[8-3.] NTFS のセキュリティ機能と落とし穴

Windowsプラットフォームでセキュリティを重視したシステムを構築する際にはファイルシステムとしてNTFSを採用することが不可欠である。とくにアクセスコントロール機能は重要だ。しかしNTFSには機能が多いために，ときとして思いがけない副作用に見舞われることもある。

● ● ●

NTFS ファイルシステム

NTFS はその名の通り，Windows NT のファイルシステムである。Windows NT の開発が進められる過程でこれまでのパーソナルコンピュータとは一線を画した（1994 年当時としては）高度で信頼性の高いファイルシステムを実現し，その上で稼動する Windows NT の信頼性そのものを高める目的で作られたものである。

セキュリティに関する十分な考慮が必要なソフトウェアシステムをWindowsプラットフォームで構築する際は，NTFS ファイルシステムの使用が欠かせない。

NTFS は当初から次のような機能を備えていた。

- 64 ビット整数によるファイルサイズ（最大 約1.8×10^{19} バイト）
- 複数データストリーム
- データ圧縮機能
- ファイル名への Unicode の採用
- 高速なファイル名検索
- アクセスコントロールリスト（ACL）によるきめ細かなアクセス制御
- 監査ログ機能
- トランザクション処理モデルにもとづくエラーリカバリ機能
- ミラーリングやストライピングによる信頼性向上機能
- 不良セクタの動的再割り当て
- POSIX サポート（ハードリンク，大文字小文字の区別のあるファイル名，互換性のあるタイムスタンプ）

これらのうちデータ圧縮機能だけはWindows NT 3.51に搭載されたNTFS 1.1（注1）で加えられたものだが，他は最初のバージョン NTFS 1.0 から備わっているものである。

さらに Windows 2000 の NTFS 5.0 では次の機能が搭載された。

- データの暗号化
- スパースファイル（巨大なファイルの中を「飛び飛びに」使用したとき容量が節約される機能）

・マウントポイント（パス名の延長上に別のボリュームを「接ぎ木」する機能）
・リパーズポイント（パス名解釈（パーズ）のカスタムモジュールによりファイルシステムを拡張できる機能。上の「マウントポイント」も実はこの機能を利用している。）
・ディスククォータ（使用量の管理機能）
・変更ジャーナル

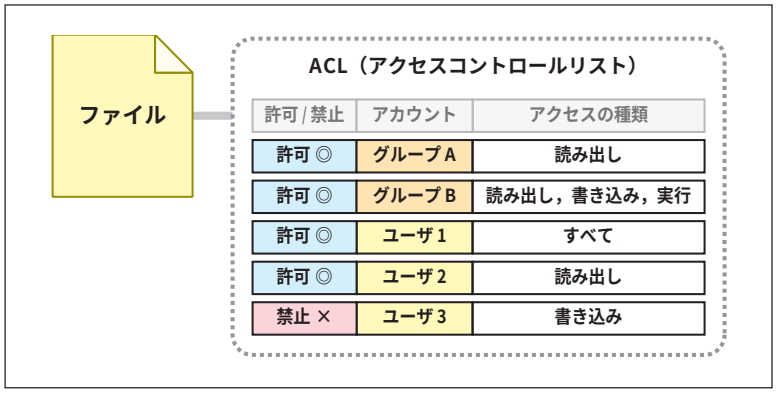
このNTFS 5.0はマイナーバージョンアップされてNTFS 5.1となり，Windows XPに搭載されている。

（**注1**・NTFS 1.1は通称NTFS 4.0とも呼ばれている。Windows NT 4.0に搭載されていることとNTFS 5.0と対比されることが多いためである。NTFS 5.0は正式なバージョン番号である。）

アクセスコントロール

NTFSがもつセキュリティ機能は，アクセスコントロールリスト（ACL）によるきめ細かなアクセス制御，監査ログ機能，データの暗号化などであるが，その中で最も重要なものは，最初に挙げたアクセス制御である。NTFSのアクセス制御はWindows NTカーネルが実現しているOSの機能と直結したものだ。

図1　ACLのイメージ



Windows NTおよびその後継OSのアクセス制御機能の特徴は，システムのオブジェクト（NTFSの場合はファイルやディレクトリ）の一つ一つに対してアクセスコントロールリスト（ACL）を設定できることにある。ACLに，どのユーザやグループのアカウントにどの種類のアクセスを許し，あるいは禁止するかという指定を記述することにより，きめ細かなアクセス制御ができる（**図1**）。

許可したり禁止したりするアクセスの種類も，読み出し，書き込み，削除，実行，ディレクトリの参照に加え，細かなオプションがある。Windows 2000などのユーザインタフェースにはあまり多くの項目は表示されないが，システム内部では32ビットのビットパターンでアクセスの種類が表現されている（**表1**）。

表1　ファイルのアクセス権（アクセスの種類）

ビットパターン（16進）	アクセス権（記号名）
80000000	GENERIC_READ
40000000	GENERIC_WRITE
20000000	GENERIC_EXECUTE
10000000	GENERIC_ALL
01000000	ACCESS_SYSTEM_SECURITY
00100000	SYNCHRONIZE
00080000	WRITE_OWNER
00040000	WRITE_DAC
00020000	READ_CONTROL
00010000	DELETE
00000100	FILE_WRITE_ATTRIBUTES
00000080	FILE_READ_ATTRIBUTES
00000040	FILE_DELETE_CHILD
00000020	FILE_EXECUTE（ファイルの場合）
00000020	FILE_TRAVERSE（ディレクトリの場合）
00000010	FILE_WRITE_EA
00000008	FILE_READ_EA
00000004	FILE_APPEND_DATA
00000002	FILE_WRITE_DATA
00000001	FILE_READ_DATA
（よく使われる組み合わせパターン）	
00120089	FILE_GENERIC_READ
00120116	FILE_GENERIC_WRITE
001200A0	FILE_GENERIC_EXECUTE
001f01ff	FILE_ALL_ACCESS

DACLとSACL

NTFSのACLには実は2種類ある。上記でACLと呼んでいるものは，厳密にはDACL（Descretionary Access Control List）と呼ばれるものだ。もう一つのACLはSACL（System Access Control List）と呼ばれる。SACLはアクセス監査ログを記録する条件を記述するものである。この記事では簡略化して，DACLのことをACLと呼んでいる。

アクセス権設計

Windows 2000 や NT を対象としたアプリケーションを開発の現場では，特定のデータ処理で利用しようとしたファイルにアクセス許可がなくてプログラムがうまく動かない，といった問題に遭遇することがある。

しばしば行われる解決策は，次のようなものだ。

1）ファイルへのアクセス許可をより多くのアカウントに与える
2）プログラムを Administrator あるいは SYSTEM 権限で動作させる

しかし，どちらの方法も危険を招くおそれがある。アクセス許可を広く与えてしまえば（極端な場合「許可：Everyone，フルコントロール」のように），目的とするプログラムだけでなく誰もがそのファイルに干渉できるようになってしまう。

プログラムに高い権限を与えれば，誤りや故意により本来アクセスすべきでないファイルにまで高い権限でアクセスできるから，部外者がそのプログラムを悪用してシステムに悪い影響を与えるおそれがある。このプログラムが何らかのコマンドを外から与えられるようになっているものであれば，最悪の場合システムが乗っ取られることも考えられる。

実際に上記のような対処方法以外，選択肢がないこともある。しかし開発の過程でアクセス拒否の問題に出会ったら，アクセス制限を緩めたりプログラムの権限を高める前にもう一度アクセス権の設計を見直すべきである。もし，アクセス権に関する十分な考慮なしにソフトウェアの開発を始めてしまっているとしたら，この機会に簡単に整理しておくのがよい。

アクセス権設計は，アプリケーションソフトウェアのそれぞれの事情によってさまざまなものに変わり得るが，たとえば次のように類型化して整理する方法がある。

ファイル
- f1）権限の低い一般ユーザが共通にアクセスしてよいファイル
- f2）本人はよいが他人には見せてはならないファイル
- f3）ソフトウェアを運用する際損なわれては困る重要なファイル

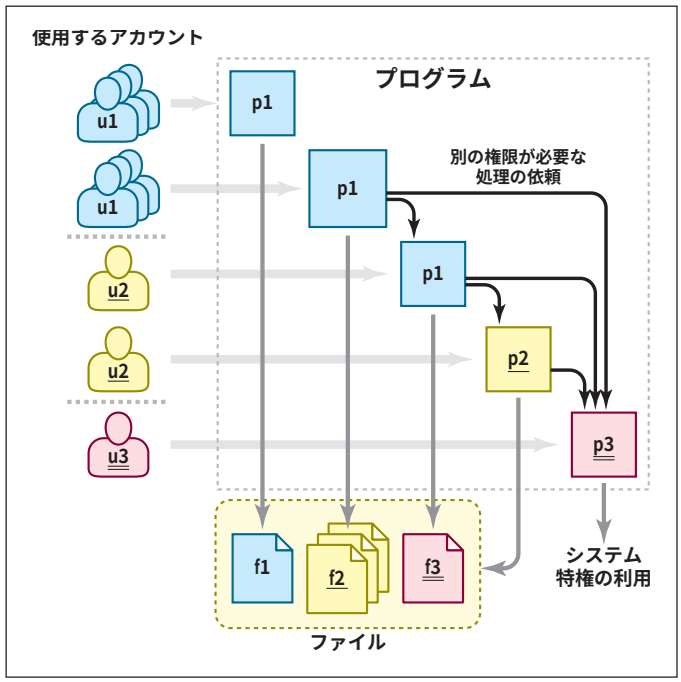
プログラム
- p1）通常のデータ処理が主で高い権限は必要ないプログラム
- p2）バックアップなど，データ全体に影響を及ぼすプログラム
- p3）システムの特別な機能を利用するため高い権限が必要なプログラム

アカウント
- u1）一般ユーザ用アカウント
- u2）アプリケーション管理用アカウント
- u3）コンピュータ管理者アカウント（Administrator）

これらの組み合わせ次のようになるように調整するのである（**図２**）。

1）通常のファイル処理は特別なことを行わないプログラムから行う（p1）

図２　アクセス権設計: 権限の分離と絞り込み



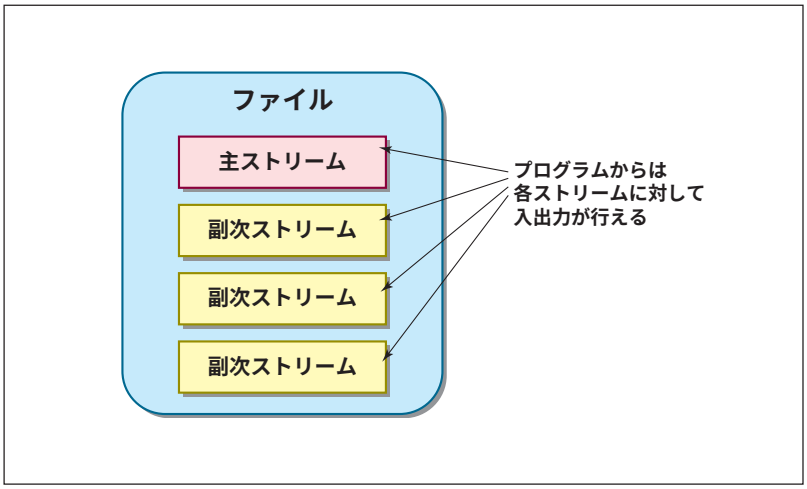
2）特別な処理（p2，p3）は分離してそこだけ高い権限のアカウント（u2，u3）を使う。

3）別の権限による処理が必要な場合は，プロセス間通信などを用いて該当するプログラムに処理を依頼する形にプログラムを分割する。

複数データストリーム

ファイルシステムのファイルといえば「一つのいれもの」としてしばしば捉えられるが，実はNTFSのファイルはそれ以上の収容能力を持っている。NTFSのファイルは一つのファイルの下に複数のデータストリームを保持できるのである。いわば「ファイルの中に複数のファイルがある」といった構図だ（**図3**）。

図3　NTFSのファイルは複数のデータストリームをもてる



ふだんわれわれがデータをファイルに保存するとき，NTFSのファイルがもつことのできるデータストリームのうち一つしか使っていないのである。

プログラムからはNTFSのファイルの各ストリームに対して入出力を行うことができる。ここのストリームは次のような形式のパス名で識別される。

```
d:\dir\file.ext:stream
```

（ここに，`d:\dir\`＝ディレクトリ，`file.ext`＝ファイル名，`stream`＝ストリーム名　である）

すなわち，ファイル名の後ろにコロン（:）で区切ってストリーム名をつければよい。

ストリーム名の指定を省略すると，ファイルの主内容を保持する「無名の」ストリームが指定された，と解釈される。これで，複数ストリームを持たない他の（通常の）ファイルシステムのファイルとの互換性が保たれている。

副次ストリーム

ファイルの主内容を保持する無名ストリームは「主ストリーム」(prime stream)，それ以外の名前を持つストリームは「副次ストリーム」(alternate stream) とそれぞれ呼ばれる。副次ストリームはこれまであまり目立つ場所で使われてこなかった。Macintoshに対するファイルサーバ機能や，Windows 2000で各ファイルにさまざまなプロパティを記録できる機能がその少ない実例である。

注意しなければならないのは，プログラマが全く意識していないときでもこの副次ストリームにアクセスできる機能が常に有効になっていることだ。コロン（:）を含むパス名が入力データなどから入ってきた結果として次のようなステートメント，

```
FILE* pf = fopen ("foo.txt:bar", "w");
```

を実行してしまうと，プログラムが書き込むデータはみなファイルfoo.txtの副次ストリームbarの中に書き込まれてしまう。もちろん，もう一度ストリームを示すパス名「foo.txt:bar」を明示してアクセスすれば支障なくデータは読み出せるが，問題はそれ以外のところにある。

Windows Explorer などをはじめとする，Windows 2000 や NT に通常備わっているプログラムはほとんどこの副次ストリームを意識して作られていない。したがって，どこかのプログラムが書き込んだ副次ストリームはとても見つかりにくい存在なのである。ファイルのサイズを調べても主ストリームのデータ量しか報告されない。

副次ストリームは次のような不正行為に使われるおそれがある。

1）悪意あるプログラムやデータの隠し場所
2）ひそかにファイル容量を消費させ業務を妨害する

もちろん，このほか単に「迷子のデータ」が入り込むといった問題は常に起こり得る。

副次ストリームは通常のファイルユーティリティでは見つけだせないが，Windows 2000やNTのファイルバックアップ機能はNTFS の複数データストリームを意識して作られているので，それを使う方法がある。BackupReadなどのAPIを用いたプログラムを書き，ファイル一つ一つの副次ストリームをリストアップして，その中に不審なものがないか調べるのである（注2）。

（**注2**・バックアップAPIを使ってストリームを取り扱う手法については，参考文献に挙げたWebページ『プログラマから見たNTFS 2000 Part1: ストリームとハードリンク』をご覧いただきたい。）

主ストリームの別名

上で主ストリームは「無名」であると述べたが，技術的には主ストリームに名前を指定して呼び出すことが可能だ。ストリーム名として「:$DATA」を指定するのである。ファイルfoo.txtの主ストリームを参照する名前は，区切りのコロン（:）がさらに加わり次のような形になる。

```
foo.txt::$DATA
```

NTFSの全てのファイルはファイル名の末尾に「::$DATA」を付加した別名を持つことになる。これが問題となるのはファイル名をパラメタとして受け取るプログラムである。たとえば，名前を指定されて実行するがその内容を開示してはならないスクリプトファイルsecret.batがあったとする，ユーザが通常のファイル名の代わりにsecret.bat::$DATA のようなファイル名でこのスクリプトファイルを呼び出したとき，もしあらかじめ定められた拡張子のファイル以外についてはその内容を表示するようなロジックが組まれていたら，秘密でなくてはならないスクリプトの内容が漏洩してしまう。

プログラムでパス名を取り扱うときには，副次ストリームや主ストリームの別名が問題を起こさないような配慮をすべきである。すなわち，ドライブ文字を表す以外の記号コロン（:）が含まれていたらそのパス名を不当なものとして排除するのである（注3）。

（**注3**・Windowsのパス名についてはNTFSのストリーム名以外にも注意すべき点がいくつもある。それらについては関連記事『8-1. Windowsパス名の落とし穴』をご覧いただきたい。）

まとめ

NTFSは高度かつ高信頼性のオペレーティングシステムの稼動を目的として開発されたファイルシステムである。NTFSがもつアクセスコントロール機能はしばしばプログラムの動作を邪魔する存在でもあるが，簡単にそれを迂回させるとセキュリティが弱まってしまう。逆にこれはうまく活用すべきものである。また，副次ストリームのような普段利用しない機能も常に有効になっており，思わぬ問題が生じないようパス名の取り扱いには配慮が必要である。

関連記事

『8-1. Windows パス名の落とし穴』

『9-4. 特権処理の局所化』

参考文献

『インサイド Windows NT 第２版』， David A. Solomon著，有限会社東京ネットワーク情報サービス訳， 1998年，日経 BP ソフトプレス

『NTFS 5.0 ファイルシステムの新しい機能』
http://www.microsoft.com/japan/support/kb/articles/JP183/0/90.asp

『プログラマから見た NTFS 2000 Part1: ストリームとハードリンク』
http://www.microsoft.com/japan/developer/windows2000/techart/ntfs5.asp

『Insider's Computer Dictionary:“ Windows ”』
http://www.atmarkit.co.jp/icd/root/49/5787249.html

『CreateFile』（Win32 API ドキュメント）
http://www.microsoft.com/JAPAN/developer/library/jpwinpf/_win32_createfile.htm

『Inside Windows NT, Second Edition』（英文），David A. Solomon 著，1998 年，Microsoft Press
（『インサイド Windows NT 第２版』の原書）

『Inside Microsoft Windows 2000, Third Edition』（英文），
David A. Solomon，Mark Russinovich 著，2000 年，Microsoft Press

『The Evolution of NTFS: NTFS 1.1』（英文）
http://www.arstechnica.com/paedia/n/ntfs/ntfs4-1.html