

【論文】

GNU/Linux のネットワークブートを応用したリカバリーシステム

山田丈博[†] 本田 貢[†] 松本勝哉^{††}

GNU/Linux Network Boot System Applied to Recovery

Takehiro YAMADA[†] Mitsugu HONDA[†] Katsuya MATSUMOTO^{††}

Abstract : The most undesirable situation for long time operation of the personal computer (PC) is that the operating system of PC is not available to boot suddenly because of the hard disk troubles. It's difficult to access to the system files/data of local client PC on a kind of situation. We proposed the new recovery system for the damaged PC system on laboratory local area network by use of both of "network boot" and "KNOPPIX". This paper presents the summary of the construction of this recovery system and the verification of them.

Keyword: Recovery, KNOPPIX, Network boot

1 はじめに

特定のパーソナルコンピュータ(以下、PC)やネットワークを長期運用する場合、最も想定しやすく、且つ避けたいトラブルは、コンピュータウィルスの影響やシステムファイルの損失により、ハードディスクドライブ(以下、HDD)にアクセスできず、OSが起動不可能な状態になることである。OSが起動できなければ、クライアント機のローカルHDDへのアクセスが困難になり、データ或はシステムの退避復旧が非常に困難になる。

本研究は、研究室内のクライアント機やサーバ機に前述のような不具合が発生した場合を想定し、ネットワークブートとKNOPPIXを応用したリカバリーシステムを構築し、システム復旧を行うための環境を整えることを目的としたものである。以下、構築したシステムの概要と検証結果を報告する。

2 リカバリーシステムの概要

2.1 システム構築にあたって

HDDの故障には大きく分けて「物理的な損傷」と「論理的な損傷」の2種類に分けることが出来る。前者の物理的な損傷とは、その名の通り物理的にHDDが損傷し、データを参照できなくなることを示し、データの復旧は不可能に近い。逆に、論理的な損傷とは、HDDは物理的には壊れていないのだが、誤ったフォーマットやシステムの変更や損傷、

が、誤ったフォーマットやシステムの変更や損傷、ウィルス感染、パーティションテーブルやマスターブートレコードの損傷などによりOSが起動しなくなることを示す。この場合、HDD内のデータは存在するのだが、OS自体が起動できないためデータを読み出すことは困難である。

論理的な損傷は、システムの基幹に関わる部分を扱う場合に起こりやすく、ネットワーク環境においては、サーバ機クライアント機に関わらず発生する可能性があり、対策を講ずる必要がある。幸い、論理的な損傷の場合、復旧対象となるデータはHDD内に残っている。OSの起動が不可能になっている要因さえ取り除ければ再度HDDからのOS起動が可能になる。また、OS復旧が不可能な状態に陥っても、HDD内に残されたデータを救出する手段は幾つかある。

本論文では、そのような点に着目してネットワークブートとKNOPPIXを活用したリカバリー環境の構築の提案をし、動作検証をしている。

2.2 システムの構成

本システムの構成において、まず、GNU/Linuxのディストリビューションの1つであるKNOPPIXをリカバリ作業用途向けにリマスタリングした。また、ネットワークブートサーバを構築し、サーバ内にKNOPPIXのイメージファイルを置くことでリカバリーシステムを構築した。このような構成をとることで、仮にクライアント機に障害が発生したとしても、ネットワークブートにてKNOPPIXを起動させ

[†] 工学研究科電気工学専攻1年

^{††} 電気工学科

ることが可能となる。また、ネットワークブートにて起動したKNOPPIXを介してローカルHDDにアクセスすることも可能となるため、復旧作業を行うための環境を実現する事ができる。復旧作業は、リマスタリングの際にKNOPPIXに導入した各種ツールを用いることで実施できる。図1にシステムの概要を示す。

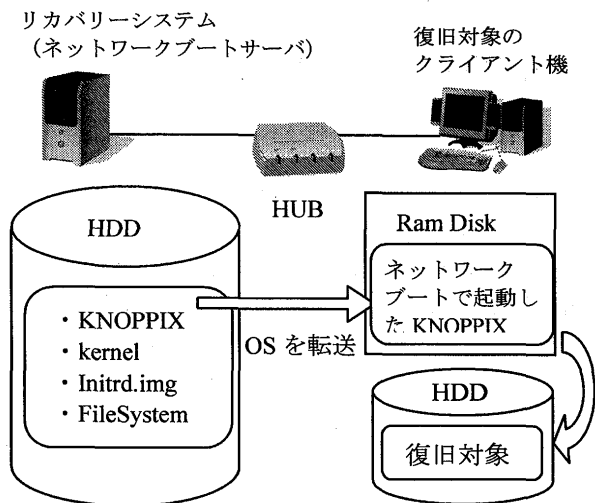


図1 システムの概要

3 KNOPPIXのネットワークブート

本研究においては、起動できなくなったクライアントのローカルOSの代わりに、別マシン上にあるKNOPPIXをネットワークブートにて起動させる手段を用いる。ここではネットワークブートで用いるOSであるKNOPPIX及び、ネットワークブートの手法について述べる。

3.1 KNOPPIX

KNOPPIXとは、Debian GNU/Linuxをベースとして開発されているLinuxディストリビューションの1つである。特徴としては、先ずHDDへのインストールを必要とせず、CD-ROMから直接起動して利用できるいわゆる1CD Linuxという点であろう。

通常、1CD Linuxは、インストールが不要であるといった手軽さが長所である反面、データを格納しているメディアの容量にOS環境や機能を左右される点やデバイスの認識等の問題など、数々の短所も持ち合わせている。しかし、KNOPPIXは、独自の圧縮ループデバイス cloop(compressed loopback device)を使うことで、約2GBの容量を700MBのCD-ROMに収めている。また、利点としてデバイスの knoppix-autconfig というシェルスクリプトを用いて優れた自動認識・自動設定機能を再現しており、システムメンテナンスに関するアプリケーションも多数収録されている。このように、KNOPPIXは1CD Linuxの欠点を見事に補っているOSであるといえる。

今回は、KNOPPIXの持ち得る利点であるデバイ

スの自動認識・自動設定に優れている点と、メンテナンスやリカバリーに長けているOSである点に着目し、KNOPPIXを更にシステムリカバリーやバックアップ、メンテナンスが容易になるようにカスタマイズし、それをネットワークブートさせることで、有事の際に研究室内の環境を保守したり修繕したりすることを目的とした非常時における有用なリカバリーネットワークシステムである。

3.2 KNOPPIXのリマスタリング

KNOPPIXや、その他の代表的なLinuxディストリビューションはインストール直後からOSを汎用的に利用できるような設定となっており、普通にOSを利用する場合においては特に問題はない。しかし、専門性が高い用途や特定の目的でOSを利用する場合、その目的に適ったソフトウェアが収録されていない場合などがある。そのような場合、目的に応じてソフトウェアのインストールやkernelの再構築を行う必要がある。このことを、リマスタリングと呼ぶ。

KNOPPIXはシステム復旧等に長けたOSである。しかし、初期から収録されているソフトウェアだけでは、必ずしもすべての場合に対応できるとは限らない。例えばバックアップツールを考えてみると、単にバックアップツールといっても、ファイルシステム全体をバックアップするソフトウェアもあれば、データベースやネットワーク上のファイルサーバ内のデータといった特定のものをバックアップするものなど、対象によって使用するツールが異なってくる場合がある。このような場合を想定して、複数の種類のソフトウェアを事前にインストールする必要がある。複数のリカバリーツールやバックアップツールなどをインストールし、KNOPPIXのカスタマイズを行うことで、更にリカバリーやメンテナンスに特化したOSにすることが可能であるといえる。

通常のHDDにインストールされたLinuxであれば、リマスタリングは容易である。しかし、KNOPPIXのファイルシステムはCD-ROMに保存されているため、データの追記はできず、通常の方法ではリマスタリングが不可能である。KNOPPIXのカスタマイズを行うには、以下のような手順を踏まなければならない。

- (1) CD-ROM内にある cloop ファイルの中身をHDDの空き領域にコピーする
- (2) chroot コマンドでコピー先のディレクトリを仮のルートファイルシステムとして扱う。
- (3) アプリケーションの削除や追加の作業を行う
- (4) サーバやデスクトップなどに関する各種設定ファイルの編集を行う
- (5) chroot を抜けて、編集したファイルシステムを cloop ファイルに圧縮する
- (6) CDイメージファイルを作成する。

このように、一旦 cloop ファイルの中身を適当な領域に展開した後、データの追加と削除を行い、その

後 `cloop` ファイルに再圧縮する手順を踏む必要がある。今回のシステム構築に際しては、前述と同様の手順で `KNOPPIX` のリマスタリングを行った。

3.3 ネットワークブート

ネットワークブートとは、ローカルストレージ (HDD など) から OS を起動させるのではなく、ネットワーク経由でファイルサーバからカーネルをダウンロードし、さらにネットワーク上に存在するファイルシステムをマウントすることで OS を起動させる方法である。ネットワークブートの手法は幾つかあるが、今回は `PXE` (Preboot eXecution Environment) という手法でリマスタリングした `KNOPPIX` を起動させる手段を選んだ。`PXE` とは、Intel の作成したネットワークブートの為の規格であり、サーバおよびクライアントが従うべきプロトコルなどが規定されている。`PXE` 準拠の NIC は、これに準拠したソフトウェアを ROM 上にもっている。つまり、`PXE` でのネットワークブートの場合は、ブート用のフロッピーディスクや CD-ROM など別途用意する必要がなく、`PXE` に対応した NIC さえあればネットワークブートを再現できるのである。

3.3.1 通常のブートプロセス

通常、LinuxOS のブートプロセス概略は、以下の図 2 に示す通りの流れとなる。

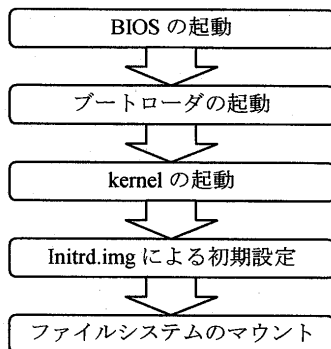


図 2 通常の LinuxOS のブートプロセス

先ず、PC の電源が ON になると BIOS が読み込まれ、ハードのセルフテストや起動ドライブのサーチ等が行われた後、マスターブートレコード (MBR: Master Boot Record 以降、MBR と記す) が読み込まれる。この MBR とは、起動ドライブの先頭セクタのことであり、この領域に OS の起動に必要なプログラムであるブートローダが記述されている。MBR が読み込まれると、MBR 内にあるブートローダが読み込まれる。読み込まれたブートローダは、予め決められた設定に従って、指定された kernel を起動させる。それと同時に、`initrd.img` というファイルを読み込む。そして、kernel はそれに含まれるスクリプトを用いてファイルシステムの環境を整えた後にローカルストレージ内にあるメインの

ファイルシステムをマウントし、GUI を再現するプロセスなどを生成していく。通常のブートプロセスの場合は、ブートローダや OS の核ともいえる kernel はローカルストレージからメモリ上に呼び出されて実行されている。それに対し、ネットワークブートは、他の PC のストレージからネットワークを介してローカルメモリにこれらのものを呼び出す。要は、必要なものを参照する先がローカルかネットワーク越しのマシンかの違いだけであり、OS 起動のプロセスは殆ど変わらない。ネットワークブートの仕組みを考えることは、OS の起動プロセスを考えるというよりも、必要なものをどのような手段でネットワーク越しから取得するかを考える側面が強いのである。

3.3.2 ネットワークブートプロセス

前述のように、ネットワークブートの場合は、どのような手段で必要なものを取得するかを考えなければならない。`PXE` でのネットワークブートは図 3 に示すプロセスでブートローダ等を取得している。

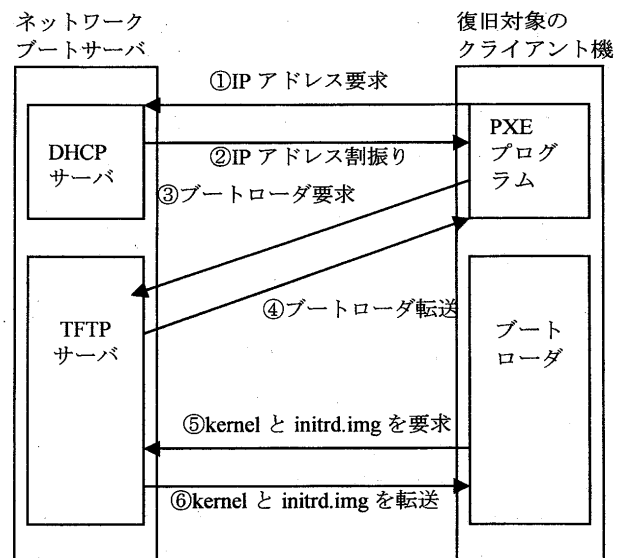


図 3 ネットワークブートの流れ

PC の電源が入り、BIOS が起動する。その後、NIC 上の ROM に書き込まれている `PXE` プログラムが起動する。`PXE` プログラムが DHCP 要求を発行した後、`PXE` ネットワークブートサーバとして動作しているリカバリーシステムの DHCP サーバが IP アドレス IP アドレスやブートローダのパスを含んだ応答を返す。IP アドレスの設定が終了した後、`PXE` プログラムはサーバ側に TFTP でブートローダファイル取得を要求し、システム上の TFTP サーバが要求されたブートローダファイルをクライアントに転送する。ここまでの `PXE` プログラムの役割である。その後、クライアントが取得したブートローダを起動する。以降ブートローダが制御を行う。最初に、ブートローダは、システム上の TFTP

サーバにブートローダの設定ファイルを要求する。そしてシステム側の TFTP サーバが要求のあった設定ファイルをクライアントへ転送する。クライアントはブートローダ用の設定ファイルから kernel のパスを取得し、kernel を TFTP 経由で取得する。こうしてクライアントのローカルメモリに kernel を呼び出して起動する。このような流れを得て、ネットワーク越しで kernel 呼び出され、OS が起動する。

4 リカバリーシステムの構築

リカバリーシステムとして構築したサーバ環境の詳細や、各種サーバの設定方針や内容について示す。

4.1 KNOPPIX リマスタリングの実践

序論で述べたように、本研究の目的は「ネットワークブートを応用して非常時システムを構築すること」である。リマスタリングを行うにあたって、前述の目的に適した OS に仕上がるよう留意する必要がある。本研究において、リマスタリングを行う際に、目的に適した OS にするべく幾つか方針を立てた。その方針を以下に示す。

- OpenOffice や Gimp など、明らかにリカバリーやバックアップに不要なソフトウェアは削除する。
- 不要なソフトウェアの削除が完了した後、リカバリーやバックアップ、メンテナンスに関するソフトウェアを追加する。
- 前述の変更に合わせてデスクトップも変更する。
- 各種サーバの設定を済ませておき、非常時におけるサービスが即座に開始できるようにしておく。

上記の方針を踏まえ、KNOPPIX のリマスタリング手順に沿って作業を行い、リカバリー目的の KNOPPIX を作成した。

4.2 PXE サーバの構築と設定

クライアント機をネットワークブートさせるには、DHCP、NFS、TFTP サーバを組み合わせ、PXE サーバを構築し、設定を行う必要がある。ここでは、各種サーバの設定について述べる。

4.2.1 DHCP サーバの設定

クライアント機がネットワークブート可能な状態にするために、サーバ機側の DHCP が成すべき役割は大きく分けて 2 つある。まず 1 つは、クライアント機の電源が ON になった際にプライベート IP アドレスを割り振ることである。次に、クライアント機が PXE でネットワークブートする際に利用するブートローダファイルの指定である。ネットワークブートを行う際は、最低限この 2 点の設定は行わなければならない。

他にも、割り当てる IP アドレス範囲の指定や IP アドレス貸し出し時間などの指定等、サブネットを構築するにあたり必要不可欠な設定が幾つかある。

それらの事を鑑み、DHCP の設定を次のように行った。

- クライアント機のネットワークブートを可能にするために、BOOTP プロトコルを用いる設定とした。
- IP アドレス貸し出し時間の設定を行った。
- サブネット番号とサブネットマスク、及びゲートウェイサーバの設定を行った。
- クライアントに割り振る IP アドレス範囲の設定を行った。
- クライアントが参照するブートローダの場所を指定した。
- ゲートウェイサーバを介して DNS サーバに対する問い合わせを可能にした。

上記の設定を DHCP サーバの設定ファイルである /etc/dhcp3/dhcpd.conf ファイルを編集することで反映させた。

4.2.2 TFTP サーバの設定

TFTP サーバは、クライアント機の電源が ON になり、NIC の ROM から PXE クライアントプログラムが起動し、DHCP によってクライアント側に IP アドレスが割り振られた後に動作する。TFTP サーバが果たすべき役割としては、まず、サーバ機側からクライアント機側にネットワークブート用の PXE ブートローダを転送することが挙げられる。また、予め指定しておいたブートローダの設定に基づいて、カーネルと initrd.img ファイルをクライアント側に転送することである。果たすべき役割が単純であるが故に、設定も容易である。

行うべき設定としては、クライアント側に転送するブートローダと kernel、initrd.img ファイルを格納するディレクトリのパスを指定することである。また、TFTP はユーザ認証を必要としないファイル転送プロトコルであるから、セキュリティの関係上、OS の初期設定の段階では使用は許可されていない。従って、TFTP プロトコルの使用を許可する設定も行う必要がある。

以上の点を踏まえた上で TFTP サーバの設定を行い、クライアント側にブートローダ、kernel、及び initrd.img ファイルの転送を可能とした。

4.2.3 NFS サーバの設定

NFS サーバはサーバ内の任意のディレクトリを公開し、その公開領域をクライアント側のルートファイルシステムとして扱うために用いる。つまり、NFS で公開するファイル領域がクライアント側のルートファイルシステムとなるように設定する必要がある。

今回は、CD イメージファイルを NFS で公開し、そのイメージファイル内にある cloop ファイルとして圧縮されたファイルシステムをクライアント側のルートファイルシステムとして扱う。よって、CD イメージファイルのマウントポイントを NFS で公

開する必要がある。今回は、CD イメージファイルのマウントポイントとして/mnt/pxe/というディレクトリを用意した。サーバ機内の/mnt/pxe を NFS で公開するように設定すればよい。具体的には、以下の項目に留意しつつ設定を行う。

- ・サーバ機の IP アドレスの指定
- ・NFS 公開領域である/mnt/pxe のパス指定
- ・NFS に合わせた host.allow ファイルの編集

上記の設定を行うことで、サーバ機内にある CD イメージファイルに含まれる cloop ファイルをクライアント側のルートファイルシステムとして扱うことが可能となった。

4.2.4 PXE ブートローダの設定

PXE ブートローダは、前述したように TFTP によってクライアント側に転送される。転送された後は、あらかじめ指定しておいた設定に基づいてカーネルや initrd を呼び出し、動作させる。ここでは、PXE ブートローダの設定について述べる。

今回、PXE ブートローダとして使用したのは syslinux というパッケージに含まれる pxelinux.0 というファイルである。この pxelinux.0 をクライアント側に転送するためには、最低限の準備として、pxelinux.0 とその設定ファイルを、TFTP で指定しているディレクトリに置く必要がある。また、設定ファイルには、呼び出す kernel と initrd.img ファイルの指定をしておく必要がある。具体的な設定項目を以下に列挙する。

- ・デフォルトで呼び出す kernel の指定、あるいはラベルの指定
- ・ブートプロンプトの表示の有無
- ・ブートプロンプトの表示時間
- ・ラベル名と、ブートプロンプトにラベル名が入力された際に呼び出される kernel と initrd.img の指定およびブートオプションの指定。

PXE ブートローダの設定を行う際は、以上の事柄に関しては適切に設定しなければならない。今回は、ブートプロンプトの表示を許可し、その表示時間を 30 秒とした。また、デフォルトで実行されるラベルを KNOPPIX とし、kernel は vmlinuz、initrd.img ファイルは miniroot.gz が呼び出されるように設定した。

4.2.5 kernel 及び initrd ファイルの準備

各種サーバソフトの設定とブートローダの設定を終えたら、ネットワークブートの際にクライアント側に渡す kernel と initrd.img ファイルを準備する必要がある。その際、ネットワークブートに使用する kernel は NFS をサポートするように構築されていなければならない。また、initrd.img ファイルに含まれる linuxrc スクリプトを NFS 向けに編集している必要がある。

今回は、KNOPPIX に導入されている knoppix-terminalserver で利用されているカーネルと initrd を

流用した。knoppix-terminalserver とは、CD ブートした KNOPPIX を、ネットワークを介しても起動できる仕組みである。本研究にて構築したシステムと基本的な部分は似ているが、大きな違いとしては、knoppix-terminalserver は、クライアントのファイルシステムとして CD-ROM 内にある cloop ファイルを参照する。従って、CD ブートにて起動した KNOPPIX でしか正常に動作しない。また、管理者がクライアント側のファイルシステムに変更ができず、クライアント側の起動速度が CD ドライブの速度と回線速度の両方に依存するという欠点も持っている。本研究で構築したシステムは、その欠点を補いシステム変更に関して柔軟性を持たせたものとも言える。

さて、用意したカーネルと initrd.img ファイルは、前もって設定した TFTP で指定した領域に置いておく。そうすることで、PXE ブートローダがクライアント機に転送された後、設定に基づいてカーネルと initrd.img が起動し、OS の起動プロセスが走る。

以上の各種サーバ設定を終えたら、DHCP、TFTP、PORTMAP、NFS の順番で各サーバソフトを起動させれば、PXE サーバとして動作させることが可能となる。

4.3 ファイルシステム

リカバリーシステムは、OS としてカスタマイズ済の KNOPPIX を HDD にインストールして、それをルートファイルシステムとして用いている。また、DHCP、TFTP、NFS 等の各種サーバソフトを組み合わせ、PXE サーバとして動作するように構築している。PXE サーバとして構築することで、クライアント側は、サーバ機が NFS で公開している領域をルートファイルシステムとしてマウントすることが可能となる。即ち、ネットワークブートが可能となる。

さて、ここで、「NFS で公開するクライアント用ファイルシステムをどうするか」といった問題点が浮かび上がってくる。通常、NFS の公開領域をクライアント側のルートファイルシステムとして扱う場合、サーバ管理者は、ルートファイルシステムとは別に、クライアント機の台数分ファイルシステムを別途用意して NFS で公開するといった手法をとる。或いは、ルートファイルシステムとは別のファイルシステムをひとつだけ用意して、その領域を NFS で公開して、複数のクライアントが NFS 公開領域を共有して使うといった手法が取られる。いずれの場合も、サーバ機のファイルシステムとクライアント用のファイルシステムが同じ HDD 内に存在するにも関わらず、設定的に乖離した状態となり、システム的な管理が難しくなる。その点を踏まえると、通常のネットワークブートによる NFS ファイルシステムマウントよりも容易な管理が可能になるようなシステムであることが望ましいといえる。そ

ここで、元々CDブートのOSであるKNOPPIXを用いているという利点を活かしつつ、且つ管理が簡単になるような仕組みを構築する方針でクライアント側のファイルシステムを構築することにした。

次に、クライアント側のファイルシステムの構築について述べる。まず、予め用意しておいたKNOPPIXのCD化に必要なデータをサーバの作業領域にコピーする。また、サーバのファイルシステムで、何かしらのシステム的な変更が起こる可能性のあるディレクトリも作業領域にコピーし、それらのファイルを `cloop` ファイルとして圧縮した後、CDイメージとしてISOイメージファイルとして生成する。そのファイルを任意のディレクトリにマウントし、そのマウントポイントをNFSで公開してクライアント用のファイルシステムとする。つまり、サーバの設定を反映させたKNOPPIXをクライアント側でネットワークブートさせるということである。サーバのファイルシステムをCD化するという作業を自動的に且つ定期的に行うことで、サーバでの仕様変更もクライアント側に反映される。また、定期的に生成されたCDイメージファイルから実際にCD版KNOPPIXを生成する。そうすることで、仮にサーバ自体に何かしらの不具合が生じて動作しなくなったとしても、サーバのファイルシステムをCD化したKNOPPIXを再度HDDにインストールすることで元の環境をある程度の段階までは比較的簡単に再現できる。これらのサーバの動作を図で示すと図4のようになる。

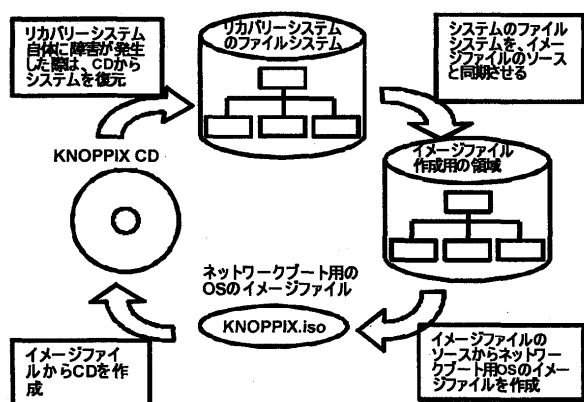


図4 ファイルシステムの生成と管理

上記の図4が示すように、サーバのルートファイルシステムを基にしてKNOPPIXのCDイメージファイルを生成し、それをクライアント側のファイルシステムとして扱う。それと同時に、生成されたCDイメージファイルから実際にCD-ROMのKNOPPIXを生成することで、サーバのルートファイルシステムの内容が反映されたKNOPPIXができることになる。こうすることで、仮にサーバ機に不具合が生じて再構築を余儀なくされたとしても、生成したCD-ROMのKNOPPIXを再度HDDにインストールすることで容易に復旧が可能となる。このよ

うに、ファイルシステムの利用や保持がサーバのHDD内のルートファイルシステムからCDイメージファイルへ、CDイメージファイルからCD-ROMへ、CD-ROMから再度HDDといった具合に循環するような仕組みを目指した。

4.3.1 ファイルシステムのCDイメージ化

サーバのファイルシステムを基にCDイメージファイルを作成するにあたり、サーバのファイルシステムを適当な作業領域に転送して、その転送データを基にCDイメージファイルを作成するのだが、単にサーバのファイルシステムをそのままコピーしたものを `cloop` 圧縮してイメージを作成した場合は、正常に動作しない。サーバのルートファイルシステムをそのままコピーした場合、`/proc` 以下に含まれるサーバのハードウェア情報を記したファイルや、サーバ機自体のファイルシステム構成を記した `/etc/fstab` などといったハードウェア固有の情報を記した設定ファイルも大量にコピーされてしまう。実際に作成したCDイメージファイルは、NFSで公開して別のPCからネットワークブートでのマウントや、CD化して他のPCでの起動を実現したいので、サーバ機固有の設定情報が含まれたイメージを用いると、クライアント機のハードウェア環境や設定内容と、CDイメージファイル内に残った固有設定ファイルの情報との整合性が取れなくなり、不具合につながる可能性が高くなる。すなわち、サーバ固有の設定を引き継がない方法でルートファイルシステムをCDイメージ化する必要がある。そのためには、CDイメージ化する際、サーバ固有の設定ファイルをCDブートでも動作するような設定のファイルに置き換えるか、転送対象から除外しなければならない。ただ、膨大とも言えるほど大量に存在するファイルの中から、1つ1つの設定ファイルの削除や転送から除外するという作業を行うのは非効率的である。そこで、以下のような対処法を考えた。

まず、HDDのルートファイルシステムとオリジナルKNOPPIXのCD内に収録されているルートファイルシステムを比較し、CDイメージに含んだ場合に不具合が起きる可能性があるファイルやサーバ機固有の設定を記したファイルをリストアップする。それらのファイルとCD化に必要なファイルを別途別の場所に予め保存しておき、CD化するためのテンプレートとして扱う。そして、ルートファイルシステムから作業領域に転送するデータは `/bin`、`/dev`、`/etc`、`/usr` などアプリケーションのインストールがあった際に変更される可能性のあるディレクトリのみとし、それらのデータを、テンプレートを基に編集をした後 `cloop` ファイルに圧縮し、最終的にCDイメージを作成する方法を採用した。作業の流れを図として示すと、図5のようになる。ここで示したプロセスを踏むことでルートファイルシステムのCDイメージ化が可能となった。

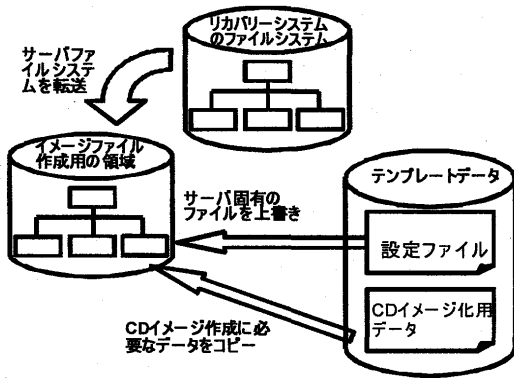


図5 CDイメージ化の流れ

4.3.2 CDイメージ作成の自動化

前述の手順で生成されたCDイメージファイルはクライアント側のファイルシステムとして利用する。よって、定期的にCDイメージを作成することで、サーバのファイルシステム内容をクライアント側のファイルシステムに反映させる必要がある。ただ、クライアント側のファイルシステム更新のために、毎回前述の手順を踏んでCDイメージを再生成するのは膨大な時間と手間が掛かり非効率的である。そこで、実際の処理内容を `makeknopixcd` という名前のシェルスクリプトとして組み上げた。これにより、`makeknopixcd` というコマンド1つでCDイメージ作成が可能になり、定期的なCDイメージの再生成も容易になった。

5 運用・検証

本章では、実際に構築したリカバリーネットワークシステムを実際に運用し、システムとしての利便性の有無や、システムの問題点と可能性を検証する。

5.1 システムの起動と動作の検証

まず、実際に構築したシステムを起動し、その起動時間と動作の検証を行った。検討にあたり、サーバ機からネットワークブートにてKNOPPIXを起動する場合の他に、同じクライアント機を利用して以下の状況との比較を行った。

- クライアント機のHDDにKNOPPIXをインストールし、HDDブートおよびネットワークブートでの起動と動作の比較
- クライアント機にCD-ROM版のKNOPPIXを挿入し、CDブートおよびネットワークブートでの起動と動作の比較

リカバリーネットワークシステムの動作を、HDDブートでのKNOPPIXの動作やCDブートでのKNOPPIXでの動作と比較をすることで、構築したリカバリーネットワークシステムが実用に足るものかを検証する。

なお、検証するにあたって使用したクライアント機の実環境は表1の通りである。

表1 検証に使用したクライアント機の仕様

CPU	Intel(R) Pentium(R) 3 CPU 550MHz		
メモリ	192MB		
HDD容量	/dev/hda1	ブートパーティション	100MB
	/dev/hda2	ルートファイルシステム	10GB
ファイルシステムのフォーマット	ext3		
LANの回線速度	100Mbps		
CDドライブの回転速度	24倍速		

表2 各起動方法におけるKNOPPIXの起動時間

起動方法	起動スピード (s)	
	Xwindowなし	Xwindowあり
HDDブート	90	135
CDブート	105	187
ネットワークブート	125	155

前述のクライアント機にてHDDブート、CDブート、ネットワークブートでの夫々のOS起動時間の測定を行った。実際の測定結果は表2に示す通りである。

測定は、Xwindowssystem即ちデスクトップが立ち上がる前までと、完全に立ち上がった後までを測定している。電源を入れてデスクトップが立ち上がる前までの時間を、表中では「Xwindowなし」と表記し、デスクトップが立ち上がった後までの時間を含めたものを「Xwindowあり」と表記している。

測定結果を見ると「Xwindowなし」の場合、起動スピードはHDD、CD、ネットワークの順となっており、「Xwindowあり」の場合はHDD、ネットワーク、CDの順となっている。いずれの場合もHDDブートには及ばないが、十分に実用レベルの起動速度である。

5.2 リカバリーシステムを利用したデータ復旧の検証

実際にリカバリーネットワークシステムを活用してデータの退避や復旧に関する実験を行った。

5.2.1 データの退避

リカバリーシステムにインストールされているファイル同期ツールの `rsync` と、ファイルサーバの `samba` を使ってデータを復旧対象からリカバリーシステムに退避させる実験を行った。いずれの場合もクライアント側からリカバリーシステム側へのデータ退避は成功した。これにより、クライアント機のローカルOSが起動しない状態であっても、データの退避は可能である事が検証できた。

5.2.2 データの復旧

続いて、データが破損した際の復旧実験を `unrm` で行った。`unrm` とは、TCT(The Coroner's Toolkit)と呼ばれるツールキットに含まれるツールである。

TCT とは、消去されたファイルやファイルの修正された回数などのデータを収集することができるツールキットである。このツールに含まれる `unrm` は、一度削除したデータを復旧させることが可能である。これは、復旧対象データが `ext2` 或いは `ext3` 形式でフォーマットされているファイルシステムに存在する場合以外は利用できないため、色々な環境が混在している研究室においては、一部復旧できない環境も存在した。

以上の結果から、今回構築したリカバリーシステムを利用してのデータ救済作業は、OS が起動しない状況下におけるデータの退避は比較的容易であるが、削除されたデータの復旧は困難であることが検証できた。

5.3 リカバリーシステムを利用したシステム復旧の検証

ここでは PC が起動しないなどのシステム復旧に関して実践したことを述べる。また、それを通じて、今回構築したリカバリーネットワークシステムがシステムリカバリーに有用か否かを検証する。

5.3.1 OS の復旧

OS が起動しない要因として、最も可能性の高いものは、「MBR の損傷」「kernel の損傷や起動に必要なスクリプトの削除」が挙げられる。

MBR が損傷した場合、ブートローダを再インストールすれば復旧できる可能性がある。その場合、復旧対象クライアント機からネットワークブートにてリカバリーシステムを起動し、予めリカバリーシステム内に用意しておいたブートローダの GRUB をローカル HDD にインストールすればブートローダの復旧は可能である。

kernel が何らかの原因で損傷、或いは削除してしまった場合の復旧について述べる。一般的な LinuxOS には、デフォルトの状態では kernel が 1 つだけ含まれる。よって、損傷した場合、システムの kernel が 1 つしかないのでブート不能になる。このような場合、応急処置として、ネットワークブートにて起動した KNOPPIX の kernel を、復旧対象 PC に転用することで、復旧が可能となる。

5.3.2 パーティションの復旧

今まで述べた検証内容は、「復旧対象 PC の HDD パーティションが破損していない場合」が前提である。MBR の復旧に失敗し、パーティションテーブルが消失してしまった場合、OS 起動の際にパーティションに関する情報を参照できないため、パーティションが消失してしまった状態となってしまう。

通常、パーティションテーブルが消失したがために HDD にアクセスできない要因は、そもそもパーティションに関する情報が消えたため、パーティ

ションを参照できないからである。つまり、「データやパーティション自体は HDD 内に存在するが、認識できない」といった状況というわけである。このような場合、要はパーティション情報を突き止めさえすれば、パーティションの復旧作業を試みるのが可能になる。パーティションの復旧には `gpart(guess partition)` というツールで復旧が可能ながある。`gpart` は、ハードディスクをスキャンし、パーティション開始ポイントの目印を探し、得られた情報を基にした推測によってパーティションテーブルを作成するツールである。拡張パーティションの復旧には不得手という弱点もあるが、非常に強力なツールである。ただし、`gpratt` が必ずしも正しい解析結果を返すとは限らないので注意を要する。一度 `gpratt` にて割り出したパーティションテーブルの結果を、復旧する者自身で判断する必要がある。

以上の検証より、OS が起動しなくなる要因として、最も可能性の高い事例に対しては、復旧手段が残されており、リカバリーシステムを介してその復旧を試みるのが可能であることが検証できた。しかし、システム復旧手段は、OS のバージョンやハードウェアの種類、そしてその組み合わせによって生じる環境の差異に大きく影響される。従って、全ての状況に対して同じ手法が使えるとは限らないということも再確認した。

6 まとめ

本論文では、ネットワークブートと KNOPPIX を応用したリカバリーシステムの構築について述べた。また、実際に構築したシステムを利用した検証についても述べた。

KNOPPIX とネットワークブートを応用してシステムを構築したことによって、当初の目的は達した。本システムを構築したことにより、研究室内のクライアント機に障害が発生した際は、容易に復旧環境を呼び出すことが可能となった。今後の課題として、クライアント向けの OS のイメージファイルを生成する過程を見直し、より安全性のあるシステムにする必要がある。これについての改善、改良は目下検討中である。

参考文献

- [1] 山田丈博, 本田貢: 「ネットワークブートと KNOPPIX を応用したリカバリーネットワークシステム」, 九州産業大学工学部平成 17 年度卒業論文 (2006-3)
- [2] 山田丈博, 本田貢, 松本勝哉: 「GNU/Linux のネットワークブートを応用したリカバリーシステム」, 平成 18 年度電気関係学会九州支部連合大会, p532(2006-9)