

ファイルシステムインタフェース

<略>

10.1 ファイルの概念

<略>

10.1.1 ファイルの属性

<略>

10.1.2 ファイルの操作

ファイルは抽象データ型 (abstract data type) である。ファイルを正しく定義するには、ファイルに対して実行される操作を考えなければならない。オペレーティングシステムは、ファイルの作成、書込み、読出し、位置変更、削除、切詰め (truncation) を行うシステムコールを提供している。オペレーティングシステムがこれら六つのファイル操作を実行するのに、それぞれ、どのようなことをしなければならないか調べてみよう。そうすれば、ファイル名の変更など、類似した操作がどのように実現されているかを容易に知ることができるはずである。

- ファイルの作成：ファイルを作成するには二つのステップが必要である。まず、ファイルシステムの中に、そのファイルのための領域を見つけなければならない。第11章で、ファイルの領域をいかに割り当てるかについて論じる。次に、ディレクトリに新しいファイルの項目を作成する必要がある。
- ファイルの書込み：ファイルに書込みを行うには、ファイル名とファイルに書くべき情報の両方を指定するシステムコールを呼び出す。与えられたファイル名に対し、システムはディレクトリを探索して、ファイルの位置を求める。システムは、次の書込みが行われるファイル中の位置を指す、「書込み」ポインタを保持している必要がある。書込みポインタは、書込みが行われるたびに更新される。
- ファイルの読出し：ファイルから読み出すには、ファイル名と (メモリ内の) どこにファイルの次のブロックを置けばよいかを指定するシステムコールを呼び出す。ここでも、当該の項目を見つけるためディレクトリが探索され、システムは、次の読出しが行われるファイル中の位置を指す、「読出し」ポインタを保持する。読出しが行われると、読出しポインタが更新される。プロセスは、通常、ファイルから読み出すかファイルに書き込むかどちらかなので、現在の操作位置は、プロセスに一つの現在ファイル位置ポインタ (current-file-position pointer) に保持しておくことが可能である。読出しにも書込みにもこの同じポインタを用いることで、領域を節約し、システムが複雑になるのを抑えている。
- ファイルの位置変更：ディレクトリを探索して適当な項目を見つけ、現在ファイル位置ポ

インタを与えられた値に設定する。ファイルの位置変更は、実際の入出力操作を必要としない。このファイル操作はまた、ファイルの「シーク」(seek)とも呼ばれている。

- ファイルの削除：ファイルを削除するには、ディレクトリ項目を探索し、指定した名前を持つファイルを見つける。当該のディレクトリ項目を見つけると、すべてのファイルの領域を解放して他のファイルが再利用できるようにし、そのディレクトリ項目を消去する。
- ファイルの切詰め：ファイルの属性を保持したまま、ファイルを消去したいという場合、この関数があれば、ファイルのサイズ以外のすべての属性を変えないで、ファイルの長さをゼロにし、ファイルの領域を解放することができる。ユーザは、ファイルをいったん削除し、作成し直すようなことをしなくてもよい。

これら六つの操作は、必要なファイル操作の最小セットである。その他、共通に見られる操作に、既存のファイルの最後への新しい情報の「追加」(append)や、既存のファイルの「改名」(rename)がある。これらの基本操作を結合して、他のファイル操作を実行することもできる。たとえば、ファイルの「複製」を作成したり、ファイルをプリンタやディスプレイなど、他の入出力デバイスにコピーしたりすることができる。これは、新しいファイルを作成し、古いファイルから読み出した内容を新しいファイルに書き込むという操作による。また、ファイルの様々な属性を設定したり、読み出したりする操作が欲しい場合もある。たとえば、ユーザは、ファイルの長さなどファイルの状態を特定したり、所有者などの属性を設定する操作が欲しい場合があるかもしれない。

上に述べたファイル操作のほとんどは、指定された名前のファイルの項目を見つけるためにディレクトリの探索を行っている。この常に行われる探索を避けるため、多くのシステムでは、ファイルを最初に使用する前に `open()` システムコールを呼び出すことを義務付けている。オペレーティングシステムは、オープンしたすべてのファイルに関する情報を含む、オープンファイルテーブル (open-file table) という小さなテーブルを持っている。ファイル操作が要求されるときは、ファイルはこのテーブルのインデックスで指定されるため、ディレクトリの探索を行う必要はない。ファイルをこれ以上使用しないというときには、プロセスは `close()` システムコールを呼び出す。これにより、オペレーティングシステムは、オープンファイルテーブルから当該の項目を削除する。`create` および `delete` システムコールは、オープンしているファイルではなく、クローズしているファイルに対して適用される。

システムによっては、最初の参照がなされたときに、暗黙のうちにファイルをオープンするものもある。ファイルをオープンしていたジョブやプログラムが終了すると、ファイルは自動的にクローズされる。しかしながら、ほとんどのシステムでは、プログラマがファイルをオープンする際、`open()` で明示的に指定しなければならない。`open()` 操作は、ファイル名を入力としてディレクトリを探索し、見つけたディレクトリの項目をオープンファイルテーブルにコピーする。`open()` システムコールはまた、アクセスモード情報 (作成、読取り専用、読み書き、追加専用など) を入力として受け取る。このモードは、ファイルのアクセス許可と照合される。要求されたモードが許可されたとき、ファイルは呼び出したプロセスに対してオープンされる。`open()` システムコールは、通常、オープンファイルテーブルの項目を指すポインタを返す。このポインタが、実際のファイル名の代わりに、あらゆる入出力操作に使用されるため、余分な探索が不要となり、システムコールのインタフェースが簡略化される。