

iDefrag 4

i・デフラグ

クイックスタートガイド



2010年3月15日（PDF版/第1版）



はじめに

このガイドには、iDefragの概要が書かれていますが、iDefragのすべての機能を網羅しているわけではありません。iDefrag内のヘルプが包括的なリファレンスガイドになっています。



このガイドを読み、ヘルプを参照してもこの製品の特定の使い方がわからない場合は、[サポート情報 \(P.10\)](#)を参照してください。

本ガイドで説明する内容は次の通りです。

- [操作を始める前に \(P.3\)](#)
- [アクセス許可 \(P.4\)](#)
- [メイン表示 \(P.4\)](#)
- [スタートアップボリュームのデフラグ \(P.6\)](#)
- [アルゴリズム \(P.7\)](#)
- [ディスクにデフラグが必要かどうか？の判断 \(P.8\)](#)
- [次のステップ \(P.8\)](#)
- [温度の監視 \(P.9\)](#)
- [情報パネル \(P.9\)](#)
- [サポート情報 \(P.10\)](#)
- [付録-ファイルシステムとディスク \(P.11\)](#)

操作を始める前に



iDefrag で操作を始める前に、重要なデータをバックアップしておく必要があります。

デフラグ操作は本来危険を伴うものであり、デフラグ中にディスクにアクセスし続けると、そのディスクでハードウェア上の問題が発生する可能性があります。ただし、データ消失を発生させる可能性のある iDefrag での不具合をすべて把握するのは困難です。

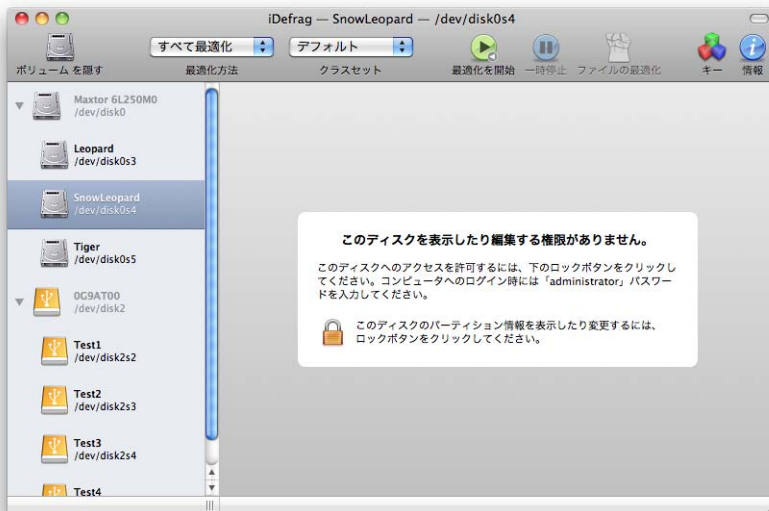
また、ファイルシステムにエラーがないかチェックする必要があります。これを行うには、[最適化] メニューから [エラーチェック] を選択するか、ディスクユーティリティを使用します。どちらの操作も全く同じチェックを実行します。ディスクユーティリティを使用する場合は、[ディスクのアクセス権を検証] ボタンではなく、[ディスクを検証] ボタンを選択してください。



ディスクユーティリティソフトを使用しなくても、データが壊れる可能性は日常的にあります。そのためにも日頃からバックアップを取っておく必要があります。大切な写真、音楽、ホームムービーのデータをマシンに保存している場合には、安全対策をとる必要があります。

アップル社ではデータの安全のために "タイムマシン" と呼ばれる使いやすいバックアップシステムを提供しています。上級ユーザーは他のソフトウェアも選択可能ですが、一般的にはタイムマシンを使用することで、セットアンドフォーゲット（設定後の面倒な操作が不要な）バックアップを実現できます。

図 1 コンピューターがこのディスクへのアクセス許可を要求しています。



アクセス許可

Mac OS Xには、アプリケーション（および他のユーザー）がディスクに直接アクセスするのを防ぐセキュリティ機能が組み込まれています。そのため、ディスク選択時に iDefrag がアクセス許可を要求してくることがあります。その場合には、[図 1 \(P.3\)](#) に示すようなロックボタンが表示されます。

このロックボタンをクリックし、入力を要求されたら OS X のユーザー名とパスワードを入力してください。iDefrag の購入時に提供されたユーザー名とパスワードではありません。

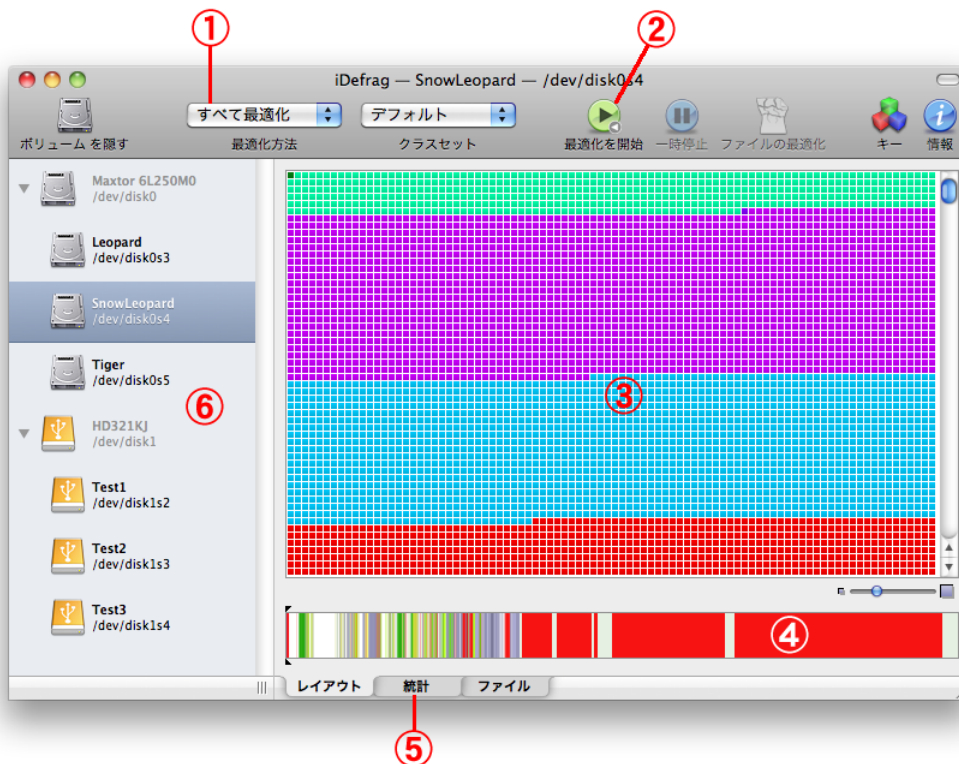


ロックボタンが表示された場合、特定の操作を行うための許可をコンピューターが要求しています。画面に表示されるウィンドウの内容を見て、どのプログラムがどのような理由で許可を要求しているのかを確認し、何が発生するのか予想できない場合には、OS X 管理者ログインの詳細情報のみを入力してください。

メイン表示

以下の [図 2](#) は iDefrag のメインウィンドウです。

図 2 iDefrag のメインウィンドウ



1. このポップアップリストを使用して、実行する最適化アルゴリズムを選択できます。
各アルゴリズムについては「[アルゴリズム](#)」(P.7)を参照してください。
2. 準備ができたら、[開始] ボタンをクリックしてデフラグを開始します。
[開始] ボタンの下の方に、白い左矢印を含んだグレーの小さい丸が表示されます。
これは再起動バッジで、このディスクをデフラグするためにコンピューターを再起動することを iDefrag が要求しています。再起動せずにデフラグできるディスクもあり、その場合にはこのバッジは表示されません。
3. ウィンドウのこの領域内の四角は、それぞれファイルシステム内の単一のブロックを表し、ブロックを使用しているファイルの種類（またはクラス）が色分けされています。
例えば、紫色はジャーナルファイルを表します。iDefrag では、明暗パターン内の同じ色の隣接するブロックが影で表示されるため、ファイルまたは断片の終わりを確認できます。
赤色のものはすべて断片化されています。



デフォルトの色を変更したい場合は、[キー] パネルから色を変更できます。
将来のセッション用に、このカスタムカラーが iDefrag のプレファレンスに保存されます。デフォルト設定と同様に、設定した色が明暗表示されるため、ディスク上の個々の断片を確認できます。

4. ウィンドウのこの部分には、ディスク全体が表示されます。白い部分は空き領域を表します。
この例では、断片化ファイル（赤色）が多数あり、空き領域が（白色）が少ないことを確認できます。



多くの外部ドライブは出荷時に FAT でフォーマットされています。
外部ディスクを購入すると、多くの場合、PC で使用できるようにフォーマットされています。
パフォーマンスを最大化し、Mac OS X の機能をすべて使用できるように、ディスクユーティリティの [Mac OS 拡張 (ジャーナリング)] オプションを使用する前に、外部ディスクを再フォーマットすることを推奨しています。

5. 一番下には、このボリュームの詳細を確認するためのタブがいくつかあります。
タブの詳細については、ヘルプを参照してください。
6. 左側には、選択可能なボリュームのリストが表示されます。
iDefrag がサポートする Mac ファイルシステムフォーマット (HFS+) のボリュームのみが表示され、FAT や NTFS などの Windows フォーマットは表示されません。



[クラスセット] ポップアップの機能

上級ユーザーは、ファイルの新カテゴリ（またはクラス）を作成し、「クラスセット」ファイル内に自分専用のクラスを定義することによって、それらのクラスに対して iDefrag でどのような処理をするか指定できます。
詳細については iDefrag のヘルプを参照してください。

スタートアップボリュームのデフラグ



iDefrag では、クイック（オンライン）アルゴリズムを使用する場合を除き、デフラグするボリュームに排他アクセスする必要があります。

他のプログラムがディスクを使用していると、排他アクセスできない可能性があります。状況によっては、他のプログラムを終了することでこの問題を解決できますが、スタートアップボリュームに関しては、オペレーティングシステムがディスクを使用しているため、下記のいずれかのオプションを使用する必要があります。

また、ウィルス対策プログラムなどのバックグラウンドソフトウェアを実行していると、iDefrag で排他アクセスできない場合があります。このようなケースでは、以下のオプションが役立つ場合があります。

最初にデフラグを行わずに iDefrag の操作を開始したい場合には、このセクションは省略し、必要に応じて参照してください。

スタートアップボリュームをデフラグするには、次のいずれかの操作を実行できます：

- iDefrag を特定のモードからリスタートするように設定できます。これにより、このモードからボリュームに排他アクセスできるようになります。iDefrag が必要に応じてこのモードを設定します。約 1GB の空き領域が必要で、iDefrag の実行中は他のプログラムを実行できません。
- 起動用 DVD/CD を作成して使用できます。これを行うには、iDefrag メニューから [ブートディスクの作成] を選択して、画面上の指示に従います。DVD/CD 書き込みプログラムと記録メディアが必要です。DVD/CD から起動すると、通常のスタートアップボリュームと比べて、起動にかなりの時間（数分）がかかる場合があります。
- 別の Mac がある場合は、[ターゲットディスクモード] を使用できます。このモードの実行方法については、アップル社のサポートサイトの次のページを参照してください。
http://support.apple.com/kb/HT1661?viewlocale=ja_JP



旧型の PowerPC (G4) Mac の中には、[FireWire ターゲットディスクモード] で不具合が発生するものがあります。

それらのマシンでは、FireWire ロゴの表示の移動が停止し、それが原因で iDefrag がレインボーカーソルを表示する場合があります。

旧型の PowerPC システムでは別の方法でお試してください。

- 外部ドライブまたは、パーティション化したメインのハードディスクドライブがある場合には、そのドライブに OS X をインストールし、そのドライブから起動できます。

ここでは詳しい手順について説明しませんが、これに関する記事はインターネットで検索できるでしょう。

アルゴリズム

iDefragには、様々な最適化アルゴリズムが用意されていて、多くのユーザーは、最初にどのアルゴリズムを使用するか悩むようです。

ここでは、iDefragで使用可能な以下のアルゴリズムについて説明します。

すべて最適化

どのアルゴリズムを選択してよいかわからない場合には、「すべて最適化」を選択してください（これは実際には、メタデータ/カスタム・アルゴリズムを組み合わせたものです）。「すべて最適化」は、一般ユーザー向けの最も包括的なアルゴリズムです。これを実行するとすべてのファイルがデフラグされ、ディスク上で最適な位置に配置されます。

コンパクト

このアルゴリズムは、空き領域を圧縮しますがファイルをデフラグしません。「すべて最適化」よりはるかに高速です。**Boot Camp** アシスタントで問題が発生した場合や、「すべて最適化」オプションを実行する時間がない場合にはこれを使用してください。このアルゴリズムはファイルをデフラグしませんが、OS Xに組み込まれているデフラグ機能（空き領域の断片化が増えすぎたときに処理を中止する）を使用可能にできるため、極めて効果的です。

クイック（オンライン）

このアルゴリズムは、デフラグ対象のボリュームにiDefragから排他アクセスできない場合や、再起動する時間がない場合に役立ちます。これは、現在使用していないファイル（すべてのシステムファイルを除外する）に対してのみ動作します。

カスタム/メタデータ

これらのアルゴリズムの詳細については、ヘルプを参照してください。特別な理由がない限り、これらの2つのアルゴリズムを組み合わせた「すべて最適化」アルゴリズムを使用してください。

ディスクにデフラグが必要かどうか？の判断

iDefragには、デフラグが必要かどうかを判断するための方法がいくつか用意されていますが、最も簡単な方法は、画面最下部のディスク全体の表示を確認することです。下の図3では、断片化が多すぎるディスクのスクリーンショットを確認できます。

図3 断片化が多すぎるディスク



空き領域を示す白色の大きな領域がなく、断片化ファイルを示す赤色の部分がかかなり多くなっています。

別の方法として「統計」タブを確認できます。このタブには、ボリュームに関する詳細情報が表示されます。各数字の正確な意味については、iDefragのヘルプを参照してください。デフラグを実行するタイミングについては、「統計」タブ内の数値をメモしておくのとデフラグ実行前後の数値の変化を確認できますので、それを元に調整すればよいでしょう。

次のステップ

デフラグを開始する準備ができれば、「開始」ボタンをクリックします。

使用するアルゴリズムの種類によっては、このプロセスに時間がかかる場合があります。フルデフラグアルゴリズムでは数時間かかる場合があるため、夜間に実行することをお奨めします。iDefragの完了後の処理、例えばマシンの電源を切るなどの処理を選択できます。iDefragメニューから「環境設定...」を選択してください。

オフラインアルゴリズムの場合は、iDefragの実行対象となるボリュームにアクセスできませんが、マウント済みの他のボリュームに関しては処理を継続できます。起動ディスクからiDefragを実行している場合は、選択肢が限られます。

デフラグはいつでも中断できますが、ボリュームにアクセスできなくなります。また、iDefragがプロセッサリソースを使用できなくなります。デフラグの中断は、マシンのプロセッサにしばらく間、他のタスクを実行させたい場合に役立ちます。

温度の監視

デフラグの実行中は、iDefrag がディスクの温度を監視し、[環境設定...] に定義済みのしきい値を超えると実行を中止します。デフォルトでは、一般的なディスク向けに 55℃ 以上で中止するように設定されていますが、高温でも動作するディスクがあります。

iDefrag が頻繁に中断する場合には、ディスク動作時の温度をチェックしてください。インターネットからディスクのモデル番号を検索して情報を見つけることができます。ディスクのモデル番号は、OS X に付属のシステムプロファイラアプリケーションを使用して見つけることができます。

情報パネル

ツールバーの [情報] ボタンをクリックすると、下の [図 4](#) のような情報パネルが表示されます。

図 4 情報パネル



情報パネルは、iDefrag のメインウィンドウ内のブロックにマウスカーソルを置くと自動更新され、そのブロックに関する詳細情報が表示されます。

いずれかのブロックをクリックすると、そのブロックを含むファイルを構成している全ブロックを選択状態にし、他の位置にマウスカーソルを置いていないときには、このファイルに関する情報が情報パネルに表示されます。

選択状態のファイルが断片化されている場合には、ツールバーの [ファイルの最適化] ボタンをクリックしてファイルをデフラグできます。

他のアプリケーションで使用中のシステムファイル（複数）はデフラグできないため（オンラインアルゴリズムの場合と同様）、いずれかのオフラインアルゴリズムを使用してデフラグする必要があります。

サポート情報

iDefrag に関して問題が発生した場合は、iDefrag のヘルプ、特に「トラブルシューティング」のセクションを参照してください。

また、ネットジャパン製品のサポート資料検索 (<http://www.netjapan.co.jp/r/faq/>) もチェックしてください。

資料や検索で解決策が見つからない場合は、以下のテクニカルサポートにお問い合わせください。

support@netjapan.co.jp

iDefrag に関する不具合の報告、新機能の要望、改善の提案に関しては、iDefrag のメニューオプションか、以下のサポートフォームをご利用ください。

https://www.netjapan.co.jp/r/support/supform/id_supform.php

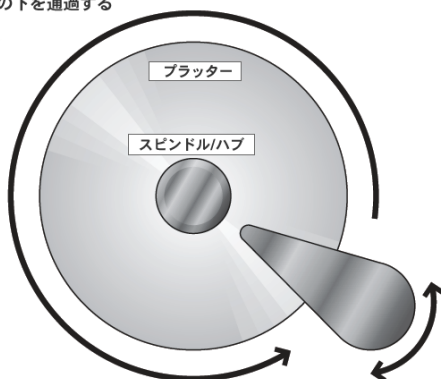
付録 - ファイルシステムとディスク

この付録では、プログラムが表示する内容や最初にディスクをデフラグする理由を理解するのに役立つ、ハードディスクの仕組みに関する概念とコンピューター内部のファイルの保存方法についての説明します。

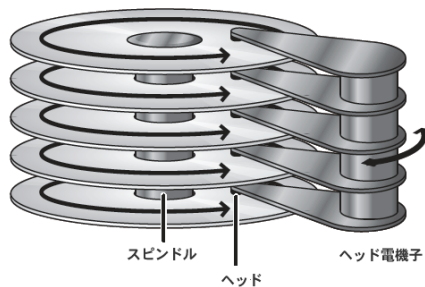
ハードディスクについて

図5 一般のハードディスクの内部には、回転するスピンドルに1組のプラッターが固定され、サーフェス（磁気面）のすぐ上を浮動するディスクヘッド（複数）があります。

プラッターが回転し、
ヘッドの下を通過する



一般のドライブには、中央のスピンドルに
複数のプラッターが固定され、ヘッド電機子に
複数のヘッドが固定されている



ヘッド電機子が回転し、ディスク
サーフェスに沿ってヘッドを移動する

ハードディスクのファイルにデータが記録されることは誰もが知っていますが、実際の仕組みを知る人、また仕組みを理解しようとする人は少ないかもしれません。

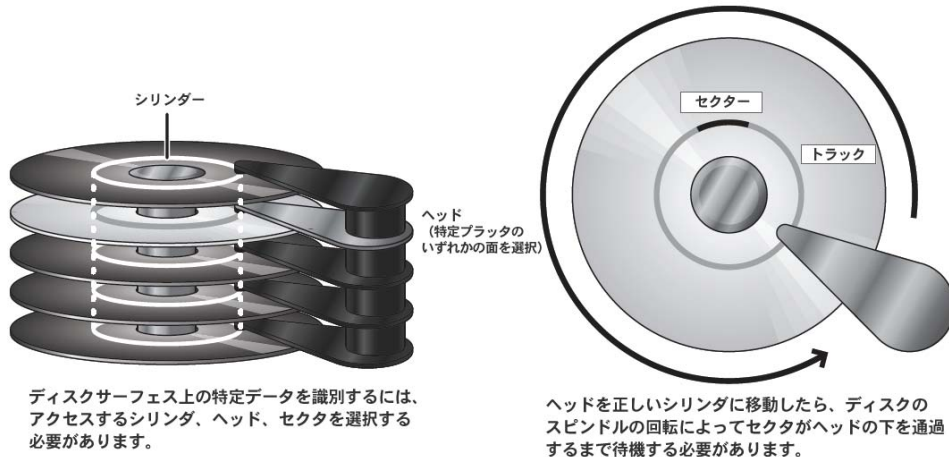
以下に磁気ディスクについて説明します。テープ、光ディスク、ソリッドステートストレージデバイスなどを含む他のストレージデバイスの仕組みは異なります。フロッピーディスクやハードディスクなどの磁気ディスクは次のパーツで構成されます。

- 磁気媒体でコーティングされた1つ以上の円盤型プラッター（ディスクメディア）
- プラッターの磁気面上の信号を読み取ったり、磁気面に信号を記録するための1つ以上のディスクヘッド
- プラッターを高速（通常のハードディスクは5,400rpm以上）で回転させるスピンドルモーター
- ディスクヘッドを内側/外側に移動できるアクチュエーター（通常、フロッピーディスクの場合はステッピングモーター、ハードディスクの場合は音声コイルが使用されます）

図5は、一般的なハードディスク内部の簡略図です。プラッターごとにヘッドが2つあり（サーフェスごとに1つ）、プラッターと電機子はかなり接近しています。ディスクヘッド自体は非常に小さいパーツです。

ディスクからデータを読み取るには、ディスクドライブのスピンドルから正しい距離の位置にヘッドを位置設定し、データが記録されているプラッター用のヘッドの下にデータが通過するまで待機する必要があります。ヘッドの位置は、シリンダー番号で定義されているスピンドルからの相対位置になります。シリンダー番号とヘッド番号を組み合わせると、特定プラッターの特定面上の円周トラックが定義されます。各トラックはさらにセクターに分割され、通常は各セクターに512バイトのデータを記録できます。そのため、シリンダー番号、ヘッド番号、セクター番号を組み合わせることで、ディスクメディア上のセクターを一意的に識別できます。

図 6 シリンダー番号、ヘッド番号、セクター番号による特定セクターの選択



ディスクドライブのメカニズムに関しては、ヘッドの選択は電子的に実行され、基本的に時間はかかりません。特定セクターがディスクヘッドの下を通過するまでに要する時間は、ドライブの回転速度とセクターサイズで決まります。これが高速回転のハードディスクが好まれる理由ですが、高速に回転するということは、データの各ビットがヘッドの下を通過することになり、結果として、データ転送も高速になることを意味します。

シリンダーの変更は比較的遅く、最近のドライブで数ミリ秒ほどかかります。最近のドライブでよく使われる位置調整機構は、ヘッドを1シリンダー分どちらの方向にも高速に移動でき、また1シリンダーをどちらの方向にも確実に移動できます。ただし、移動量が多い場合、ヘッドを正しい位置に移動するのを保証できるほど精密な機構になっていないため、ヘッドが現在あるトラックをドライブに識別させるための情報を、各トラックに記録しています。

ドライブが最初に正しいトラックに移動できなかった場合には、正確な位置が見つかるまで少しずつ移動し直す必要があります。

最も遅い移動を一般に"フルストローク"移動と言います。その場合、ディスクヘッドが一番内側のシリンダーから一番外側のシリンダーに、またはこの逆向きに移動します。

図 6 は、シリンダー番号、ヘッド番号、セクター番号を使用してディスク上の特定セクターを見つけるプロセスを示したものです。

SCSIディスクを除く旧式のディスクでは、通常、ホストコンピューターがCHS（シリンダーヘッドセクター）とも呼ばれるこの形式でデータを配置することが期待されます。しかし、この方式にはいくつかの短所があり、シリンダーとヘッドの数がディスクごとに異なるため、トラックごとに異なる番号のセクターを選択する必要があります。特定ディスクのシリンダー、ヘッド、セクターの配置を"ジオメトリ"と言います。実際のCHS値を使用すると、磁気メディアが損傷した場合、ディスク上のセクターの再割り当てが困難になります。その場合、コンピューターのソフトウェアでこの問題を解決する必要があります。

最後になりますが、ディスクのサイズが大きくなると、シリンダー番号、ヘッド番号、セクター番号用に確保されているフィールド（ソフトウェアおよびハードウェア上）が小さくなりすぎて、処理できなくなります。その結果、IDEディスクのジオメトリに関する情報（PC BIOSとの互換性を維持するための情報）がホストコンピューターに伝わらないため、コンピューターが自動的に不良セクターを管理し始める可能性があります。

最近のディスクでは通常、CHS値は使用されず、ゼロまたは1から始まる番号がセクターに割り当てられます。これを"論理ブロックアドレス（LBA）"と言います。この方法でアクセスする場合、ディスクの製造業者は、ドライブのメカニズムのパフォーマンスを最適化できるように、物理メディア上の各セクターの位置を自由に選択できます。通常は、ブロックをLBAの順序でアクセスした場合に、パフォーマンスを確実に高める方法でブロックが割り当てられます。

ファイルシステム、ボリューム、パーティション

ディスクは、512バイトのデータを記録できる番号付きのセクター配列と見なすことができます。セクターサイズがこのサイズと異なるディスクがありますが、現在使用されているほとんどのハードディスクは、このような512バイトブロックを採用しています。

しかし、すべてのディスクが同じ方式とは限りません。1.44MBフロッピーディスクでさえ、2つのヘッド（ディスクの両面に対応する）があり、80シリンダーと各トラックに18セクターが保持され、合計セクター数は $2 \times 80 \times 18 = 2,880$ になります。使用中のセクターとディスク上のファイル位置を追跡するには、複雑な機構が必要となります。

一方、最近の500GBハードディスクには数十億のセクターがあり、何十万というファイル、場合によって数十億のファイルを保存できます。先週保存した写真をどのようにして見つけられるでしょう？複数のファイリングキャビネットを作成する必要があります。

この単調な作業はコンピューターが得意とするものです。貴重なすべてのデータを追跡記録するソフトウェアを"ファイルシステムドライバー"と言います。例えば、アップル社のHFS+（Mac OS Extendedとも呼ばれる）やMicrosoftのFAT/NTFSファイルシステムがよく知られています。

ディスクのどの位置にファイルが保存されているか、それらのセクターが現在使用中かどうか、などの情報を保存する方法はファイルシステムごとに異なりますが、共通点としてファイルシステムで使用できるようにフォーマットされたディスクを使い、ファイルの名前があれば、そのファイル用のデータがそのディスク上のどのセクターに保持されているかを判断し、そのファイルにアクセスしようとしているプログラムにこのデータを返すことができます。

重要なことは、ファイルシステムという用語が、人によっては違う意味に解釈されることです。ディスク上のファイル位置を特定するコンピューターコードを意味することもあれば、特定フォーマットのディスクを意味することもあります。また、データ構造そのものの理論的な設計を意味する場合もあります。

これらは大きな混乱を招くため、アップル社では、次の用語を使用することを推奨しています：

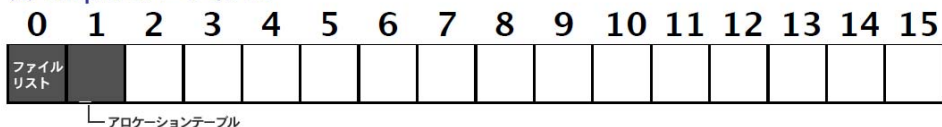
- **ファイルシステム：**
データ構造やアルゴリズムを含む理論的な設計、すなわち、ファイルシステムの仕様。実際にはデータの記録方法をユーザーに示す取扱説明書に記載されている情報。
- **ファイルシステムドライバ：**
理論的な設計を実現するコンピュータープログラム。このプログラムにより、ファイルシステム仕様に基づいてフォーマットされたディスクを Mac で読み取ることが可能。
- **ボリューム：**
ファイルシステム仕様に基づいてフォーマットされた特定ディスクの内容。

さらに、1つのボリュームを保持するためにディスク全体を使用するのではなく、ディスクをパーティションと呼ばれる区画に分割して、それぞれのパーティションに独立したボリュームを保持できます。

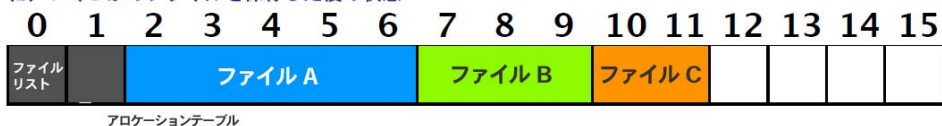
ファイルシステムの仕組みを理解するために、仮想的に SimpleFS と呼ぶファイルシステムを考えてみます。

図 7 SimpleFS バージョン 1.0

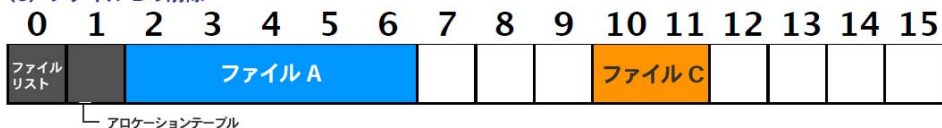
(a) SimpleFS バージョン 1.0



(b) いくつかのファイルを保存した後の状態



(c) ファイル B の削除



SimpleFS バージョン 1 では、ディスク上の連続する領域にファイルを保存できます。

ここで、合計 16 ブロックを含む小さいディスクを想定してみます。一連のファイルを保持するファイルリストに、ブロックを 1 つ割り当てます。将来的に大きなディスクを処理できるよう、(a) に示すように、どのブロックが割り当てられているかに関する情報を保持するアロケーションテーブルに、別のブロックを割り当てます。

ここでは、この情報がこれらの予約ブロックにどのように保存されるかについては言及しません。

ディスクに1つのファイルを保存し、そのファイルのサイズが3,000バイトであると想定します。このファイル内のブロック数は $3,000 \div 512 = 5.9$ となります。SimpleFSでは、ブロック全体でしか割り当てられないため、6ブロックを使用してこのファイルを保存します。これを"ファイルA"とします。また、1,300バイトサイズ(3ブロック)の"ファイルB"と、1,000バイトサイズ(2ブロック)の"ファイルC"を保存します。この結果は図7(P.14)(b)のようになります。

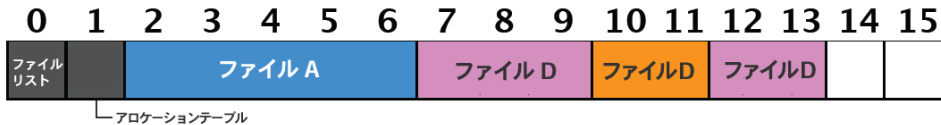
次に、ディスクから"ファイルB"を削除します。この結果は図7(P.14)(c)のようになります。ここで、新しいファイル"ファイルD"を保存すると想定します。この新しいファイルのサイズは2,400バイト(5ブロック)です。7ブロックの空きがありますが、連続する最大領域は4ブロックしかありません。これは問題です。たとえ使用可能なブロックの50%が未割り当てとマークされていたとしても、新しいファイルを保存できません。

ここで、これらのすべてのブロックを使用してデータを保存できるように新しいSimpleFS 2.0にディスクをアップグレードします。5ブロックを含んだ"ファイルD"を保存したいため、最初の3ブロックをブロック7、8、9に、残りをブロック12と13に保存します。この結果は図8(a)のようになります。

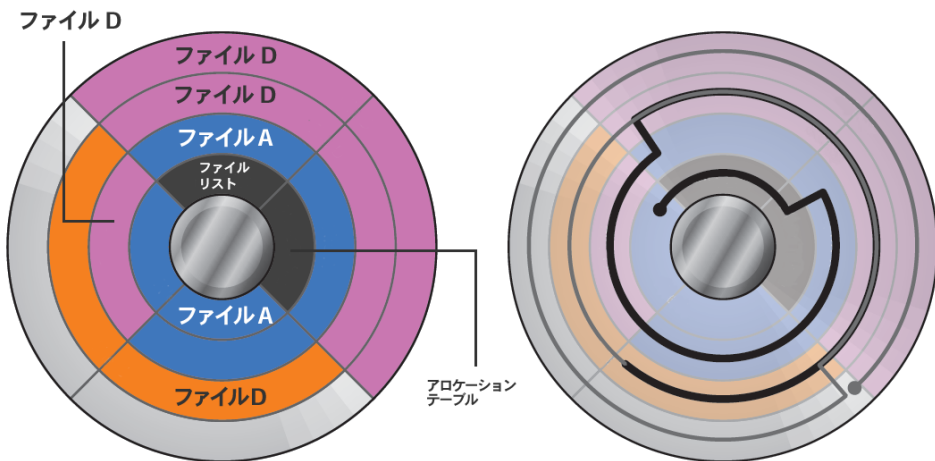
これは非常に効果的な方法です。必要に応じて、ディスク上のどのブロックも使用できます。

図8 SimpleFSバージョン2.0

(a) ファイルDを含んだ SimpleFSバージョン2.0



(b) ディスク上のレイアウト



ディスク上のブロックのレイアウト。
この場合、リニアブロックアドレスを物理ディスクブロックに単純にマッピングすることを想定しています。

ファイルDを読み取るためにディスクヘッドが辿る経路

しかし、ここで問題が発生します。SimpleFS用にフォーマットしたディスクが旧型で低速だと仮定します。便宜上、片面しかない単一のプラッターを想定し、1つのヘッドと4つのシリンダーがあり、各トラックに4セクターが保持されていると仮定します。

次のパートでは、[図 8 \(P.15\)](#) (b) の右側の黒いスポットから開始し、ディスク上のラインを時計回りに追跡することで、記録されている状態を確認できます。

ディスクから"ファイルD"を読み取る場合、最初にファイルリストまで進み、そのリストからそのファイルを見つける必要があります。ファイルリストにはディスク上の"ファイルD"の位置が示されているため、ディスク上で読み取りたい次のセクターがブロック7であり、そのブロックがトラック1にあることがわかります。現在位置はトラック0であるため、ヘッドを1トラックだけ外側に移動することになります。しかし、現在位置がブロック7を読み取る位置ではないため、ディスクが回転してヘッドがブロック7の上にくるまで待機する必要があります。正しい位置にきたら、そのブロックを読み取ることができます。ブロック7を読み取ったら、次にトラック2にあるブロック8を読み取る必要があります。ディスクヘッドはトラック1にあるため、ヘッドを再び外側に移動します。この場合も、現在位置がブロック8を読み取る位置ではない(開始位置を過ぎていた)ため、ディスクが回転するまで待機する必要があります。

ここまでくればディスクヘッドを移動することなく、ブロック8と9を順番に読み取ることができます。ただし、次のブロックはブロック12です。これは外側のトラック3にあるため、ヘッドを1トラック外側に移動する必要があります。この場合も、現在位置が正しい位置ではないため、ディスクが回転するまで待機する必要があります。

最後に、ブロック12と13を順次読み取って完了します。

このプロセス中では、ディスクが3回半回転して、ヘッドを3回移動する必要がありました。トラック0の開始位置への最初の移動はカウントしていません。

このディスクが非常に低速で、ヘッドを移動するのに0.5秒かかり、20rpmで回転すると仮定します。このファイルを読み取るのに全部で $3 \times 0.5 + 3.5 \div 20 \times 60 = 12$ 秒かかったこととなります。

ここで、"ファイルD"の終わりの位置を"ファイルC"に置き換えた場合にどのようになるか考えてみます。この単純な変更により、直前の2つのブロックと同じトラックに"ファイルD"の最終ブロックが置かれるため、ディスクヘッドを移動したりディスクの回転を待つことなく、ブロック8、9、10、11を順番に読み取ることができます。1回のヘッド移動と1回半の回転が不要になります。

このケースでは、読み取りに $2 \times 0.5 + 2 \div 20 \times 60 = 7$ 秒しかかかりません。

ディスク上で各ブロックが隣接するように"ファイルD"のブロックを移動するだけで、時間を42%節約できます。

"ファイルD"のように、ファイルを複数の小さい領域に分割した場合、"ファイルが断片化されている"という言い方をします。各領域を"断片(フラグメント)"と言います。MacのファイルシステムHFS+では、ファイルシステムでファイルの断片に関する情報を保持する方法を表す用語として、"エクステンツ"が使われることがあります。

この(最も単純な)例からわかるように、ファイルシステムでファイルを断片化させることは、トレードオフになります。断片化することでディスク上の一部の領域が使用可能になりますが、断片化したファイルは、断片化しない場合よりも読み取りや書き込みに時間がかかります。

つまり、パフォーマンスの速度を優先させるには、不必要にファイルを断片化させないことが重要です。

Macで断片化を防ぐ方法

「Macではデフラグする必要はない」という意見を聞いたことがあるかもしれません。実際、MacのファイルシステムHFS+には、断片化を防ぎ、マシンのパフォーマンスを改善する機能がいくつか組み込まれています。

最もシンプルな機能では、ディスクに割り当てるブロックを選択するという方法が使われています。コンピュータープログラムは通常、ファイルを保存する前に、そのファイルのサイズをコンピューターに知らせません。たとえコンピューターがディスクへの書き込みを遅らせたとしても、その遅延時間内にファイル全体のサイズを知ることはできません。そのため、正しいサイズの領域を選択することは困難です。

ファイルをできる限り断片化させないという目標があるときは、これは大きな欠点になることは明らかですが、コンピューター研究者は、割り当てる領域を選択する他の方法よりも、正しいサイズ用の正確な領域を選択すること（ベストフィット）が実用上優れているとは言えないとしています。

割り当て用として、他にも様々な方法が吟味されています。例えば、最初の空き領域を単純に選択する方法があります（これを"**ファーストフィット**"と言います）。特殊なものとしては、最大サイズの空き領域を選択して、常にそれを使用するという方法（ワーストフィット）があります。ここでも、一般的には、どちらの方法もベストフィットよりも悪くないということが実証されています。

多くのファイルシステムでファーストフィット（またはそのバリエーション）を使用するには、最初のブロックの選択時にファイルサイズがわからないという理由のほか、最良または最悪の選択肢を見つけるために、すべての空き領域を検索するのを避けるという理由があるからです。

しかし、アップル社のファイルシステムエンジニアは、ディスクの使い方に関してあることを発見しました。つい最近までは、使用しているハードディスクの容量に対して、ユーザーが保持する全ファイルの容量は、はるかに小さいものでした。そのため、一般的な使い方のユーザーがディスクをすべて使用するには長い時間がかかりました。最初の領域を選択したり"**最適な**"領域を探すのではなく、単に次の領域を選択するだけでよいのです。ディスク上のすべてのブロックを使用できる場合は、保存するファイルのサイズによらず、ファイルを断片化させる必要はなく、使用する次の領域を探すことは容易です。ブックマークまたはそれを示すポインタを保持するだけです。このアプローチは"**ロービングポインタアルゴリズム**"と呼ばれています。

この方式は、ワープロドキュメント、スプレッドシートなど、シンプルなデータファイルを保存するときに大いに役立ちました。このようなデータファイルを取り扱う一般的なユーザーなら、ハードディスク上のすべてのブロックにアクセスするには長い期間を要するため、コンピューターがファイルを断片化するまで時間がかかりました。

しかし、写真、iTunesやMP3ミュージックファイル、また特にビデオファイルなど、最近のメディアファイルはサイズが肥大化しているため、ユーザーによっては、このシンプルなアプローチは役に立たなくなってきました。

また、Mac OS Xでは、HFS+に制限付きの自動デフラグ機能が含まれていますが、その機能は限定的です。断片化ファイルが20メガバイト以下で8つ以上の断片化があること、隣接する十分な空き領域が想定されること、その上でユーザーがジャーナリングをオンにしてそのファイルにアクセスするときのみ、Mac OS Xは自動的にそのファイルを最適化します。

肥大化するデータ、また主にそのようなデータを取り扱うことの多いMacユーザーにとって、専用のデフラグソフトが求められる理由は、ここにあります。