は、下記のコマンドを利用することができます:

```
mouse_move dx dy [dz]
```

`dx`, `dy` で指定する座標にマウスのポインタを移動します。`dz` は任意指定で、スクロールホイールの移動量を指定します。

```
mouse_button 値
```

マウスボタンの状態を変更します (1=左ボタン, 2=中央ボタン, 4=右ボタン)。

```
mouse_set インデックス
```

どのマウスデバイスがイベントを受け取るかを指定します。インデックスで指定する値は、`info mice` コマンドで取得することができます。

13.6 利用可能なメモリ量の変更

-balloon virtio オプションを指定して仮想マシンを起動し、擬似仮想化バルーンデバイスが有効化された場合は、動的にメモリ量の割り当てを変更することができます。バルーンデバイスの有効化について、詳しくは 11.1 項「qemu-kvm を利用した基本インストール」(112 ページ)をお読みください。

バルーンデバイスに関する情報をモニタコンソール内で表示したい場合や、どのデバイスが有効になっているのかを判断したい場合は、info balloon コマンドを利用します:

```
(qemu) info balloon
```

バルーンデバイスが有効になっている場合は、balloon メガバイト コマンドを実行して、メモリ量の割り当てを指定することができます:

```
(qemu) balloon 400
```

13.7 仮想マシンのメモリダンプ

仮想マシン内のメモリをディスクやコンソール出力に保存したい場合は、下記のコマンドを実行します:

memsave アドレスサイズファイル名

仮想メモリのダンプを指定した **アドレス** から開始し、**サイズ** で指定したサイズ分だけ **ファイル名** のファイルに保存します。

pmemsave アドレスサイズファイル名

物理メモリのダンプを指定した **アドレス** から開始し、**サイズ** で指定したサイズ分だけ **ファイル名** のファイルに保存します。

x / フォーマットアドレス

仮想メモリのダンプを **フォーマット** で指定した形式で、**アドレス** のアドレスから開始します。フォーマット は 3 つのパラメータから構成される文字列で、**カウント形式サイズ** を指定します:

カウント には、ダンプされる項目数を指定します。

形式 には、x (16 進数), d (符号付き 10 進数), u (符号無し 10 進数), o (8 進数), c (キャラクタ), i (アセンブラ命令) のいずれかを指定します。

サイズには、b (8 ビット), h (16 ビット), w (32 ビット), g (64 ビット) のいずれかを指定します。x86 環境では、h や w は i を併記することもでき、この場合はそれぞれ 16 ビットや 32 ビットのコード 命令サイズを選択することができます。

xp / フォーマットアドレス

物理メモリのダンプを フォーマット で指定した 形式で、アドレス のアドレスから開始します。フォーマット は 3 つのパラメータから構成 される文字列で、*カウント形式サイズ* を指定します:

カウント には、ダンプされる項目数を指定します。

形式 には、x (16 進数), d (符号付き 10 進数), u (符号無し 10 進数), o (8 進数), c (キャラクタ), i (アセンブラ命令) のいずれかを指定します。

サイズには、b (8 ビット), h (16 ビット), w (32 ビット), g (64 ビット) のいずれかを指定します。x86 環境では、h や w は i を併記することもでき、この場合はそれぞれ 16 ビットや 32 ビットのコード 命令サイズを選択することができます。

13.8 仮想マシンのスナップショット管理

警告

QEMU モニタを利用したスナップショット管理の機能は、SUSE では公式に サポートはしていませんが、下記の情報は特定の用途で役立つものであるため、説明のみ行なっているものです。

仮想マシンのスナップショットとは、CPU や RAM の状態、および全ての書き込み可能な ディスクの内容を含む、その時点での完全な仮想マシンの状態を保存するための仕組み です。仮想マシンのスナップショット機能を利用するには、少なくとも 1 台以上の 固定された (リムーバブルでない) 書き込み可能なブロックデバイスが必要で、このディスクは qcow2 形式で存在していなければなりません。

スナップショットは、お使いの仮想マシンを特定の状態で保存しておきたい場合に便利な 仕組みです。たとえば仮想サーバ内でネットワークサービスを設定し、その時点の状態を 保存しておいて、全く同じ状態に素早く復元したいような場合に利用します。またスナップショットは、何か実験的なことを行なって VM ゲスト を不安定にさせてしまうようなことを行なうような場合、事前にいったん仮想マシンの電源を落としてから その時点でのバックアップを作成するような用途にも利用することができます。この章では前者のような場合について手順を説明しています。後者のよ

うな用途の場合は、11.2.3項「qemu-img による仮想マシンのスナップショット管理」(118 ページ) をお読みください。

QEMU モニタでは、スナップショットを管理するのに下記のようなコマンドを利用することができます：

`savevm 名前`

`名前` でタグ付けした名前で、新しい仮想マシンの スナップショットを作成します。なお同名のスナップショットが存在した場合は、上書きされます。

`loadvm 名前`

`名前` でタグ付けした名前のスナップショットを 読み込みます。

`delvm`

仮想マシンのスナップショットを削除します。

`info snapshots`

利用可能なスナップショットについて、情報を表示します。

(qemu) info snapshots

Snapshot list:

ID ^❶	TAG ^❷	VM SIZE ^❸	DATE ^❹	VM CLOCK ^❺
1	booting	4.4M	2010-11-22 10:51:10	00:00:20.476
2	booted	184M	2010-11-22 10:53:03	00:02:05.394
3	logged_in	273M	2010-11-22 11:00:25	00:04:34.843
4	ff_and_term_running	372M	2010-11-22 11:12:27	00:08:44.965

- ❶ スナップショットの識別番号です。通常は自動的に加算される値です。
- ❷ スナップショットの識別用文字列です。ID を人間にとって読みやすい形にしたものと言えます。
- ❸ スナップショットが占有するディスク領域を示しています。実行中のアプリケーションが 占有するメモリサイズが大きければ大きいほど、スナップショットのサイズも 大きくなります。
- ❹ スナップショットを作成した日時を示しています。
- ❺ 仮想マシンの時計に関する状態を示しています。

13.9 仮想マシンの一時停止 (サスペンド) と復元

下記に示すコマンドが、それぞれ仮想マシンを一時停止 (サスペンド) させたり 復元させたりするためのものです：

stop

仮想マシンの実行を一時停止 (サスペンド) します。

cont

仮想マシンの実行を再開 (レジューム) します。

system_powerdown

マシンに対して ACPI シャットダウン要求を送信します。実際のマシンで言うところの 電源ボタンに似た動作になります。

q または quit

QEMU を即時に終了します。

13.10 ライブマイグレーション

ライブマイグレーション処理は、任意の仮想マシンを一方のホストシステムから他方のホストシステムに移行するための仕組みで、可用性を一切損なうことがありません。移行の処理は恒久的に行なうこともできますし、メンテナンスなどの理由で一時的な移行を行なうこともできます。なお、移行元と移行先の各ホストは同じアーキテクチャであることがお勧めですが、AMD と Intel アーキテクチャの間でも移行できます。

ライブマイグレーションの要件は下記のとおりです：

- ライブマイグレーションは、同じ CPU 機能を持つ VM ホストサーバ 同士でのみ実現 できます。ライブマイグレーション対応の CPU モデルは、`-cpu qemu64` (既定値) のみで、追加の機能が何も指定 されていない場合のみです。
- 物理デバイスをホストからゲストにパススルーすることはできません。
- VM ホストサーバと VM ゲスト は、それぞれ適切な時刻維持機能がインストール されている必要があります。
- AHCI インターフェイス, `virtfs` 機能, `-mem-path` コマンドラインオプションは、それぞれマイグレーションとの互換性が ありません。
- SP3 が動作しているホスト上のゲストを、SP2 や SP1 の動作している ホストに移行することはできません。
- 仮想マシンのイメージは、移行元と移行先の両方からアクセス可能でなければなりません。たとえば共有の NFS ディスクなどに配置されている場合などが 該当します。

- イメージのディレクトリは、両方のホストで同じパスに配置されている必要があります。
- 両方のホストは同じサブネット内に存在していなければなりません。
- 移行元と移行先のゲストは、いずれも同じ方法で起動されていなければなりません。

ライブマイグレーションは下記の手順で行ないます:

- 1 まずは移行元の仮想マシンを動作中の状態にします。
- 2 次に仮想マシンを移行先のホストで起動し、一時停止させて移行を受け付けられるモードにします。起動時のパラメータは移行元のホストと同じもののほか、`-incoming tcp:IP アドレス:ポート` を追加で指定します。ここで *IP アドレス* にはライブマイグレーションのデータを受け付ける IP アドレスを、*ポート* には TCP ポートをそれぞれ指定します。IP アドレスに 0 を指定した場合は、仮想マシンは全てのインターフェイスでライブマイグレーションを受け付けるようになります。
- 3 次に移行元のホストでモニタコンソールを表示させ、`migrate -d tcp:移行先 IP アドレス:ポート` コマンドを入力して、移行作業を開始します。
- 4 移行状況を表示するには、移行元のホストにあるモニタコンソールから、`info migrate` コマンドを実行します。
- 5 移行作業をキャンセルするには、移行元のホストにあるモニタコンソールから、`migrate_cancel` コマンドを実行します。
- 6 また、移行作業での最大ダウンタイム (サービス停止時間) を指定するには、`migrate_set_downtime 秒数` コマンドを実行します。
- 7 移行処理の最大転送速度を [バイト毎秒] の単位で指定するには、`migrate_set_speed 速度` コマンドを実行します。

補足

A.1 擬似仮想化ドライバのインストール

A.1.1 Microsoft Windows* への virtio ドライバのインストール

Microsoft Windows 環境では、インストール中に擬似仮想化のドライバを設定することはできません。これは、インストールの際に擬似仮想化ハードディスクから起動しようとしても、拒否されてしまうためです。そのため、擬似仮想化ドライバはインストール済みの Windows 環境に対して追加することになります。

下記の手順では、インストール済みの Windows が単一の IDE ハードディスク上に存在していて、ネットワークアダプタが 1 枚だけ接続されている場合を想定しています。Windows 向けの virtio ドライバを含む ISO イメージは kvm パッケージ内に含まれていて、KVM ホストの `/usr/share/qemu-kvm/win-virtio-drivers.iso` に配置されます。8.4項「Virtual Machine Manager を利用したフロッピーディスクまたは CD/DVD-ROM メディアの取り出しと交換」(91 ページ) の手順に従って、ISO イメージを仮想マシン内の CD-ROM として設定してください。お使いの仮想マシンが CD-ROM デバイスの無い設定になっている場合は、8.2項「Virtual Machine Manager を利用した CD/DVD-ROM デバイスの追加」(89 ページ) を参照して CD-ROM デバイスを 2 番目のドライブとして追加してください。

Windows での virtio ドライバの検出

Windows XP 32 ビット版

メモリバルーン: balloon¥install¥XP¥x86¥balloon.inf
ネットワーク: NetKVM¥install¥XP_Win2003¥x86¥netkvm.inf
ストレージ: viostor¥install¥XP¥x86¥viostor.inf

Windows XP 64 ビット版

メモリバルーン: 未対応です
ネットワーク: NetKVM¥install¥XP_Win2003¥amd64¥netkvm.inf
ストレージ: viostor¥install¥XP¥amd64¥viostor.inf

Windows Server 2003 32 ビット版

メモリバルーン: balloon¥install¥Win2003¥x86¥balloon.inf
ネットワーク: NetKVM¥install¥XP_Win2003¥x86¥netkvm.inf
ストレージ: viostor¥install¥Win2003¥x86¥viostor.inf

Windows Server 2003 64 ビット版

メモリバルーン: balloon¥install¥Win2003¥amd64¥balloon.inf
ネットワーク: NetKVM¥install¥XP_Win2003¥amd64¥netkvm.inf
ストレージ: viostor¥install¥XP¥amd64¥viostor.inf

Windows Vista/Server 2008 32 ビット版

メモリバルーン: balloon¥install¥Vista_Win2008¥x86¥balloon.inf
ネットワーク: NetKVM¥install¥Vista_Win2008¥x86¥netkvm.inf
ストレージ: viostor¥install¥Vista_Win2008¥x86¥viostor.inf

Windows Vista/Server 2008 64 ビット版

メモリバルーン: balloon¥install¥Vista_Win2008¥amd64¥balloon.inf
ネットワーク: NetKVM¥install¥Vista_Win2008¥amd64¥netkvm.inf
ストレージ: viostor¥install¥Vista_Win2008¥amd64¥viostor.inf

Windows 7 32 ビット版

メモリバルーン: balloon¥install¥Win7¥x86¥balloon.inf
ネットワーク: NetKVM¥install¥Win7¥x86¥netkvm.inf
ストレージ: viostor¥install¥Win7¥x86¥viostor.inf

Windows 7 64 ビット版

メモリバブルーン: balloon¥install¥Win7¥amd64¥balloon.inf

ネットワーク: NetKVM¥install¥Win7¥amd64¥netkvm.inf

ストレージ: viostor¥install¥Win7¥amd64¥viostor.inf

A.1.1.1 Windows 7

下記では、Windows 7 に対して擬似仮想化ストレージドライバとネットワークドライバをインストールするための手順を示しています。なお、ストレージドライバをインストールする場合、下記に示す手順は **必ず** 守ってください。手順から外れたことをしてしまうと全く起動できなくなってしまう 場合があるほか、「ブルースクリーン」になってしまう場合もあります！

重要: 技術サポートについて

下記に示す手順では、`virsh edit` コマンドを使用して行ないます。このコマンドは原則として SUSE で技術的に サポートしていないものですが、このような特殊な用途 (Windows への擬似仮想化 ストレージドライバのインストール) は本規則の例外として扱われ、適切な 範囲でのサポートが提供されます。

手順 A.1 Windows 7 32 ビット版に対する擬似仮想化ストレージドライバのインストール

- 1 Windows 7 VM ゲスト をシャットダウンし、Virtual Machine Manager を利用して `virtio` 形式の追加のハードディスク (擬似仮想化ハードディスク) を設定します。このディスクは一時的に必要なもので、VM ゲスト から後で削除します。
- 2 必要であれば Virtual Machine Manager を利用し、`[Boot Device Order]` (起動デバイス順序) を設定します。この起動デバイス順序は `[Hard Disk]` (ハードディスク) から 始まっていなければならない、それ以外の設定では システムディスクが擬似仮想化されてしまい、起動できなくなってしまう。変更を行なったら変更点を確認して `[Apply]` を押します。それ以外を押してしまうと、設定が保存されません。
- 3 VM ゲスト を再起動します。いったん起動したら、スタートメニューから `[ファイル名を指定して実行]` を選び、`devmgmt.msc` と入力して Enter を押すなどして、`[デバイスマネージャ]` を開きます。
- 4 `[その他のデバイス]` > `[SCSI コントローラ]` の項目を開きます。すると、表示されている項目に感嘆符のマークが付けられ、問題があることを 示しているものが

あります。その項目をマウスの右ボタンで押して、[ドライバソフトウェアの更新] を選択します。

- 5 ドライバをインストールします。[コンピュータを参照してドライバソフトウェアを検索します] を選んでから、[参照] ボタンを押し、お使いのオペレーティングシステムとアーキテクチャに適合した ドライバ CD のディレクトリを選択します (例: viostor¥install¥Win7¥x86¥)。あとはセキュリティ 警告が表示されますので、そのまま [インストール] を押します。
- 6 ドライバのインストールが完了すると、[デバイスマネージャ] 内の [記憶域コントローラ] に、新しい [Novell VirtIO SCSI Adapater] が表示されるようになります。また、[ディスクドライブ] の欄には、一時的に設定した 擬似仮想化ディスクが表示されるようになります。このディスクドライブは [Novell VirtIO SCSI Disk Device] という名称で 表示されます。
- 7 Windows 7 VM ゲスト をシャットダウンし、一時的に設定していた擬似仮想化ディスクを削除します。
- 8 仮想ハードディスクの種類を変更することは、現時点の Virtual Machine Manager ではサポートしていません。そのため、XML 設定ファイルを直接編集する必要があります。端末を開いて下記のコマンドを入力してください (なお *名前* の欄には、お使いの Windows 7 VM ゲスト の名前を指定します)。リモートのホストから作業を行なっているような場合は、-c オプションを利用して接続 URL を指定する必要があります。

```
virsh edit 名前
```

するとエディタ (既定では vi) が表示されます。下記のようなブロックを探してください:

```
<disk type='block' device='disk'>
  <driver name='qemu' type='raw' />
  <source file='/var/lib/libvirt/images/win7.raw' />
  <target dev='hda' bus='ide' />
  <address type='drive' controller='0' bus='0' unit='0' />
</disk>
```

まずは <address> タグを削除します。次に <target> タグの属性を dev='vda' および bus='virtio' に書き換えます:

```
<disk type='block' device='disk'>
  <driver name='qemu' type='raw' />
  <source dev='/dev/Virtual/win7' />
  <target dev='vda' bus='virtio' />
</disk>
```

ファイルを保存します。保存が成功すると、Domain 名前 XML configuration edited. というメッセージが表示されます。何らかのエラーが表示された場合 (たとえば不正な XML が書き込まれたなど) は、設定は変更されません。

- 9 VM ゲスト を再起動します。Virtual Machine Manager 経由で起動している場合は、起動を行なう前に [Details] (詳細) 画面に変更後のハードウェア 設定が表示されていることを確認します (なお、virsh で設定の変更を行なってから、画面に表示されるまでに数秒程度の時間を 必要とします)。変更後の設定が表示されない場合は、Virtual Machine Manager が直近で 使用していた設定で上書きされてしまいます。

これですべての作業は終わりです。お使いの Windows 7 VM ゲスト で 擬似仮想化システムディスクを使用するようになっています。

擬似仮想化ネットワークドライバのインストールは、ストレージドライバの インストールに似ています：

手順 A.2 Windows 7 に対する擬似仮想化ネットワークドライバのインストール

- 1 Windows 7 VM ゲスト をシャットダウンし、Virtual Machine Manager を利用して virtio 形式のネットワークアダプタ (擬似仮想化ネットワークアダプタ) を設定します。追加のハードディスクを設定します。これはドライバをインストールする際に ネットワークの接続を維持するためのものです。
- 2 VM ゲスト を再起動し、上記と同じ手順で [デバイスマネージャ] からドライバをインストールします。なお、新しいネットワークアダプタは [その他のデバイス] > [イーサネットアダプタ] 内に表示されます。ドライバのインストールが完了すると、[デバイスマネージャ] 内の [ネットワークアダプタ] に、新しい [Novell VirtIO Ethernet Adapter] が表示されるようになります。
- 3 Windows 7 VM ゲスト をシャットダウンし、Virtual Machine Manager を利用して元々設定されていた 擬似仮想化でないほうのネットワークアダプタを設定から削除します。これでゲストを再度起動すると、擬似仮想化ネットワークアダプタを利用することができるようになります。

A.1.1.2 その他のバージョンの Windows (XP, Server 2003/2008, Vista)

その他のバージョンの Windows に対して擬似仮想化ドライバをインストール する作業は、Windows 7 の場合 (A.1.1.1項「Windows 7」(169 ページ)) にとても

似ています。なお、デバイスは「デバイスマネージャ」から手作業で開始したりはしないでください。Windows ではドライバの インストールを促すメッセージを表示する仕組みが備わっているためです。また、ドライバインストールの際には手動でドライバの場所を指定してください。

警告: Windows Vista に対する擬似仮想化ストレージドライバ

現時点では、Windows Vista 向けの擬似仮想化ストレージドライバは、擬似仮想化ディスクから起動する機能に対応していません。擬似仮想化ディスクは、起動ディスク以外での使用のみをサポートしています。

注記: Windows XP に対する擬似仮想化ドライバの非推奨について

Windows XP 上での擬似仮想化ストレージドライバの使用は、性能面での利点が 得られないばかりか、場合によっては性能が悪化してしまう場合があります。そのため、この環境では擬似仮想化ドライバを使用することは *お勧めしません*。技術的な詳細については、<http://www.mail-archive.com/kvm@vger.kernel.org/msg22834.html> をお読みください。

また、上記の非推奨は Windows XP 上の擬似仮想化 ストレージドライバのみを対象としたものです。その他のバージョンの Windows では 良好な性能を得ることができますし、Windows XP 上の擬似仮想化ネットワークドライバについても利用する価値があります。

A.2 x509 クライアント／サーバ証明書の生成

x509 のサーバ／クライアント証明書を利用するには、証明機関 (CA) に対して 依頼を行ない、それらを発行してもらう必要があります。libvirt で利用する場合は、独自の 証明機関を構築しておくことをお勧めします。

- 1 まずは 項「ルート CA の作成」(第16章 *X.509 証明書の管理*, ↑セキュリティガイド) の手順に 従い、証明機関を構築します。
- 2 次に 項「ユーザ証明書の作成と失効化」(第16章 *X.509 証明書の管理*, ↑セキュリティガイド) の手順に 従い、サーバ／クライアントの各証明書を作成します。サーバ証明書のコモン ネーム (CN) は完全修飾ドメイン名 (FQDN) でなけ

ればなりませんが、クライアントのコモンネームは自由に設定することができます。その他の項目については、YaST が提示する既定値を設定してください。

作成したサーバ／クライアント証明書を、一時的な場所 (たとえば /tmp/x509/ など) にエクスポートします。具体的には 下記の手順を実施します:

2a [証明書] タブで証明書を選択します。

2b [エクスポート] > [ファイルにエクスポート] > [証明書と鍵を暗号化せずに PEM 形式で] を選択し、[証明書のパスワード] を入力します。また、[ファイル名] の欄には保存先のファイル名をフルパスで指定します。たとえば /tmp/x509/server.pem や /tmp/x509/client.pem のようになります。

2c 端末を開いて証明書を保存したディレクトリに移動し、下記のコマンドを実行して 証明書と鍵を別々のファイルに分離します (下記の例ではサーバの証明書と鍵を 分離しています):

```
csplit -z -f s_ server.pem '/-----BEGIN/' '{1}'  
mv s_00 servercert.pem  
mv s_01 serverkey.pem
```

2d 上記の手順を、それぞれエクスポートしたクライアント／サーバ証明書に対して 繰り返し実施します。

3 最後に証明機関 (CA) の証明書をエクスポートします。下記の手順で行ないます:

3a [説明] タブに切り替えます。

3b [詳細設定] > [ファイルにエクスポート] > [証明書のみを PEM 形式で] を選択し、[ファイル名] の欄に保存先のファイル名をフルパスで 指定します。たとえば /tmp/x509/cacert.pem のようになります。



GNU ライセンス

本付録には、GNU General Public License バージョン 2 と GNU Free Documentation License バージョン 1.2 を掲載しています。

なお、八田真行氏 (mhatta@gnu.org) [<mailto:mhatta@gnu.org>] による各ライセンスの日本語訳を併記しています。

ただし、各日本語訳は *非公式*なものであり、フリーソフトウェア財団 (the Free Software Foundation) によって発表されたものではないことにご注意ください。法的に有効なものは常に原文 (つまり英語版) 側であり、日本語訳は各ライセンスをよりよく理解する支援を行なう目的で 作成されたもの、という扱いです。

また、日本語訳は DocBook (novdoc) に合わせて段落を分割しているほか、引用符のタグ化 ("blah" -> <quote>blah</quote>) とリンクの生成 (ulink) を行なっています。

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The «Program», below, refers to any such program or work, and a «work based on the Program» means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term «modification».) Each licensee is addressed as «you».

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and [any later version], you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the 「copyright」 line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.  
Copyright (C) yyyy name of author
```

```
This program is free software; you can redistribute it and/or  
modify it under the terms of the GNU General Public License  
as published by the Free Software Foundation; either version 2  
of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License  
along with this program; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details  
type `show w' . This is free software, and you are welcome  
to redistribute it under certain conditions; type `show c'  
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a 「copyright disclaimer」 for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
```

interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [<http://www.fsf.org/licenses/lgpl.html>] instead of this License.

GNU 一般公衆利用許諾契約書 (日本語訳)

バージョン 2, 1991年6月

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

この利用許諾契約書を、一字一句そのままに複製し頒布することは許可する。しかし変更は認めない。

はじめに

ソフトウェア向けライセンスの大半は、あなたがそのソフトウェアを共有したり 変更したりする自由を奪うように設計されています。対照的に、GNU 一般公衆利用許諾契約書は、あなたがフリーソフトウェアを共有したり変更したりする自由を保証する--すなわち、ソフトウェアがそのユーザすべてにとってフリー であることを保証することを目的としています。この一般公衆利用許諾契約書は、フリーソフトウェア財団のソフトウェアのほとんどに適用されており、また GNU GPLを適用すると決めたフリーソフトウェア財団以外の作者によるプログラムにも適用されています(いくつかのフリーソフトウェア財団のソフトウェアには、GNU GPLではなくGNU ライブラリー一般公衆利用許諾契約書が適用されています)。あなたもまた、ご自分のプログラムにGNU GPLを適用することが可能です。

私たちがフリーソフトウェアと言うとき、それは利用の自由について言及している のであって、価格は問題にしていません。私たちの一般公衆利用許諾契約書は、あなたがフリーソフトウェアの複製物を頒布する自由を保証するよう設計されています (希望に応じてその種のサービスに手数料を課す自由も保証されます)。また、あなたがソースコードを受け取るか、あるいは望めばそれを入手することが可能であるということ、あなたがソフトウェアを変更し、その一部を新たなフリーの プログラムで利用できるとのこと、そして、以上で述べたようなことができる ということがあなたに知られるということも保証されます。

あなたの権利を守るため、私たちは誰かがあなたの有するこれらの権利を否定する ことや、これらの権利を放棄するよう要求することを禁止するという制限を加える 必要があります。よって、あなたがソフトウェアの複製物を頒布したりそれを変更 したりする場合には、そういった制限のためにあなたにある種の責任が発生することになります。

例えば、あなたがフリーなプログラムの複製物を頒布する場合、有料が無料に 関わらず、あなたは自分が有する権利を全て受領者に与えなければなりません。また、あなたは彼らもソースコードを受け取るか手に入れることができるよう 保証しなければなりません。そして、あなたは彼らに対して以下で述べる条件を示し、彼らに自らの持つ権利について知らしめるようにしなければなりません。

私たちはあなたの権利を二段階の手順を踏んで保護します。(1) まずソフトウェアに対して著作権を主張し、そして (2) あなたに対して、ソフトウェアの複製や頒布または改変についての法的な許可を 与えるこの契約書を提示します。

また、各作者や私たちを保護するため、私たちはこのフリーソフトウェアには 何の保証も無いということを誰もが確実に理解するようにし、またソフトウェアが 誰か他人によって改変され、それが次々と頒布されていったとしても、その受領者は 彼らが手に入れたソフトウェアがオリジナルのバージョンでは無いこと、そして 原作者の名声は他人によって持ち込まれた可能性のある問題によって影響される ことがないということを周知させたいと思います。

最後に、ソフトウェア特許がいかなるフリーのプログラムの存在にも不断の脅威を 投げかけていますが、私たちは、フリーなプログラムの再頒布者が個々に特許 ライセンスを取得することによって、事実上プログラムを独占的にしてしまうという 危険を避けたいと思います。こういった事態を予防するため、私たちはいかなる特許も 誰もが自由に利用できるようライセンスされるか、全くライセンスされないかの どちらかでなければならないことを明確にしました。

(訳注: 本契約書で「独占的(proprietary)」とは、ソフトウェアの利用や再頒布、改変が禁止されているか、許可を得ることが必要とされているか、あるいは厳しい 制限が課せられていて自由にそうすることが事実上できなくなっている状態のことを 指す。詳しくは <http://www.gnu.org/philosophy/categories.ja.html#ProprietarySoftware> [<http://www.gnu.org/philosophy/categories.ja.html#ProprietarySoftware>] を参照せよ。)

複製や頒布、改変についての正確な条件と制約を以下で述べていきます。

複製、頒布、改変に関する条件と制約

0. この利用許諾契約書は、そのプログラム(またはその他の著作物)を この一般公衆利用許諾契約書の定める条件の下で頒布できる、という告知が 著作権者によって記載されたプログラムまたはその他の著作物全般に適用される。以下では、「プログラム」とはそのようにしてこの契約書が適用され プログラムや著作物全般を意味し、また「プログラムを基にした著作物」とは「プログラム」やその他の著作権法の下で派生物と見なされるもの 全般を指す。すなわち、「プログラム」かその一部を、全く同一の ままか、改変を加えたか、あるいは他の言語に翻訳された形で含む

著作物のことである(「改変」という語の本来の意味からはずれるが、以下では 翻訳も改変の一種と見なす)。それぞれの契約者は「あなた」と表現される。

複製や頒布、改変以外の活動はこの契約書ではカバーされない。それらはこの契約書の 対象外である。「プログラム」を実行する行為自体に制限はない。また、そのような「プログラム」の出力結果は、その内容が「プログラム」を基にした著作物を構成する場合のみ この契約書によって保護される(「プログラム」を実行したことによって 作成されたということは無関係である)。このような線引きの妥当性は、「プログラム」が何を
するのかに依存する。

1. それぞれの複製物において適切な著作権表示と保証の否認声明(disclaimer of warranty) を目立つよう適切に掲載し、またこの契約書および一切の保証の不在に触れた 告知すべてをそのまま残し、そしてこの契約書の複製物を「プログラム」のいかなる受領者にも「プログラム」と共に頒布する限り、あなたは「プログラム」のソースコードの複製物を、あなたが受け取った通りの形で複製または頒布することができる。媒体は問わない。

あなたは、物理的に複製物を譲渡するという行為に関して手数料を課しても良いし、希望によっては手数料を取って交換における保護の保証を提供しても良い。

2. あなたは自分の「プログラム」の複製物がその一部を改変して「プログラム」を基にした著作物を形成し、そのような改変点や 著作物を上記第1節の定める条件の下で複製または頒布することができる。ただし、そのためには以下の条件すべてを満たしていなければならない:

- a) あなたがそれらのファイルを変更したということと変更した日時が良く 分かるよう、改変されたファイルに告示しなければならない。
- b) 「プログラム」またはその一部を含む著作物、あるいは「プログラム」かその一部から派生した著作物を頒布あるいは 発表する場合には、その全体をこの契約書の条件に従って第三者へ無償で 利用許諾しなければならない。
- c) 改変されたプログラムが、通常実行する際に対話的にコマンドを読むようになっているならば、そのプログラムを最も一般的な方法で対話的に実行する際、適切な著作権表示、無保証であること(あるいはあなたが保証を提供するということ)、ユーザがプログラムをこの契約書で述べた条件の下で頒布することができる ということ、そしてこの契約書の複製物を閲覧するにはどうしたらよいかというユーザへの説明を含む告知が印刷されるか、あるいは画面に表示される ようにしなければならない(例外として、「プログラム」そのものは対話的であっても通常そのような告知を印刷しない場合には、「プログラム」を基にしたあなたの著作物に そのような告知を 印刷させる必要はない)。

以上の必要条件是全体としての改変された著作物に適用される。著作物の一部が「プログラム」から派生したのではないと確認でき、それら自身 別の独立した著作物であると合理的に考えられるならば、あなたがそれらを別の 著作物として分けて頒布する場合、そういった部分にはこの契約書とその条件は 適用されない。しかし、あなたが同じ部分を「プログラム」を基にした 著作物全体の一部として頒布するならば、全体としての頒布物は、この契約書が 課す条件に従わなければならない。というのは、この契約書が他の契約者に与える 許可は「プログラム」丸ごと全体に及び、誰が書いたかは関係なく 各部分のすべてを保護するからである。

よって、すべてあなたによって書かれた著作物に対し、権利を主張したりあなたの 権利に異議を申し立てることはこの節の意図するところではない。むしろ、その趣旨は「プログラム」を基にした派生物ないし集合著作物の 頒布を管理する権利を行使することにある。

また、「プログラム」を基にしていないその他の著作物を「プログラム」(あるいは「プログラム」を基にした著作物)と一緒に集めただけのものを一巻の保管装置ないし頒布媒体に収めても、その他の 著作物までこの契約書が保護する対象になるということにはならない。

3. あなたは上記第1節および2節の条件に従い、「プログラム」(あるいは 第2節における派生物)をオブジェクトコードないし実行形式で複製または頒布 することができる。ただし、その場合あなたは以下のうちどれか一つを実施 しなければならない:

- a) 著作物に、『プログラム』に対応した完全かつ機械で読み取り可能な ソースコードを添付する。ただし、ソースコードは上記第1節および2節の条件に従いソフトウェアの交換で習慣的に使われる媒体で頒布しなければならない。あるいは、
- b) 著作物に、いかなる第三者に対しても、『プログラム』に対応した完全かつ 機械で読み取り可能なソースコードを、頒布に要する物理的コストを上回らない 程度の手数料と引き換えに提供する旨述べた少なくとも3年間は有効な書面 になった申し出を添える。ただし、ソースコードは上記第1節および2節の条件に 従いソフトウェアの交換で習慣的に使われる媒体で頒布しなければならない。あるいは、
- c) 対応するソースコード頒布の申し出に際して、あなたが得た情報を一緒に引き渡す (この選択肢は、営利を目的としない頒布であって、かつあなたが上記小節bで 指定されているような申し出と共にオブジェクトコードあるいは実行形式の プログラムしか入手していない場合に限り許可される)。

著作物のソースコードとは、それに対して改変を加える上で好ましいとされる 著作物の形式を意味する。ある実行形式の著作物にとって完全なソースコードとは、それが含むモジュールすべてのソースコード全部に加え、関連するインターフェース 定義ファイルのすべてとライブラリのコンパイルやインストールを制御するために 使われるスクリプトをも加えたものを意味する。しかし特別な例外として、そのコンポーネント自体が実行形式に付随するのでは無い限り、頒布されるものに 中、実行形式が実行されるオペレーティングシステムの主要なコンポーネント (コンパイラやカーネル等)と通常一緒に(ソースかバイナリ形式のどちらかで) 頒布されるものを含んでいる必要はないとする。

実行形式またはオブジェクトコードの頒布が、指定された場所からコピーするための アクセス手段を提供することで為されるとして、その上でソースコードも同等の アクセス手段によって同じ場所からコピーできるようになっているならば、第三者が オブジェクトコードと一緒にソースも強制的にコピーせられるようになっていなくてもソースコード頒布の条件を満たしているものとする。

4. あなたは「プログラム」を、この契約書において明確に提示された 行為を除き複製や改変、サブライセンス、あるいは頒布してはならない。他に「プログラム」を複製や改変、サブライセンス、あるいは頒布する 企てはすべて無効であり、この契約書の下でのあなたの権利を自動的に終結させる こととなる。しかし、複製物や権利をこの契約書に従ってあなたから得た人々に 関しては、そのような人々がこの契約書に完全に従っている限り彼らのライセンスまで 終結することはない。
5. あなたはこの契約書を受諾する必要は無い。というのは、あなたはこれに署名して いないからである。しかし、この契約書以外にあなたに対して「プログラム」やその派生物を改変または頒布する許可を与えるものは 存在しない。これらの行為は、あなたがこの契約書を受け入れない限り

法によって 禁じられている。そこで、「プログラム」(あるいは「プログラム」を基にした著作物全般)を改変しない頒布することにより、あなたは自分がそのような行為を行うためにこの契約書を受諾したということ、そして「プログラム」とそれに基づく著作物の複製や頒布、改変について この契約書が課す制約と条件をすべて受け入れたということを示したものと見なす。

6. あなたが「プログラム」(または「プログラム」を基にした著作物全般)を再頒布するたびに、その受領者は元々のライセンス許可者から、この契約書で指定された条件と制約の下で「プログラム」を複製や頒布、あるいは改変する許可を自動的に得るものとする。あなたは、受領者がここで認められた権利を行使することに関してこれ以上他のいかなる制限も課してはならない。あなたには、第三者がこの契約書に従うことを強制する責任はない。

7. 特許侵害あるいはその他の理由(特許関係に限らない)から、裁判所の判決あるいは申し立ての結果としてあなたに(裁判所命令や契約などにより)このライセンスの条件と矛盾する制約が課された場合でも、あなたがこの契約書の条件を免除されるわけではない。もしこの契約書の下であなたに課せられた責任と他の関連する責任を同時に満たすような形で頒布できないならば、結果としてあなたは「プログラム」を頒布することが全くてできないことである。例えば特許ライセンスが、あなたから直接間接を問わずコピーを受け取った人が誰でも「プログラム」を使用料無料で再頒布することを認めていない場合、あなたがその制約とこの契約書を両方とも満たすには「プログラム」の頒布を完全に中止するしかないだろう。

この節の一部分が特定の状況の下で無効ないし実施不可能な場合でも、節の残りの部分は適用されるよう意図されている。その他の状況では節が全体として適用されるよう意図されている。

特許やその他の財産権を侵害したり、そのような権利の主張の効力に異議を唱えたりするようあなたを誘惑することがこの節の目的ではない。この節には、人々によってライセンス慣行として実現されてきた、フリーソフトウェア頒布のシステムの完全性を護るという目的しかない。多くの人々が、フリーソフトウェアの頒布システムが首尾一貫して適用されているという信頼に基づき、このシステムを通じて頒布される多様なソフトウェアに寛大な貢献をしてきたのは事実であるが、人がどのようなシステムを通じてソフトウェアを頒布したいと思うかはあくまでも作者/寄与者次第であり、あなたが選択を押しつけることはできない。

この節は、この契約書のこの節以外の部分の一帰結になると考えられるケースを徹底的に明らかにすることを目的としている。

8. 「プログラム」の頒布や利用が、ある国においては特許または著作権が主張されたインターフェースのいずれかによって制限されている場合、「プログラム」にこの契約書を適用した元の著作権者は、そういった国々を排除した明確な地理的頒布制限を加え、そこで排除されていない国の中やそれらの国々の間でのみ頒布が許可されるようにしても構わない。その場合、そのような制限はこの契約書本文で書かれているのと同様に見なされる。

9. フリーソフトウェア財団は、時によって改訂または新版の一般公衆利用許諾書を発表することができる。そのような新版は現在のバージョンとその精神においては似たものになるだろうが、新たな問題や懸念を解決するため細部では異なる可能性がある。

それぞれのバージョンには、見分けが付くようにバージョン番号が振られている。「プログラム」においてそれに適用されるこの契約書のバージョン番号が指定されていて、更に「それ以降のいかなるバージョン(any later version)」も適用して良いとなった場合、あなたは従う条件と制約として、指定のバージョンか、フリーソフトウェア財団によって発行された指定のバージョン以降の版のどれか一つのどちらかを選ぶことが出来る。「プログラム」でライセンスのバージョン番号が指定されていないならば、あなたは今までにフリーソフトウェア財団から発行されたバージョンの中から好きに選んで構わない。

10. もしあなたが「プログラム」の一部を、その頒布条件がこの契約書と異なる他のフリーなプログラムと統合したいならば、作者に連絡して許可を求めよ。フリーソフトウェア財団が著作権を保有するソフトウェアについては、フリーソフトウェア財団に連絡せよ。私たちは、このような場合のために特別な例外を設けることもある。私たちが決定を下すにあたっては、私たちのフリーソフトウェアの派生物すべてがフリーな状態に保たれるということと、一般的にソフトウェアの共有と再利用を促進するという二つの目標を規準に検討されるであろう。

無保証について

11. 「プログラム」は代価無しに利用が許可されるので、適切な法が認める限りにおいて、「プログラム」に関するいかなる保証も存在しない。書面で別に述べる場合を除いて、著作権者、またはその他の団体は、「プログラム」を、表明されたか言外にかかわらず、商業的適性を保証するほのめかしやある特定の目的への適合性(に限られない)を含む一切の保証無しに「あるがまま」で提供する。「プログラム」の質と性能に関するリスクのすべてはあなたに帰属する。「プログラム」に欠陥があると判明した場合、あなたは必要な保守点検や補修、修正に要するコストのすべてを引き受けることになる。

12. 適切な法が書面で同意によって命ぜられない限り、著作権者、または上記で許可されている通りに「プログラム」を改変または再頒布したその他の団体は、あなたに対して「プログラム」の利用ないし利用不能で生じた通常損害や特別損害、偶発損害、間接損害(データの消失や不正確な処理、あなたが第三者が被った損失、あるいは「プログラム」が他のソフトウェアと一緒に動作しないという不具合などを含むがそれらに限らない)に一切の責任を負わない。そのような損害が生ずる可能性について彼らが忠告されていたとしても同様である。

条件と制約終わり

以上の条項をあなたの新しいプログラムに適用する方法

あなたが新しいプログラムを開発したとして、公衆によってそれが利用される可能性を最大にしたいなら、そのプログラムをこの契約書の条項に従って誰でも再頒布あるいは変更できるようフリーソフトウェアにするのが最善です。

そのためには、プログラムに以下のような表示を添付してください。その場合、保証が排除されているということを最も効果的に伝えるために、それぞれのソースファイルの冒頭に表示を添付すれば最も安全です。少なくとも、「著作権表示」という行と全文がある場所へのポインタだけは各ファイルに含めて置いてください。

one line to give the program's name and an idea of what it does.
Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

(訳)

プログラムの名前と、それが何をするかについての簡単な説明。Copyright (C) 西暦年 作者の名前

このプログラムはフリーソフトウェアです。あなたはこれを、
フリーソフトウェア財団によって発行された GNU 一般公衆利用許諾契約書
(バージョン2か、希望によってはそれ以降のバージョンのうちどれか)の
定める条件の下で再頒布または改変することができます。

このプログラムは有用であることを願って頒布されますが、*全くの無保証*
です。商業可能性の保証や特定の目的への適合性は、言外に示されたものも
含め全く存在しません。詳しくはGNU 一般公衆利用許諾契約書をご覧ください。

あなたはこのプログラムと共に、GNU 一般公衆利用許諾契約書の複製物を一部
受け取ったはずですが、もし受け取っていない場合は、フリーソフトウェア財団
まで請求してください(宛先は the Free Software Foundation, Inc., 59
Temple Place, Suite 330, Boston, MA 02111-1307 USA)。

電子ないし紙のメールであなたに問い合わせる方法についての情報も書き加えましょう。

プログラムが対話的なものならば、対話モードで起動した際に出力として 以下のような短い告知が表示されるようにしてください:

Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.

(訳)

Gnomovision バージョン 69, Copyright (C) 西暦年 作者の名前
Gnomovision は*全くの無保証*で提供されます。詳しくは
`show w' とタイプして下さい。
これはフリーソフトウェアであり、ある条件の下で再頒布することが
奨励されています。詳しくは `show c' とタイプして下さい。

ここで、仮想的なコマンド `show w' と `show c' は 一般公衆利用許諾契約書の適切な部分を表示するようになっていなければなりません。
もちろん、あなたが使うコマンドを `show w' や `show c' と呼ぶ必然性はありませんので、あなたのプログラムに 合わせてマウスのクリックやメ
ニューのアイテムにしても結構です。

また、あなたは、必要ならば(プログラマーとして働いていたら)あなたの 雇用主、あるいは場合によっては学校から、そのプログラムに関する「著作権放棄声明(copyright disclaimer)」に署名してもらうべきです。以下は例ですので、名前を変えてください:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.
```

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

(訳)

Yoyodyne社はここに、James Hackerによって書かれた
プログラム `Gnomovision' (コンパイラへ通すプログラム)
に関する一切の著作権の利益を放棄します。

Ty Coon氏の署名、1989年4月1日
Ty Coon、副社長

この一般公衆利用許諾契約書では、あなたのプログラムを独占的なプログラムに 統合することを認めていません。あなたのプログラムがサブルーチンライブラリ ならば、独占的なアプリケーションとあなたのライブラリをリンクすることを 許可したほうがより便利であると考えられるかもしれません。もしこれがあなたの 望むことならば、この契約書の代わりに GNU ライブラリー一般公衆利用許諾契約書 [<http://www.fsf.org/licenses/lgpl.html>] を適用してください。

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member

of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and

legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O.Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the

aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

GNU フリー文書利用許諾契約書

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA この利用許諾契約書を、一字一句そのままに複製し頒布することは許可する。しかし変更は認めない。

This is an unofficial translation of the GNU Free Documentation License into Japanese. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documents that uses the GNU FDL--only the original English text of the GNU FDL does that. However, we hope that this translation will help Japanese speakers understand the GNU FDL better.

(訳: 以下はGNU Free Documentation Licenseの非公式な日本語訳です。これはフリーソフトウェア財団 (the Free Software Foundation)によって発表されたものではなく、GNU FDLを適用した文書の頒布条件を法的に有効な形で述べたものではありません。頒布条件としてはGNU FDLの英語版テキストで指定されているもののみが有効です。しかしながら、私たちはこの翻訳が、日本語を使用する人々にとってGNU FDLをより良く理解する助けとなることを望んでいます。)

0. はじめに

この利用許諾契約書の目的は、この契約書が適用されるマニュアルや教科書、その他機能本位で実用的な文書を(無料ではなく)自由という意味で「フリー」とすること、すなわち、改変の有無あるいは目的の営利非営利を問わず、文書を複製し再頒布する自由をすべての人々に効果的に保証することです。加えてこの契約書により、著者や出版者が自分たちの著作物に対して相応の敬意と賞賛を得る手段も保護されます。また、他人が行った改変に対して責任を負わずに済むようになります。

この利用許諾契約書は「コピーレフト」的なライセンスの一つであり、この契約書が適用された文書から派生した著作物は、それ自身もまた原本と同じ意味でフリーでなければなりません。この契約書は、フリーソフトウェアのために設計されたコピーレフトなライセンスであるGNU一般公衆利用許諾契約書を補足するものです。

(訳注: コピーレフト(copyleft)の概念については <http://www.gnu.org/copyleft/copyleft.ja.html> を参照せよ)

この利用許諾契約書は、フリーソフトウェア用のマニュアルに適用することを目的として書かれました。フリーソフトウェアはフリーな文書を必要としており、フリーなプログラムはそのソフトウェアが保証するのと同じ自由を提供するマニュアルと共に頒布されるべきだからです。しかし、この契約書の適用範囲はソフトウェアのマニュアルに留まりません。対象となる著作物において扱われる主題が何であれ、あるいはそれが印刷された書籍として出版されるか否かに関わらず、この契約書は文字で書かれたいかなる著作物にも適用することが可能です。私たちとしては、主にこの契約書を解説や参照を目的とする著作物に適用することをお勧めします。

1. この利用許諾契約書の適用範囲と用語の定義

著作物がこの利用許諾契約書の定める条件の下で頒布される旨の告知を、著作権者がその中に書いたすべてのマニュアルあるいはその他の著作物は、いかなる媒体上にあってもこの契約書の適用対象となる。そのような告知を置くことで、全世界において、著作権使用料を必要とせず、許可の存続期間を限定されること無く、この契約書の中で述べられている条件の下で当該著作物を利用できるという許可を与えることとする。以下において、「『文書』(Document)」とはそのような告知が記載されたマニュアルないし著作物すべてを指す。公衆の一員ならば誰でも契約の当事者となることができ、この契約書中では「あなた」と表現される。あなたは、著作権法の下で許可を必要とするような方法で著作物を複製や改変、あるいは頒布することにより、この契約書を受諾することになる。

『文書』の「改変版 (Modified Version)」とは、一字一句忠実に複製したが、あるいは改変や他言語への翻訳を行ったかどうかに関わらず、その『文書』の全体あるいは一部分を含む著作物すべてを意味する。

「補遺部分 (Secondary Section)」とは、『文書』中でその旨指定された補遺ないし本文に先だって前付けとして置かれる一部分であり、『文書』の出版者あるいは著者と、『文書』全体の主題 (あるいはそれに関連する事柄)との関係のみを論じ、全体としての主題の範疇に直接属する内容を

全く含まないものである (たとえば、『文書』の一部が数学の教科書だった場合、補遺部分では数学について何も解説してはならない)。補遺部分で扱われる関係は、その主題あるいは関連する事柄との歴史的つながりのことかも知れないし、それらに関する法的、商業的、哲学的、倫理的、あるいは政治的立場についてかも知れない。

「変更不可部分 (Invariant Sections)」とは補遺部分の一種で、それらが変更不可部分であることが、『文書』をこの利用許諾契約書の下で発表する旨述べた告知中においてその部分の題名と共に明示されているものである。ある部分が上記のような「補遺」性の定義にそぐわない場合は、その部分を「変更不可」として指定することは認められない。『文書』は、変更不可部分を全く含まなくても良い。『文書』において変更不可部分が全く指定されていないければ、その『文書』に変更不可部分は存在しないということである。

「カバーテキスト (Cover Texts)」とは、『文書』がこの利用許諾契約書の指定する条件の下で発表される旨述べた告知において、「表カバーテキスト」あるいは「裏カバーテキスト」として列挙された短い文章のことを指す。表カバーテキストは最大で5語、裏カバーテキストは最大で25語までとする。

『文書』の「透過的」複製物とは、機械による読み取りが可能な『文書』の複製物のことを指す。透過的な複製物の文書形式は、その仕様が一般の人々に入手可能で、『文書』の内容を一般的なテキストエディタ、または(画素で構成される画像ならば)一般的なペイントプログラム、あるいは(図面ならば)いくつかの広く入手可能な製図エディタで簡単に改訂するのに適しており、なおかつテキストフォーマットへの入力に適する(あるいはテキストフォーマットへの入力に適する諸形式への自動的な変換に適する)ものでなければならない。透過的なファイル形式への複製であっても、マークアップ、あるいはマークアップの不在が読者によるそれ以降の改変をわざと邪魔し阻害するように仕組まれたものは透過的であるとは見做されない。ある画像形式が、相当量のテキスト文章を表現するために使われた場合、それは透過的ではない。透過的ではない複製は「非透過的」複製と呼ばれる。

透過的複製に適した形式の例としては、マークアップを含まないプレーンな ASCII 形式、Texinfo 入力形式、LaTeX 入力形式、一般に入手可能な DTD を用いた SGML あるいは XML、または人間による改変を想定して設計された、標準に準拠したシンプルなもの HTML や PostScript、PDF などが挙げられる。透過的な画像形式の例には、PNG や XCF、JPG が含まれる。非透過な形式としては、独占的なワードプロセッサでのみ閲覧編集できる独占的なファイル形式、普通には入手できない DTD または処理系を使った SGML や XML、ある種のワードプロセッサが生成する、出力のみを目的とした機械生成の HTML や PostScript、PDF などが含まれる。

「題扉 (Title Page)」とは、印刷された書籍に於いては、実際の表紙自身のみならず、この利用許諾契約書が表紙に掲載することを義務づける文章や図などを、読みやすい形で載せるのに必要なだけの、表紙に引き続き数ページをも意味する。表紙に類するものが無い形式で発表される著作物においては、「題扉」とは本文の始まりに先だって、その著作物の題名が最も目立つ形で現れる場所の近くに置かれる文章のことを指す。

「XYZ」と題された (Entitled XYZ) 部分とは、『文書』において「XYZ」と名付けられた一部分であり、その題名は正確に「XYZ」であるか、「XYZ」を他の言語に翻訳した上でその後ろに「XYZ」をそのまま括弧で括ったものを含む記述のどちらかである(ここでの「XYZ」とは、この利用許諾契約書において以下で言及される特定の部分名を意味している。例えば「謝辞 (Acknowledgements)」、「献辞 (Dedications)」、「推薦の辞 (Endorsements)」、「履歴 (History)」)。あなたが『文書』を改変する場合、そのような部分の「題名を保存する (Preserve the Title)」とは、「XYZ」と題された」部分として、ここでの定義に従い題名を残すということである。

『文書』は、「保証否認警告 (Warranty Disclaimers)」を、この利用許諾契約書が『文書』に適用されると述べた告知の次に含んでも良い。この種の保証否認警告は、この契約書からの言及という形で利用条件に含まれるものと解されるが、保証の否認に関することについてののみ有効とする。こういった保証否認警告で示うその他のいかなる含意も無効であり、この契約書の効能には何ら影響を持たない。

2. 逐語的に忠実な複製

この利用許諾契約書、著作権表示、この契約書が『文書』に適用される旨述べた告知の三つがすべての複製物に複製され、かつあなたがこの契約書で指定されている以外のいかなる条件も追加しない限り、あなたはこの『文書』を、商用であるか否かを問わずいかなる形で複製頒布することができる。あなたは、あなたが作成あるいは頒布する複製物に対して、閲覧や再複製を技術的な手法によって妨害、規制してはならない。しかしながら、複製と引き換えに代価を得てもかまわない。あなたが相当量の複製物を頒布する際には、本契約書第3項で指定される条件にも従わなければならない。

またあなたは、上記と同じ条件の下で、複製物を貸与したり複製物を公に開示することができる。

3. 大量の複製

もしあなたが、『文書』の印刷された (あるいは通常は印刷された表紙を持つ媒体における)複製物を100部を超えて出版し、また『文書』の利用許諾告知がカバーテキストの掲載を要求している場合には、指定されたすべてのカバーテキストを、表カバーテキストは表表紙に、裏カバーテキストは裏表紙に、はっきりと読みやすい形で載せた表紙の中に複製物本体を綴じ込まなければならない。また、両方の表紙において、それらの複製物の出版者としてのあなたをはっきりとかつ読みやすい形で確認できなければならない。表表紙では『文書』の完全な題名を、題名を構成するすべての語が等しく目立つようにして、視認可能な形で示さなければならない。それらの情報に加えて、表紙に他の文章や図などを加えることは許可される。表紙のみを変更した複製物は、それが『文書』の題名を保存し上記の条件を満たす限り、ほかの点では逐語的に忠実な複製物として扱われる。

もしどちらかの表紙に要求されるカバーテキストの量が多すぎて読みやすく収めることが不可能ならば、あなたはテキスト先頭の一文(あるいは適切に収まるだけ)を実際の表紙に載せ、続きは隣接したページに載せるべきである。

あなたが『文書』の「非透過的」複製物を100部を超えて出版あるいは頒布する場合、それぞれの非透過な複製物と一緒に機械で読み取り可能な透過的複製物を添付するか、それぞれの非透過な複製物(あるいはそれに付属する文書)中で、公にアクセス可能なコンピュータネットワーク上の所在地を記述しなければならない。その場所には、非透過な複製物と内容的に寸分違わず、余計なものが追加されていない完全な『文書』の透過的複製物が置かれ、またそこから、ネットワークを利用する一般公衆が、一般に標準的と考えられるネットワークプロトコルを使ってダウンロードすることができなければならない。もしあなたが後者の選択肢を選ぶならば、その版の非透過な複製物を公衆に(直接、あるいはあなたの代理人ないし小売業者が)最後に頒布してから最低1年間は、その透過的複製物が指定の場所でアクセス可能であり続けることを保証するよう、非透過な複製物の大量頒布を始める際に十分に慎重な手順を踏まなければならない。

これは要望であり必要条件ではないが、『文書』の著者に、『文書』の更新された版をあなたに提供する機会を与えるため、透過非透過を問わず大量の複製物を再頒布し始める前には彼らにきちんと連絡しておいてほしい。

4. 改変

『文書』の改変版を、この利用許諾契約書と細部まで同一の契約の下で発表する限り、すなわち原本の役割を改変版で置き換えた形での頒布と改変を、その複製物を所有するすべての人々に許可する限り、あなたは改変版を上記第2項および第3項が指定する条件の下で複製および頒布することができる。さらに、あなたは改変版において以下のことを行わなければならない。

- A. 題扉に(もしあればその他の表紙にも)、『文書』および『文書』のそれ以前の版と見分けがつく題名を載せること(もし以前の 版があれば、『文書』の「履歴 (History)」の部分に列記されているはずで ある)。もし元の版の出版者から許可を得たならば、以前の版と同じ題名を使っても良い。
- B. 題扉に、改変版における改変を行った1人以上の人物 が団体名を列記すること。あわせて元の『文書』の著者として、最低5人(もし5人以下ならばすべて)の主要著者を列記すること。ただし元の著者たちが この条件を免除した場合は除く。
- C. 題扉に、改変版の出版者名を出版者として記載すること。
- D. 『文書』にあるすべての著作権表示を残すこと。
- E. 他の著作権表示の近くに、あなたの改変に対する適 当な著作権表示を追加すること。
- F. 著作権表示のすぐ後に、改変版をこの契約書の条件 の下で利用することを公衆に対して許可する告知を含めること。その形式は この契約書の末尾にある付記で示されている。
- G. 元の『文書』の利用許諾告知に書かれた、変更不可 部分の完全な一覧と、要求されるカパーテキストとを、改変版の利用許諾告知 中でもそのまま残すこと。
- H. この契約書の、変更されていない複製物を含めること。
- I. 「履歴 (History)」と題された部分とその題名を保存し、そこに改変版の、少なくとも題名、出版年、新しく変更した部分の著 者名、出版者名を、題扉に掲載するのと同じように記載した一項を加えること。もし『文書』中に「履歴」と題された部分が存在しない場合には、『文 書』の題名、出版年、著者、出版者を題扉に掲載するのと同じように記載した部分を用意し、上記で述べたような、改変版を説明する一項を加えること。
- J. 『文書』中に、『文書』の透過的複製物への公共的 アクセスのために指定されたネットワークの所在地が記載されていたならば、それを保存すること。同様に、その『文書』の元になった以前の版で指定 されていたネットワークの所在地も載っていたならば、それも保存すること。これらの情報は「履歴(History)」の部分に置いてもいい。ただし、それが『文書』自身より少なくとも4年前に出版された著作物の情報であったり、あるいは改変版が参考になっている版の元々の出版者から許可を得たならば、その情報を削除してもかまわない。
- K. 「謝辞 (Acknowledgement)」あるいは「献辞 (Dedication)」等と題されたいかなる部分も、その部分の題名を保存し、そ の部分の内容(各貢献者への謝意あるいは献呈の意)と語調を保存すること。
- L. 『文書』の変更不可部分を、その本文および題名を 変更せずに保存すること。章番号やそれに相当するものは部分の題名の一 部とは見做さない。
- M. 「推薦の辞 (Endorsement)」というような章名が題 された部分はすべて削除すること。そのような部分を改変版に含めてはならない。
- N. すでに存在する部分を「推薦の辞 (Endorsement)」と題されるように改名したり、題名の点で変更不可部分のどれかと衝突する ように改名してはならない。
- O. 保証否認警告を保存すること。

もし改変版に、補遺部分としての条件を満たし、かつ『文書』から複製物された文章や図などをいっさい含んでいない、前書き的な章あるいは付録が新しく含まれるならば、あなたは希望によりそれらの部分の一部あるいはすべてを変更不可と宣言することができる。変更不可を宣言するためには、それらの部分の題名を改変版の利用許諾告知中の変更不可部分一覧に追加すれば良い。これらの題名は他の章名とは全く別のものでなければならない。

含まれる内容が、さまざまな集団によるあなたの改変版に対する推薦の辞のみである限り、あなたは、「推薦の辞 (Endorsement)」と題された章を追加することができる。推薦の辞の例としては、ピアレビューの陳述、あるいは文書がある標準の権威ある定義としてその団体に承認されたという声明などがある。

あなたは、5語までの一文を表カパーテキストとして、25語までの文を表表紙テキストとして、改変版のカバーテキスト一覧の末尾に加えることができる。一個人ないし一団体が直接(あるいは団体内で結ばれた協定によって)加えることができるのは、表カパーテキストおよび裏カパーテキストとしてそれぞれ一文ずつのみである。もし以前すでにその文書において、表裏いずれかの表紙にあなたの(またはあなたが代表する同じ団体内で為された協定に基づく)カパーテキストが含まれていたならば、あなたが新たに追加することはできない。しかしあなたは、その古い文を加えた以前の出版者から明示的な許可を得たならば、古い文を置き換えることができる。

『文書』の著者あるいは出版者は、この利用許諾契約書によって、彼らの名前を利用することを許可しているわけではない。彼らの名前を改変版の宣伝に使ったり、改変版への明示的あるいは黙示的な保証のために使うことを許可するものではない。

5. 文書の結合

あなたは、上記第4項において改変版に関して定義された条件の下で、この利用許諾契約書の下で発表された複数の文書の一つにまとめることができる。その際、原本となる文書にある変更不可部分を全て、改変せずに結合後の著作物中に含め、それらをあなたが統合した著作物の変更不可部分としてその利用許諾告知において列記し、かつ原本にある全ての保証否認警告を保存しなければならない。

結合後の著作物についてはこの契約書の複製物一つ含んでいれよく、同一内容の変更不可部分が複数ある場合には一つで代用してよい。もし同じ題名だが内容の異なる変更不可部分が複数あるならば、そのような部分のそれぞれの題名の最後に、(もし分かっているならば)その部分の原著者あるいは出版者の名前で、あるいは他と重ならないような番号を括弧で括って記載することで、それぞれ見分けが付くようにしなければならない。結合後の著作物の利用許諾告知における変更不可部分の一覧においても、章の題名に同様の調整をすること。

結合後の著作物においては、あなたはそれぞれの原本の「履歴 (History)」と題されたあらゆる部分をまとめて、「履歴 (History)」と題された一章にしなければならない。同様に、「謝辞 (Acknowledgements)」あるいは「献辞 (Dedications)」と題されたあらゆる部分もまとめなければならない。あなたは「推薦の辞 (Endorsements)」と題されたあらゆる部分も削除しなければならない。

6. 文書の収集

あなたは、この利用許諾契約書の下で発表された複数の文書で構成される収集著作物を作ることができる。その場合、それぞれの文書が逐語的に忠実に複製されることを保障するために他のすべての点でこの契約書の定める条件に従う限り、さまざまな文書中のこの契約書の個々の複製物を、収集著作物中に複製物一つ含めることで代用することができる。

あなたは、このような収集著作物から文書一つ取り出し、それをこの契約書の下で頒布することができる。ただしその際には、この契約書の複製物を抽出された文書に挿入し、またその他のすべての点でこの文書の逐語的に忠実な複製に関してこの契約書が定める条件に従わなければならない。

7. 独立した著作物の集積

『文書』あるいはその派生物を、他の別の独立した文書あるいは著作物と一緒にし、一巻の記憶装置あるいは頒布媒体に収めた編集著作物は、編集に起因する著作権が編集著作物に含まれる個々の著作物がその利用者に許可した法的権利を制限するよう行使されない限り、「集積」著作物と呼ばれる。『文書』が集積著作物に含まれる場合、この契約書は、『文書』と共にまとめられた他の独立した著作物には、それら自身が『文書』の派生物で無い限り適用されることにはならない。

このような『文書』の複製物において、この利用許諾契約書の第3項によりカバーテキストの掲載が要求されている場合、『文書』の量が集積著作物全体の2分の1以下であれば、『文書』のカバーテキストは集積著作物中で『文書』そのものの周りを囲む中表紙、あるいは『文書』が電子的形式である場合には表紙の電子的等価物にのみ配置するだけでよい。その場合以外は、カバーテキストは集積著作物全体を取り巻く印刷された表紙に掲載されなければならない。

8. 翻訳

翻訳は改変の一種と見做すので、あなたは『文書』の翻訳をこの利用許諾契約書の第4項の定める条件の下で頒布することができる。変更不可部分を翻訳によって置き換えるには著作権者の特別許可を必要とするが、元の変更不可部分に追加する形で変更不可部分の全てないし一部の翻訳を含めることはかまわない。この契約書や『文書』中の利用許諾告知、保証否認警告すべての英語原本も含める限り、あなたはこの契約書、告知、警告の翻訳を含めることができる。契約書や告知、警告に関して翻訳と英語原本との間に食い違いが生じた場合、英語原本が優先される。

典型的な例として、『文書』のある部分が原文で「Acknowledgements」、「Dedications」、あるいは「History」と題されていた場合、実際の題名を変更するには、題名を保存する(この契約書の第1項)ための条件(同第4項)を満たすことが必要となる。

9. 契約の終了

この利用許諾契約書の下で明確に提示されている場合を除き、あなたは『文書』を複製、改変、サブライセンス、あるいは頒布してはならない。このライセンスで指定されている以外の、『文書』の複製、改変、サブライセンス、頒布に関するすべての企ては無効であり、この契約書によって保証されるあなたの権利を自動的に終結させることとなる。しかし、この契約書の下であなたから複製物ないし諸権利を得た個人や団体に関しては、そういった人々がこの契約書に完全に従ったままである限り、彼らに与えられた許諾は終結しない。

10. 将来における本利用許諾契約書の改訂

フリーソフトウェア財団は、時によってGNU フリー文書利用許諾契約書の新しい改訂版を出版することができる。そのような新版は現在の版と理念においては似たものになるであろうが、新たに生じた問題や懸念を解決するため細部においては違ったものになるだろう。詳しくは <http://www.gnu.org/copyleft/> を参照せよ。

GNU フリー文書利用許諾契約書のそれぞれの版には、新旧の区別が付くようなバージョン番号が振られている。もし『文書』において、この契約書のある特定の版が「それ以降のどの版でも」適用して良いと指定されている場合、あなたはフリーソフトウェア財団から発行された(草稿として発表されたものを除く)指定の版かそれ以降の版のうちどれか一つを選び、その条項や条件に従うことができる。もし『文書』がこの契約書のバージョン番号を指定していない場合には、あなたはフリーソフトウェア財団から今までに出版された(草稿として発表されたものを除く)版のうちからどれか一つを選ぶことができる。

付録: この利用許諾契約書をあなたの文書に適用するには

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

(訳:

Copyright (C) 西暦年 あなたの名前。
この文書を、フリーソフトウェア財団発行の GNU フリー文書利用許諾契約書(バージョン1.2かそれ以降から一つを選択)が定める条件の下で複製、頒布、あるいは改変することを許可する。変更不可部分、表カバーテキスト、裏カバーテキストは存在しない。この利用許諾契約書の複製物は「GNU フリー文書利用許諾契約書」という章に含まれている。
)

もし変更不可部分や表カバーテキスト、裏カバーテキストがあれば、「変更不可部分…は存在しない。」というところを以下で置き換えてください:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

(訳:

(章の題名を列記)は変更不可部分であり、(表カバーテキストを列記)は表カバーテキスト、(裏カバーテキストを列記)は裏カバーテキストである。

)

変更不可部分はあるがカバーテキストは存在しないなど、その他の三者の組み合わせに関しては、状況に合わせて上記二つの選択肢を混ぜてください。

あなたの文書に、他に類を見ない独自のプログラムコードのサンプルが含まれる場合、フリーソフトウェアにおいてそのコードを利用することを許可するために、そういったサンプルに関してはこの利用許諾契約書と同時にGNU 一般公衆許諾契約書のようなフリーソフトウェア向けライセンスのうちどれか一つを選択して適用してもよい、というような条件の下で発表することを推奨します。

