

# 日曜研究室

技術的な観点から日常を綴ります

## [xv6 #54] Chapter 5 – File system – Overview

テキストの63ページ

### 本文

xv6のファイルシステムの実装は、図5-1のように、6層で構成される。

一番下の層は、バッファキャッシュを通してIDEディスク上のブロックを読み書きする。

そしてそれは、ディスクブロックへのアクセスを同期化し、一度に一つのカーネルプロセスだけが、どこかの特定のブロックに保管されているファイルシステムのデータを編集出来る、ということを保証する。

2番目の層は、上位の層がトランザクションを使って複数のブロックを更新する事を可能にする。

トランザクションは、その複数のブロックがアトミックに更新される事を保証する。

（そのすべてのブロックが更新されるか、どのブロックも全く更新されないか、必ずどちらかになることを保証する）

3番目の層は、名前がないファイルを提供し、それらはinodeと、ファイルのデータを保持しているブロックの連なりを用いて表現される。

4番目の層は、内容にディレクトリエントリの連なりを持っている特殊なinodeとして、ディレクトリを実装する。

ディレクトリエントリはそれぞれ、名前と、名付けられたファイルのinodeへの参照を含む。

5番目の層は、再帰的な探索を用いて、/usr/rtn/xv6/fs.cのような階層的なパス名を提供する。

最後の層は、ファイルシステムのインターフェイスを使った多くのUnix資源（例えば、パイプやデバイスやファイル等々）を抽象化する。

これは、アプリケーションプログラマの生活をシンプルにする。

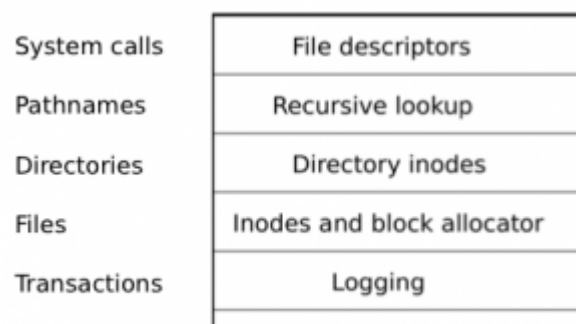




図5-1 xv6のファイルシステムの層

ファイルシステムは、inodeと、内容を格納するためのブロックをディスク上のどこに保管するか、計画を持っていなければならない。

そのためにxv6は、図5-2のように、ディスクをいくつかの区画に分割する。

ファイルシステムは、ブロック0は使わない（そこにはブートセクタがある）。

ブロック1は、スーパーブロック（superblock）と呼ばれ、ファイルシステムについてのメタデータを含む。

（ブロック単位でのファイルシステムのサイズ、データブロックの数、inodeの数、ログ区画のブロック数）

2から始まるブロック群はinodeを保持し、ひとつのブロック毎に複数のinodeを含む。

その後に続くビットマップ区画のブロックは、どのデータブロックが利用中か（ファイルで利用中かなど）を追跡する。

一番大きい区画がデータ区画で、その中の各データブロックは、ファイルやディレクトリの内容を保持する。

ディスクの終端に置かれる区画のブロックは、トランザクションのためのログを保持する。

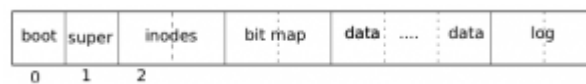


図5-2 xv6のファイルシステムの構造。fs.hヘッダファイルは、ファイルシステムの厳密なレイアウトを表現する定数とデータ構造を含む。

この章では、下の層から順番に、各層について説明していく。

よく練られた下層の抽象化が、上層のデザインを楽にすることに注意せよ。

ファイルシステムは、良くデザインされた抽象化が、どうやって驚くべき汎用性をもたらすかという事の、良い例である。

## 感想

xv6のファイルシステムもやはりシンプルな構造のようです。

それでも、シンプルではあっても現実のファイルシステムであるFAT32などよりは、トランザクション対応の点で高機能ですね。

カテゴリー: 技術 | タグ: xv6 | 投稿日: 2012/3/27 火曜日 [<http://peta.okechan.net/blog/archives/1582>] |

