

日曜研究室

技術的な観点から日常を綴ります

[xv6 #53] Chapter 5 – File system

テキストの63ページ

本文

ファイルシステムの目的は、データを整理し保管することである。

ファイルシステムは一般的に、再起動した後もデータが残るような永続性をもって、ユーザやアプリケーション間におけるデータの共有をサポートする

xv6のファイルシステムは、Unixライクなファイル・ディレクトリ・パス名（第0章を参照）を提供し、そのデータをIDEディスク上で永続的に保管する（第2章を参照）。

xv6のファイルシステムは、次のいくつかの挑戦に取り組んでいる。

- ・ ファイルシステムは、名前付けられたディレクトリとファイルのツリーを表現し、それぞれのファイルの内容を保持するブロックの同一性を記録し、ディスクのどの領域が空いてるかを記録するために、ディスク上のデータ構造を必要とする。

- ・ ファイルシステムは、クラッシュリカバリ機能をサポートしなければならない。
それは、もしクラッシュした場合（例えば急な電源断など）、再起動後も正しく動作しなければならない。

クラッシュが、ディスク上のデータ構造の一連の更新中に発生し、矛盾したデータを残してしまう事がリスクである（例えば、あるブロックがファイルで使われてるのに、空き領域としてマークされてしまっている状態）。

- ・ 複数のプロセスが、同時にファイルシステム上のデータを操作する可能性があり、インバリアントを保つために調整しなければならない。

- ・ ディスクへのアクセスは、メモリへのアクセスに比べて桁違いに遅いので、ファイルシステムはよく使われるブロックのメモリ上のキャッシュを保たなければならない。

この章では、xv6が以上の挑戦にどう取り組んでいるかについて説明する。

感想

やっとファイルシステムの章に入りました。

最後の章です。

実はこの章が一番興味があります。

この章の後も、Appendix A（PCハードウェアについて）とAppendix B（ブートローダについて）という付録がありますが、そこも読むかどうか、ブログに載せるかどうかは未定です。

クラッシュリカバリ対応ということで一瞬マジか！？と思いましたが、明示的にその機能を実装するわけではなくて、単にどのタイミングで処理が止まっても大きな問題にならないようなデータ構造と処理の順番を採用するだけなのかもです。

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/1580>] |
