

# CD-boot Linux の起動時間改善

丹 英之<sup>†</sup> 北川 健司<sup>†</sup> 阿部 大将<sup>†</sup> 千葉 大作<sup>†</sup>  
須崎 有康<sup>‡</sup> 飯島 賢吾<sup>‡</sup> 八木 豊志樹<sup>‡</sup>

**概要:** 起動が遅いと言われ、主に非常用として使われる CD-Boot Linux(以下、ライブ CD)の適用範囲を広げるため、起動時間の改善を検討した。プロファイルによって、遅い原因は CD ドライブのシークに起因することが判明した。そこで、ファイルシステムが格納されている圧縮ループデバイスのブロック配置最適化、及び、ブロックの先読みを実装した。その結果、改善前と比較し半分以上の時間で起動が完了するライブ CD を実現できた。

## 1. はじめに

CD-Boot Linux(以下、ライブ CD)は、起動メディアが書き込み不可能であるという特性から、システム不調時には電源の ON/OFF による復帰手段を採ることができる。このため、家電製品並の高ロバスト性を持つ。また、メディアが非常に安価でプレスによる大量複製が可能である。これら長所を活かし、IT 教育システムのプラットフォームとしてライブ CD の実用性を検証した[1]。その結果、被験者を対象として行った使用感のアンケートでは、システム起動の遅さを指摘する意見を多数得ることになった。ライブ CD を用いる PC は、その起動デバイスの性質上、システムを道具として利用できる状態になるまで時間を要する。実証実験では、このことが顕著に現れた。つまり、ライブ CD をプラットフォームとして利用するには、起動の遅さを解決しなければならないのは明らかであり、また、ライブ CD の適用範囲を広げるためにも、この欠点を早急に解決することが重要であると言える。

そこで、我々はアプリケーションを含むライブ CD システム全体の起動時間を改善する手法を検討した。

## 2. 起動が遅い原因と対策

ライブ CD には幾つか種類があるが、今回は、KNOPPIX 日本語版[2]を元に起動時間改善の検討を行った。KNOPPIX は Debian GNU/Linux を元

としたライブ CD で、周辺機器検出に優れていること、そして圧縮ループデバイス(以下、cloop)上にファイルシステムを格納しており、CD メディアの容量を越えたサイズのデータを収録している。

まず遅い原因を探るため、Bootchart[3]を組み込み、システム起動時の CPU、I/O、プロセス生成状況を確認した。その結果、1)周辺機器検出中には殆んど I/O アクセスが発生しない、2)X サーバ、デスクトップ環境である KDE 起動時には CPU の状態が I/O 待ちである場合が多い、の 2 点が明らかになった。デスクトップ環境起動には、様々な種類のファイルを参照している。これと、CD ドライブのシーク・回転待ち時間の相乗効果により、CPU の I/O 待ちが長くなっていると考えられる。

KNOPPIX で採用している cloop は、仮想ブロックデバイスである。デバイスへのアクセス時、その参照しようとするセクタが含まれる圧縮されたブロックのデータを随時伸長することによって、デバイスからのデータ読み出しを実現する。KNOPPIX の場合、この圧縮されたブロックデータがファイルとして CD メディアの容量の大部分を占有している。つまり、CD のアクセス単位は、圧縮ループデバイスのブロック単位である。そこで、CD ドライブのシーク・回転待ち時間を削減し、CPU の I/O 待ち状態を低減するため、ブロックの配置をシステム起動時の参照順に並べ替えることを考えた。また、周辺機器検出中、検出後に参照するブロックをあらかじめ先読みしておけば、I/O の空き時間を有効に

<sup>†</sup> 株式会社 アルファシステムズ  
Alpha Systems, Inc.

<sup>‡</sup> 独立行政法人 産業技術総合研究所  
National Institute of Advanced Industrial Science and Technology

活用でき CD ドライブからメモリへデータを効率よく転送できることになる。今回の検討では、ブロック配置最適化、ブロック先読み、の二つを実装してみることにした。

### 3. 実装と評価

システム起動時のブロック参照状況を把握するため、`cloop` ドライバ(`cloop-2.01`)にブロック参照状況として、ブロック番号、参照時刻、CD メディア上の `iso9660fs` から読み出しに要した時間を保持する機能を実装した。保持したデータは、システムの起動完了後、`procf`s を経由して取得できる。これにより、起動時に参照するブロックの順番を取得できる。これを元にブロックを再配置する、`cloop` デバイスのイメージファイルコンバータも実装した。CD ドライブは回転制御方式によって、データ転送速度が異なる。角速度一定(CAV)の場合、CD メディアの外周に配置されたデータほど転送速度が増すので、この効果も期待し、起動時に参照したブロックをイメージファイルの後半に配置するようにした。また、配置最適化によりイメージファイルのフォーマット変更を余儀なくされたので、これに対応できるよう `cloop` ドライバを拡張した。また、`cloop` 中の指定されたブロックを読み込むコマンドも用意した。これは、`init` の起動前、`cloop` デバイスがマウントされた直後に実行され、参照リストを元にブロックを読み出し、`kernel` の仮想メモリサブシステムのページキャッシュに格納する。

`kernel` は 2.6.13.3 を採用し、ブロック配置最適化を施すライブ CD には KNOPPIX Edu4.1 (KNOPPIX 3.7 日本語版ベース)[4]を採用した。つまり、`kernel` と `cloop` ドライバが新規で、ユーザランドは KNOPPIX Edu4.1 のまま、ということになる。起動時のブロック参照状況は、VMware Workstation 5.0.0 上で取得し、ブロック配置最適化を施した後、CD-R を作成した。

図1は、実機(CPU:PIII-933MHz, MEM: 256MB, CD:24x)で起動時間に要した時間をプロットしたものである。値はそれぞれ、最適化後 83 秒、KNOPPIX3.7 は 207 秒、Edu4.1 は 311 秒である。Edu4.1 の場合、`tmpfs` 上のホームディレクトリへ計 23MB のファイルを転送するのでベースとした 3.7 よりも時間を要する。しかし、ブロック配置を最適化したものは、そのファイル転送が有るにもかかわらず、ベースとした 3.7 の半分以内の時間で起

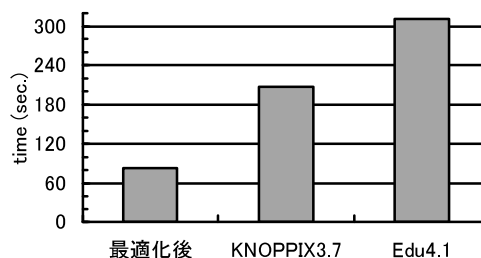


図1 ブロック配置最適化とブロック先読みを実装したライブ CD と各 KNOPPIX の起動時間

動が完了する。これらからライブ CD の起動に要する時間の殆どが CD ドライブのシーク・回転待ち時間であることは明白である。

### 4. まとめ

ライブ CD で用いられる `cloop` デバイスのブロック配置最適化、I/O 空き時間を利用したブロック先読みを実装した。その結果、従来の半分の時間で起動するライブ CD を実現できた。

今後は、周辺機器検出の非同期化、デーモン起動並列化、物理メモリ量に応じたブロックの先読みなど、更なる起動時間の改善を順次検討していきたい。

これら、起動時間を改善したライブ CD は *Accelerated-KNOPPIX*[5]として公開していく。

### 5. 謝辞

本プロジェクトは、独立行政法人情報処理推進機構(IPA)2005 年度「オープンソースソフトウェア活用基盤整備事業」[6]の一部として行われています。ご支援頂き感謝致します。

#### 参考文献

- [1] IPA2004 年度「学校教育現場におけるオープンソースソフトウェア活用に向けての実証実験」Web Page.  
<http://www.ipa.go.jp/software/open/2004/stc/eduseika.html>
- [2] “KNOPPIX 日本語版” Web Page.  
<http://unit.aist.go.jp/itri/knoppix/>
- [3] “Bootchart” Web Page.  
<http://www.bootchart.org>
- [4] “KNOPPIX Edu” Web Page.  
<http://www.alpha.co.jp/knoppix/edu/>
- [5] “Accelerated-KNOPPIX” Web Page  
<http://www.alpha.co.jp/ac-knoppix/>
- [6] IPA2005 年度上半期「オープンソースソフトウェア活用基盤整備事業」Web Page.  
<http://www.ipa.go.jp/software/open/2005/result.html>