

Windows の仕組み

パソコン救済・勉強会(2011(H23).10.23)資料

111015-1

2011(H23).5.15 T.Ogawa

IT Pro (日経 BP 社) に「本当に知っている？ Windows の基礎」という Web 連載記事がありました。それは月刊誌「日経 Windows プロ」の記事を Web 上に公開したものであり、Word に変換すると 233 頁 (36 項目) の膨大なものでした。

この資料は、上記の Web 連載記事を元ネタにして作り始めました。執筆が進むにつれて次々と疑問が湧いてきて、それらをネットで調べ、更にそれから湧いてきた疑問を調べていくうちに、内容が次第に元ネタとかけ離れてしまい、元ネタの倍ほどの内容になってしまいました。

時間の制約から、勉強会では Windows の機能の 3 項目だけを説明しますが、この資料で説明していない項目に興味のある方は元ネタを読んでみてください。この資料の目次に続けて元ネタの「目次と URL」を載せています。また、巻末に用語の解説等も載せていますので参考にしてください。

第 4 章に、最近話題になっている Windows 8 (開発コード名) について、私見を交えて追記しました。実習中心の勉強会と異なり座学中心の勉強会のため、睡魔との戦いになると思いますが、宜しくお付き合いください。

目次

1. Windows を起動する	1
1.1. Windows のブート処理	1
1.1.1. BIOS によるブート処理 (PC/AT 互換機共通)	1
1.1.2. ブート・ストラップ・ローダによるブート処理 (PC/AT 互換機共通)	3
1.1.3. IPL によるブート処理 (Windows 固有)	4
1.1.4. BOOTMNG によるブート処理 (Windows 固有)	5
1.2. 新技術 (Windows のブート関連)	6
1.2.1. HDD の大容量化に伴う GPT 化	6
1.2.2. BIOS の制約回避のための UEFI 化	8
2. Windows をシャットダウンする	9
2.1. Windows のシャットダウン	9
2.1.1. 通常のシャットダウン	9
2.1.2. 電源ボタン長押しによる強制的な電源オフ	10
2.2. Windows の省電力機能	10
2.2.1. Windows 7 の省電力機能	12
2.2.2. Windows Vista の省電力機能	14
2.2.3. Windows XP の省電力機能	14
3. NTFS はクラッシュ直前の状態にファイルを復元できる	15

3.1.	FAT32 と NTFS の比較	16
3.2.	NTFS にはサーバにも使える高度な機能が多い	16
3.3.	NTFS の主役は MFT である	18
3.4.	クラッシュ直前までロールバックしてファイルを復元	20
3.4.1.	クラッシュによるファイル破壊	20
3.4.2.	ファイル更新を効率的かつ安全に行うための手法	20
3.4.3.	NTFS にはファイルを復元できるトランザクション機能がある	21
3.4.4.	ファイル復元のためにログデータを記録とチェックポイント処理	21
3.4.5.	クラッシュ後の起動時にログデータを用いてファイルを復元する	24
4.	PC の買い替えは Windows 8 搭載パソコンを待っては・・・	26
4.1.	Windows 8 の発売時期は？	26
4.2.	Windows 8 はユーザ・インタフェースが激変する？	27
4.3.	Windows 8 の新機能	27
4.4.	Windows 8 でパソコンの仕様は変わる？	28
	本文中に出てきた用語の説明等	29

《 参考 》 ネット上の元ネタの目次と URL

本資料を作成する契機になった、ネット上の元ネタの目次（URL を併記）を以下に記載します。

この勉強会用の資料は、元ネタの目次の 2 項、3 項、6 項、36 項の 4 項目（項目名を太字で記載し、その右端に朱筆でこの資料での章番号を記載）について、より詳しく（?）、より易しく（?）説明しています。

【アーキテクチャ編】

1. なぜ NT 系 OS は堅牢といわれるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315586/?ST=win>
2. 壊れたファイルをなぜ復元できるのか？ 《 第 3 章で説明 》
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315571/?ST=win>
3. Windows はどうやって起動しているのか？ 《 第 1 章で説明 XP 》
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315572/?ST=win>
4. なぜ Windows は新しいハードを自動認識できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315587/?ST=win>
5. なぜ搭載メモリより大きなデータを処理できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315588/?ST=win>
6. 面倒なシャットダウン操作がなぜ必要なのか？ 《 第 2 章で説明 》

<http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315573/?ST=win>

【ネットワーク編】

7. なぜネットワーク上の他のコンピュータを一覧表示できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081010/316590/?ST=win>
8. ネットワーク上のマシンとなぜ通信できるのか？（前編）
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081010/316618/?ST=win>
9. ネットワーク上のマシンとなぜ通信できるのか？（後編）
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081016/317136/?ST=win>
10. ログオンとは、いかなるものか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081016/317082/?ST=win>
11. なぜ安全にインターネットと接続できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081016/317137/?ST=win>
12. なぜネットワーク上でファイルを共有できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081016/317083/?ST=win>

【セキュリティ編】

13. なぜ Windows はウイルスに感染するのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081016/317139/?ST=win>
14. ウイルス対策ソフトはなぜウイルスを検知できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081022/317464/?ST=win>
15. なぜネットワーク通信を盗聴できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081022/317465/?ST=win>
16. なぜ外部から Windows マシンに侵入できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081022/317467/?ST=win>
17. なぜ暗号は当事者しか読めないのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081022/317442/?ST=win>
18. なぜファイアウォールが必要なのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081022/317471/?ST=win>

【デスクトップ編】

19. なぜスタート・メニューからプログラムを起動できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081023/317575/?ST=win>
20. なぜ「ごみ箱」に入れたファイルを簡単に元に戻せるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081023/317578/?ST=win>
21. なぜファイルをダブル・クリックするとアプリケーションが起動するのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081023/317579/?ST=win>
22. なぜユーザによって別のデスクトップが表示されるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081029/318092/?ST=win>
23. なぜ呪文のような文字が表示されるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081030/318132/?ST=win>
24. なぜ違うアプリケーション間でデータをコピーできるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081030/318054/?ST=win>

【障害回復編】

25. なぜセーフ・モードで修復できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081104/318422/?ST=win>
26. なぜコマンド・ラインで復旧できるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081104/318425/?ST=win>
27. なぜ「前回の正常起動時」に戻せるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081104/318408/?ST=win>
28. なぜ「自動」で障害回復するのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081107/318743/?ST=win>
29. なぜ「システムの復元」は切り札なのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081107/318754/?ST=win>
30. なぜ壊れたマシンからデータを救えるのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081117/319430/?ST=win>
31. 障害回復の主な機能を比較，回復への道筋を考えよう
<http://itpro.nikkeibp.co.jp/article/COLUMN/20081117/319400/?ST=win>


【続・アーキテクチャ編】

32. なぜアプリがハングしても OS は大丈夫か？
<http://itpro.nikkeibp.co.jp/article/Windows/20051111/224393/?ST=win>
33. なぜ DOS アプリが Windows で動くのか？
<http://itpro.nikkeibp.co.jp/article/Windows/20051216/226357/?ST=win>
34. なぜアクセス権をどこでも使えるか？
<http://itpro.nikkeibp.co.jp/article/Windows/20060208/228778/?ST=win>
35. Windows はどのようにデバイスドライバを利用するのか？
<http://itpro.nikkeibp.co.jp/article/COLUMN/20060221/230234/?ST=win>
36. **Windows はどうやって起動しているのか？ (Vista 編)** **《第1章で説明 Vista/7》**
<http://itpro.nikkeibp.co.jp/article/COLUMN/20090113/322746/?ST=win>

出典：日経 Windows プロ

中島 省吾 (メディアプラネット社長) :

テクニカル・ライターとしてネットワークやソフトウェア関連の記事を執筆するほか、企業研修の企画・講師、ソフトウェアの開発などを手がける。



1. Windows を起動する

1.1. Windows のブート処理

パソコンの電源スイッチを押して、OS を起動するまでの処理の流れをブート処理(プロセス)という。

↓ <http://itpro.nikkeibp.co.jp/article/COLUMN/20090113/322746/?ST=win> その他

1.1.1. BIOS によるブート処理 (PC/AT 互換機共通)

- (1) 電源スイッチを入れると BIOS が起動

パソコンの電源スイッチを入れると、ROM^{ロム} (※1) に記録されている BIOS^{バイオス} (※2) が実行される。

- (2) ハードウェアの診断と初期化

電源投入直後の BIOS によるハードウェアの診断・初期化の処理を POST^{ポスト} (※3) と言う。

ROM 内の POST プログラムは、グラフィックス、メモリ、キーボード、マウス等のハードウェアの診断と初期化を行い、異常がある場合はテキスト画面またはビープ音で知らせる。

- (3) 起動デバイス (FDD、HDD、CD) の決定

BIOS の [Boot Sequence]^{ブート シーケンス} に設定されている起動デバイス (FDD、HDD、CD 等) の順序を確認し、該当するデバイスにアクセスする。

(注) 既定では、FDD (A:) ⇒ HDD (C:) ⇒ CD/DVD の順に [Boot Sequence] に設定されている。そのため FD が挿入されていない場合は、HDD が起動デバイスになる。

- (4) マスターブートレコード (MBR) の読み込み

起動デバイスが HDD の場合は、HDD の先頭にある MBR^{エムビーアール} (※4) をメモリ (0x07C00~) に読み込み、その先頭にあるプログラム (ブート・ストラップ・ローダ) に制御を渡す (ジャンプする)。

(注) MBR の読み込み時点では、ファイルシステムが無いので、直接 HDD にアクセスするので、BIOS 入出力サブルーチン^{※5} の INT(0x13)を用いて HDD を読み込む

MBR (マスターブートレコード) の構造・・・ PC/AT 互換機の一般仕様

0000(446B)	プログラム領域 (ブート・ストラップ・ローダ ^{※6})		
0446(64B)	パーティション テーブル 1	0(1B)	ブート識別子 (0x80 : On、0x00 : Off)
		1(3B)	開始位置 (CHS 方式 ^{※7})
		4(1B)	システム ID (0x07 : NTFS、0x0B : FAT32、0x0F : 拡張パーティション、0x12 : Compaq)
		5(3B)	終了位置 (CHS 方式)
		8(4B)	開始セクタ番号 (LBA 方式 ^{※8})
		12(4B)	総セクタ数 (LBA 方式)
	同 テーブル 2	同上	同上 (計 16B)
同 テーブル 3	同上	同上 (計 16B)	
同 テーブル 4	同上	同上 (計 16B)	
0510(2B)	ブートシグニチャ	0xAA55	

(参考) 1台の HDD を 5 個以上のパーティション (仕切られた領域) に分割する方法

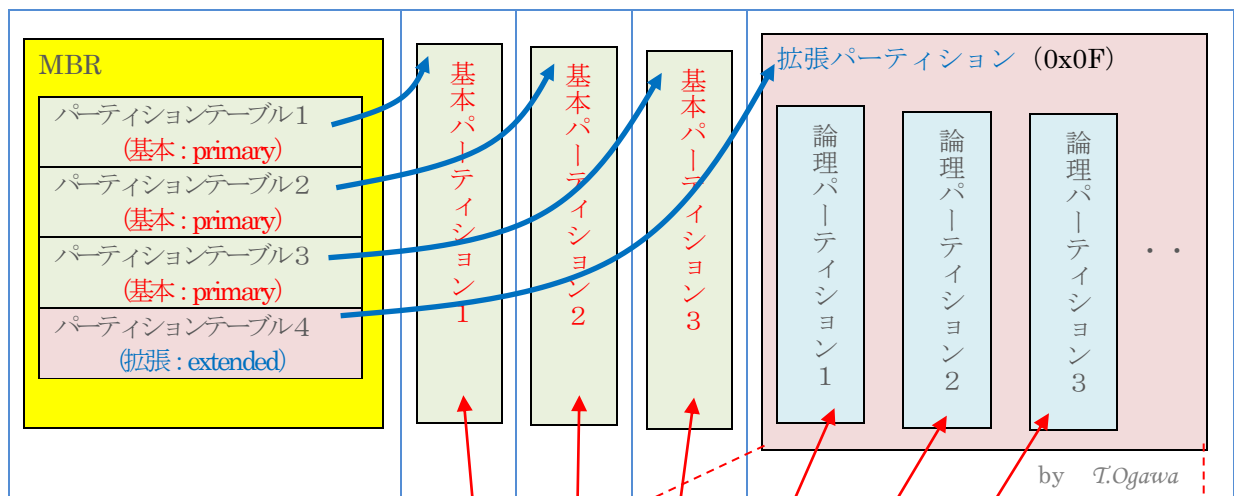
HDD の先頭にある MBR には 4 個のパーティション・テーブルがあるので、1 台の HDD は 4 個のパーティションにしか分割できないと思われがちである。

しかし、4 番目以降のパーティションを起動用に使用しないという条件付きで、次の方法で 5 個以上のパーティションに分割できる。

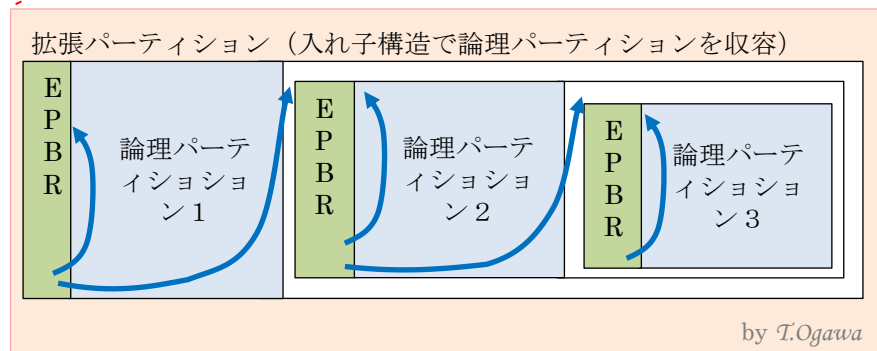
① MBR にある 4 個のパーティションテーブルのうち 3 個を基本パーティション (Primary Partition) にして、残りの 1 個を拡張パーティション (Extended Partition) にする。

② その拡張パーティションを複数 (容量が許す限り) の論理パーティション (Logical Partition) に分割することで、1 台の HDD を 5 個以上のパーティションに分割する。

詳細は、次のイメージ図を参照のこと



【ドライブレターの例】 C: D: E: F: G: H:



(補足) EPBR (Extended Partition Boot Record) : 拡張パーティション・ブートレコードは、拡張パーティションを管理するレコードであり、各論理パーティションの先頭セクタを使用する。

EPBR の構造は MBR に準じるが、先頭のプログラム領域 (ブートストラップ用の 446B) は未使用である。また、4 個のパーティションテーブルのうち 1 番目を自分 (論理パーティション) の管理用に使用し、2 番目を次の論理パーティション以降の管理用に使用し、3 番目と 4 番目は未使用である

1.1.2. ブート・ストラップ・ローダによるブート処理 (PC/AT 互換機共通)

- (1) ブート・ストラップ・ローダは自分自身を別の場所に移動した後に、処理を続行する
 (注) MBR を読み込んだ場所 (0x07C00~) に、アクティブ・パーティションの **BS** ^{ブートセクタ} (※9) を読み込むので、自分自身 (MBR) を別の場所 (0x00600~) に移動した後、移動先で処理を続ける
- (2) アクティブ・パーティションの検出
 パーティションテーブルの先頭にあるブート識別子を順番に検査して、アクティブ表示 (0x80) された起動用パーティションを探す
 (注) アクティブ表示 (0x80) が 1 個以外の場合、またはブートシグニチャが 0xAA55 以外の場合は、"Invalid partition table"として処理を停止する。
- (3) アクティブ・パーティションの先頭セクタ (BS) のアドレスの取得
 アクティブ表示されたパーティションの先頭セクタ (BS) のアドレスを取得して、BIOS の入出力サブルーチンの INT (0x13) に HDD 読み込みを依頼する
- (4) BS の読み込みと IPL (※10) へのジャンプ
 HDD の読み込み依頼を受けた BIOS は BS を 0x07C00 (ブートコード領域) に読み込んで、BS の先頭にあるプログラム (IPL) に制御を渡す (ジャンプする)。

BS (ブートセクタ) の構造 …… NTFS の例

(注) ディスクパラメータはファイルシステム (例 : NTFS、FAT32) で内容とサイズが異なる

0000(3B)	プログラム領域 1/2 (IPL へのジャンプ命令)	
0003(8B)	OEM 名	NTFS
0011(2B)	セクタサイズ	512
0013(1B)	セクタ数/クラスタ	
0014(2B)	予約セクタ数	0
0016(1B)	FAT 数	0
0017(2B)	ルートディレクトリ・エントリ数	0
0019(2B)	全セクタ数	0
0021(1B)	メディア Descriptor	0x08
0022(2B)	セクタ数/FAT	0
0024(2B)	セクタ数/トラック	
0026(2B)	ヘッド数	
0028(4B)	隠しセクタ数	
0032(4B)	全 Big セクタ数	
0036(4B)	未使用	
0040(8B)	全 NTFS セクタ数	
0048(8B)	MFT (※11) 開始クラスタ数	
0056(8B)	MFT ミラー開始クラスタ数	
0064(4B)	クラスタ数/FRS	
0068(4B)	クラスタ数/Index ブロック	

0072(4B)		シリアル番号	
0076(4B)		チェックサム	
0080(430B)	プログラム領域 2/2 (IPL)		
0510~0511	ブートシグニチャ		0xAA55

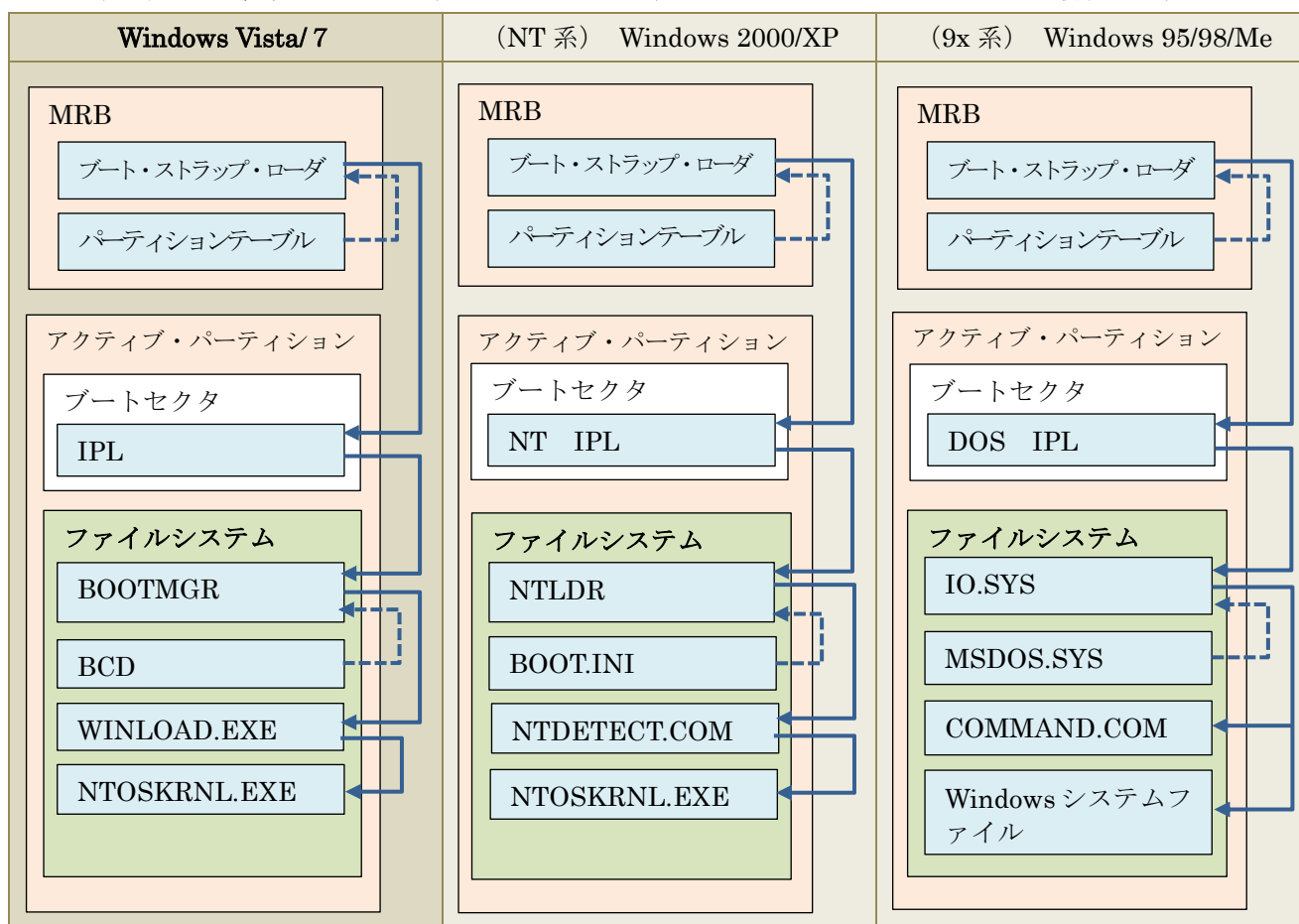
1.1.3. IPL によるブート処理 (Windows 固有)

ブート・ストラップ・ローダ (MBR 内) から制御を渡された IPL (BS 内) は、Windows 固有の手順にしたがってブート処理を行う。

(注) Windows のブート処理は、Windows 95/98/Me ⇒ Windows 2000/XP ⇒ Windows Vista/7 で大幅に変更されている。(以下、Windows Vista/7 のブート処理を説明する。)

【Windows Vista/7 のブート処理のシーケンス】

(参考までに、右 2 列に NT 系 Windows と 9x 系 Windows のブートシーケンスも併記した)



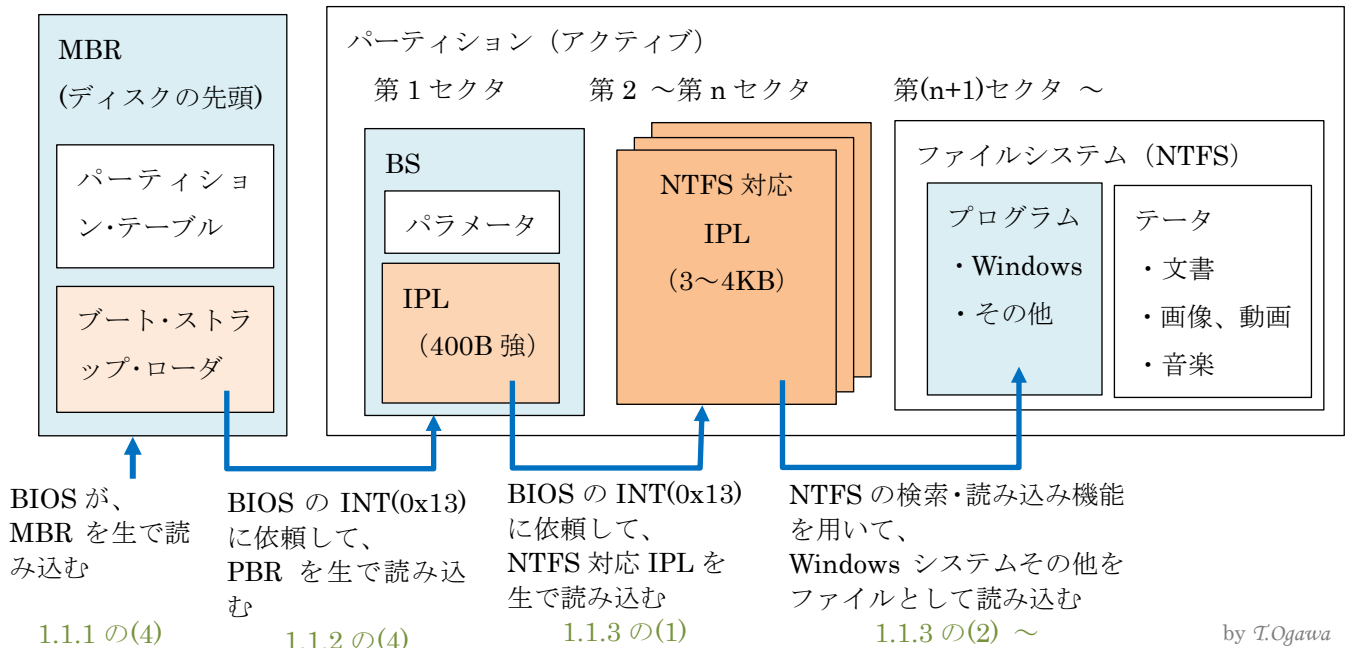
(1) IPL による NTFS 対応 IPL (仮称) の読み込み

BS 内の IPL は、BIOS の INT (0x13) を用いて、BS の後ろに続く複数セクタの NTFS 対応 IPL を読み込んでジャンプする (制御を渡す)。

(注) BS 内の IPL 用プログラム領域 (400B 強) には、ファイルシステム (NTFS) で管理されているファイルを検索し読み込むための大きな IPL を格納できない。そのため、BS 内に入る小さな IPL で、BS の後に続く複数セクタ (3~4 KB) の NTFS 対応 IPL を読み込み、その NTFS 対応 IPL でファイルシステム内のファイルを検索し読み込む方法を採用している。

(補足) BSはパーティションのフォーマット時に作成されるので、パーティションをフォーマットしたOS固有のものになる。このためIPLもOS固有のものになる。

【ブート処理の各ステージでの読み込みの流れ】



- (2) NTFS対応IPLによる^{ブートマネージャ}BOOTMNG^(※12)の読み込み
制御を渡されたNTFS対応IPLは、ファイルシステム内のBOOTMNGを検索して読み込んだ後、BOOTMNGにジャンプする(制御を渡す)。

1.1.4. BOOTMNGによるブート処理 (Windows固有)

- (1) BOOTMGRによる^{ウィンローダ}WINLOAD.EXEの読み込み
制御を渡されたBOOTMGRは、^{ブイシーディー}B C D^(※13)を参照し、次のいずれかの処理を行う
【OSが1個だけ登録されている場合】・・・**一般のパソコン**

Vista系のOSローダのWINLOD.EXEを読み込んでジャンプする(制御を渡す)

【OSが複数登録されている場合】・・・**マルチブート(上級者のパソコンのごく一部?)**

OSの切り替え画面(メニュー)を表示して、ユーザの選択を待つ

- ・ Vista系OS (Vista/7)が選択された場合は、Vista系のOSローダのWINLOD.EXEを読み込んでジャンプする(制御を渡す)
- ・ NT系OS (2000/NT)が選択された場合は、NT系のOSローダの^{エヌティイーローダ}NTLDR.EXEを読み込んでジャンプする(制御を渡す)

↑ <http://itpro.nikkeibp.co.jp/article/COLUMN/20090113/322746/?ST=win&P=4>

- (2) WINLOAD.EXEによるWindowsの立ち上げ
制御を渡されたWINLOAD.EXEは、次のような処理を行ってWindows 7を立ち上げる

↓ <http://www5a.biglobe.ne.jp/~imo/vista.html>

- ① ^{エヌティオーエスカーネル}NTOSKRNL.EXE (Windows OSの中核であるカーネル)の読み込み

- ② HAL.DLL ^(※14) (Hardware Abstraction Layer、ハードウェア抽象化層) の読み込み
- ③ 設定をレジストリ ^(※15) から読み取り
- ④ ドライバ類 ^(※16) の読み込み
- ⑤ SMSS.EXE (セッション・マネージャ・サブ・システム) ^(※17) を起動
- ⑥ WININIT.EXE (Windows スタートアップアプリケーション) ^(※18) を起動
- ⑦ LSASS.EXE (ローカルセキュリティ) ^(※19) を起動
- ⑧ SERVICES.EXE ^(※20) (サービス) を起動
- ⑨ ログイン待ち

1.2. 新技術 (Windows のブート関連)

Windows のブート関係の新技術に、「HDD の大容量化に伴う MBR から GPT への移行」と「BIOS の制約回避のための BIOS から UEFI への移行」があり、前者を実現するには後者が必要である。

1.2.1. HDD の大容量化に伴う GPT 化

パソコン用に発売された最初の HDD は 1983 年に発売された IBM PC/XT 用の外付け HDD で、シーゲート・テクノロジーの 5.25in の 10MB or 20MB (当時で 40 万円前後?) であった。その後の技術進歩と普及により、急速な大容量化と低価格化がみ、2010 年には、PC/AT 互換機の基である MBR の管理限界の 2TB ^(※21) を超えた 3TB (3.5in) の HDD が販売された。

ここで説明する G P T (G U I D Partition Table) は、8ZB ^(※22) (8TB の 10 億倍) までの超大容量の HDD を管理できる。

(1) GPT の構造・・・フリー百科事典ウィキペディア (Wikipedia) より

↓ <http://ja.wikipedia.org/wiki/GUID%E3%83%91%E3%83%BC%E3%83%86%E3%82%A3%E3%82%B7%E3%83%A7%E3%83%B3%E3%83%86%E3%83%BC%E3%83%96%E3%83%AB>

MBR				LBA0	
GPT ヘッダ (第一)				LBA1	
エントリ 1	エントリ 2	エントリ 3	エントリ 4	LBA2	} GPT エントリ (プライム)
エントリ 5	エントリ 6	エントリ 7	エントリ 8	LBA3	
				.	
				.	
エントリ 125	エントリ 126	エントリ 127	エントリ 128	LBA33	
パーティション領域 (最大 128 個のパーティション)					
エントリ 1	エントリ 2	エントリ 3	エントリ 4		} GPT エントリ (バックアップ)
エントリ 5	エントリ 6	エントリ 7	エントリ 8		
				.	
				.	
エントリ 125	エントリ 126	エントリ 127	エントリ 128		
GPT ヘッダ (バックアップ)				最終セクタ	

- ① **GPT領域**: LBA0～LBA33の34セクタ(512B/セクタ)のLBA(Logical Block Addressing)で構成される

- ・ LBA0 : MBR (MBR を前提としたディスクユーティリティ用に残している) HDD 全体を1個のパーティション (システム ID : 0xEE) としている
- ・ LBA1 : GPT ヘッダである

【GPTヘッダの概略】・・・Webで調べるも詳細な内部構造は不明

項目	項目の説明
ジューアイディ GUID	システムを識別するための16BのID
第1GPTヘッダ	自GPTヘッダのLBAとサイズ
第2GPTヘッダ	バックアップ用GPTヘッダのLBAとサイズ
属性	属性
CRC32	LBA1のCRC32(チェックサム)

- ・ LBA2～LBA33は、各セクタにGPTエン트리(128B)を4個持ち、32セクタで128個(32セクタ×4エン트리/セクタ)のパーティションを管理できる配列になっている。

【GPTエントリの概略】・・・Webで調べるも詳細な内部構造は不明

項目	サイズ	項目の説明
システムGUID	16B	システムを識別するためのGUID
パーティションGUID	16B	パーティションを識別するためのGUID
先頭LBA	8B	パーティション先頭のLBA
最終LBA	8B	パーティション最終のLBA
属性その他	80B	パーティション名、ドライブ文字、その他属性

- ② **パーティション領域**: GPT領域(①)の直後からバックアップ用GPT領域(③)の直前までが、パーティション領域であり、最大128個のパーティションを収容できる。
 なお、最大128個のパーティションの全てが基本パーティションであり、OSをインストールして起動することが可能である
- ③ **バックアップ用GPT領域**:冗長性を持たせるために、HDDの最後部にGPTエン트리(LBA2～LBA33)とGPTヘッダ(LBA1)をバックアップとして持っている。

(2) WindowsのGPTへの対応

Windowsの各バージョンのWindowsへの対応状況は、次表に示すとおりである。

【WindowsのGPTへの対応状況】

	Windowsバージョン	GPTからの起動		GPTへのアクセス		備考
		BIOS経由	UEFI経由	データ用	ブート用	
32bit	Windows Vista	×	×	○	×	
	Windows 7	×	×	○	×	
64bit	Windows XP	×	○	○	×	
	Windows Vista	×	○	○	×	
	Windows Vista SP1	×	○	○	○	
	Windows 7	×	○	○	○	

1.2.2. BIOS の制約回避のための UEFI 化

BIOS は PC/AT 互換機^(※23) の仕様のうち、27 年も経った現在も使用されている機能であり、技術進歩の激しいパソコン技術の中では珍しい技術である。

BIOS は、1MB のメモリ空間を使用するリアルモード用に 16bit コードでアセンブラを用いて作成され、拡張やメンテナンスの障害になっている。

そのためインテルとコンパック（現 HP）が協力して、2000（H12）.12 に、イーエフアイ エクシクルーシブ E F I（Extensible ファームウェア インターフェース Firmware Interface、拡張ファームウェア・インタフェース）を開発した。

この EFI の普及を図るべく UEFI フォーラムを 2005（H17）に設立し権利を移管して、ユーイーエフアイ U E F I（ユニファイド イーエフアイ Unified E F I）と名を改めた。UEFI フォーラムは 2007（H19）.1 に UEFI V2.1 を公開した。なお、最新版は、2011（H23）.7 公開の UEFI V2.3.1 である。

こうして規格化された UEFI は、メモリ空間と開発言語からの解放により、“ミニ OS” と言われるほどの機能と拡張性を持ち、ユーザ・インタフェースも CUI（文字 UI）から GUI（画像 UI）へと発展しつつある。

【重要】 27 年間も生きながらえた BIOS を急に廃止して、新しいファームウェアと置き換えるのは困難なため、現実解として Windows Vista 以降で既存の BIOS と OS との中間にソフトウェアで UEFI を実現している。
これに伴い、Vista 以降のブート処理が大きく変更された。

(1) Windows Vista は、UEFI サポートのためにブート処理を変更

↓ <http://itpro.nikkeibp.co.jp/article/COLUMN/20090113/322746/?ST=win&P=4>

Windows Vista は UEFI をサポートするために、NT 系の OS ロード（NTLDR+BOOT.INI）を捨てて、新しい OS ロード（BOOTMGR+BCD）に変更した

- ① Vista 以降では、BS の IPL（NTFS 対応 IPL）から制御を渡された BOOTMGR は、本来の OS ロード機能の他に、UEFI のファームウェアとしての機能も併せ持つ
- ② Vista 以降での UEFI は、BIOS と OS ロードの中間に位置するソフトインターフェース仕様であり、BIOS の上にある OS に EFI インタフェースを持たせて、BIOS を変更せずに UEFI ドラバなどを追加できるようにした

(2) UEFI 実装のマザーボードが出始めた

- ① 単体でも販売する Windows は、数年前に発売された旧来の BIOS 搭載の PC にも対応する必要もあり、ファームウェアの BIOS とソフトウェアの OS ロードの中間に位置する形で UEFI を実現している。
- ③ 市販 PC（カタログ仕様、カスタム仕様）では、UEFI を搭載した機種を探しえなかった。
- ④ 自作 PC 用のマザーボードとして、ASUS（例：F1A75-M LE、P8Z68-V LE、P8Z68-M PRO）や MSI（例：Z68A-GD80(G3)、A75A-G55、Z68A-G43(B3)）等が UEFI 搭載のマザーボードの販売を始めている。

2. Windows をシャットダウンする

2.1. Windows のシャットダウン

2.1.1. 通常のシャットダウン

↓ <http://itpro.nikkeibp.co.jp/article/COLUMN/20080929/315573/?ST=win> その他

シャットダウンとは、起動しているソフトウェアを安全に終了させた後、自動的にパソコンの電源を切る一連の処理をいう。

シャットダウン操作を行うと、アプリケーションの終了 ⇒ サービスの終了 ⇒ マネージャの終了の順に処理が行われてゆき、データやシステムを安全に保護した後に電源が切られる。

(1) ユーザの確認

- ① 他のユーザがログイン中の場合は、「ログイン中ユーザのデータが失われる可能性がある」ことを警告し、「はい」の応答があれば、シャットダウン処理を続行する

(2) アプリケーションの終了処理

- ② 実行中のアプリケーションにメッセージを送り、実行を終了するよう指示した後、アプリケーションからの終了報告を待つ

(補足) この終了指示メッセージは、タスクマネージャでアプリケーションを指定し「タスクの終了」をクリックしたのと同じものである

- ③ 終了指示を受けたアプリケーションは、必要な終了処理を行った後、Windows に終了報告を返す
- ④ 一定時間 (5 秒) 経過してもアプリケーションからの終了報告がない場合は、Windows の終了がブロック (一時停止の状態) される。

↓ <http://attosoft.info/blog/item-autoendtasks/#NotResponding>

(参考) Windows の終了がブロックされると、次の警告画面を表示してユーザの指示を待つ

- ◇ Vista/7 では、「・・・プログラムが閉じられていません」のブロック画面を表示して、ユーザの指示を待つ

- ・ シャットダウンを「キャンセル」してユーザ自身が対応する

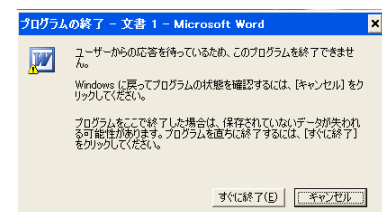
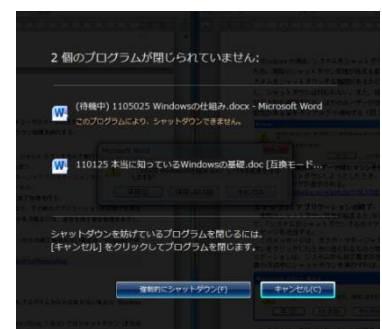
- ・ Windows が強制的にシャットダウンを進める

なお、Vista/7 では、シャットダウン処理より前に「応答なし」となっていたプログラムは自動的に強制終了させる

- ◇ NT 系では、「このプログラムは応答していません・・・」というプログラムの終了画面を表示して、ユーザの指示を待つ

- ・ シャットダウンを「キャンセル」してユーザ自身が対応する

- ・ Windows がプログラムをすぐに終了させる



- (3) サービス^(※24)の終了処理
- ① サービス・コントロール・マネージャ (SCM) にサービスの終了を指示する
 - ② サービス・コントロール・マネージャは、実行中のサービスに対してシャットダウン・コマンドを送ってサービスの終了を指示し、サービス終了の報告を待つ
 - ③ サービスから「サービス終了処理を進行中」の応答メッセージを受け取った場合は、該当サービスからのサービス終了の報告を待つ
 - ④ すべてのサービスからサービス終了の報告を受け取るか、「サービス終了処理を進行中」の応答メッセージを送ってくるサービスがなくなるまで待機する。
ただし、待機時間のタイムアウト (既定値は 20 秒) を過ぎた場合は、強制的にマネージャの終了処理に移行する
 - ⑤ すべてのサービスからサービス終了の報告を受け取った場合は、マネージャの終了処理に移行する。
- (4) マネージャの終了処理
- ① 構成マネージャ^(※25)、メモリマネージャ^(※26)、I/O マネージャ^(※27)、エグゼクティブ^(※28)、パワーマネージャ^(※29)等に終了メッセージを送り、マネージャ類を終了させる
- (5) 処理終了/電源オフ
- ① アプリケーションの終了処理 ⇒ サービスの終了処理 ⇒ マネージャの終了処理が終了した後、自動的に PC の電源を切る

【重要】最近のノート PC は電源オフでも電力を消費する機種が出てきた

次のようなユーザーニーズのため、Windows シャットダウン後も電力を消費する PC が出てきた。

- ① ノート PC 内蔵のバッテリーを充電 (80~90%) するための電力供給
- ② 各種機器への充電等のために、USB のセルフパワー (Max.2.5W) の電力供給

2.1.2. 電源ボタン長押しによる強制的な電源オフ

システムがフリーズ、シャットダウンができない、シャットダウンが終了しない、何らかの理由で緊急にシャットダウンさせたい等々のトラブルが発生した場合は、緊急時シャットダウン機能がない Vista/7 の場合は、電源ボタンを長押し (5 秒前後?) して強制的に電源を切る。

(注) XP には、「緊急時シャットダウン」と呼ばれる機能がある。これは PC がフリーズした場合などに使用するもので、次の手順で PC を即時停止できる。

- ① **Ctrl**+**Alt**+**Delete** キーで [Windows セキュリティ] 画面を表示する
- ② **Ctrl** キーを押しながら [シャットダウン] をクリック、[緊急時シャットダウン] 警告メッセージを表示する
- ③ [OK] をクリックしてパソコンを停止する

↑ <http://hehao1.seesaa.net/article/21499150.html>

↑ <http://www.microsoft.com/japan/athome/magazine/ucontents/users/tips/windows/145.mspx>

2.2. Windows の省電力機能

多くのパソコンユーザは、使用時に電源を入れて Windows を起動し、使用終了時にシャットダウン操作で電源を切り、「起動に時間がかかる」、「シャットダウンに時間がかかる」と嘆いている。

パソコンには、1 W 未満のスリープ (スタンバイ) や電力皆無の休止状態と呼ばれる省電力機能が

備わっている。この機能を使うと前者のスリープは数秒で中断／再開ができ、後者の休止状態は 10～20 秒程度で中断／再開ができるようになっている。

筆者は、特別の場合を除きスリープ機能を使っているが、未だトラブルに見舞われたことはない。

(参考) パソコンの取説にスリープ時の消費電力が記載されていて、1 W と書かれた PC が多いようである。近年のノート PC の場合は 1 W と書かれていてもネット上に掲載された実験データによると 0.5W 前後のようである。

仮に、スリープ時の消費電力を多めの 1 W とした場合の電気料金は、**1 年間で 210 円程度 (365 日×24 時間×0.024 円／1 時間=210 円…契約容量と使用量で異なる)** である。

(参考) パソコンの省電力に関する規格について

1997 年にインテル、マイクロソフト、東芝が策定した ACPI (アドヴァンスト コンフィグレーション アンド パワー インターフェース Advanced Configuration and Power Interface) は、PC/AT 互換機等の電源と構成要素に関する規格である。これは Windows 98 以降で採用されている。2000 年にはコンパック (現 HP)、フェニックステクノロジーズも加わり ACPA 2.0 規格となった。

更に 2006 年に ACPI 3.0b が、2010 年に 4.0a が公開されている。

【ACPI で定義したシステムスリープの状態】

	システムスリープの状態 (ACPI 3.0 b 規格)
S0	通常の状態
S1	スタンバイとよばれるメモリ、デバイス、レジスタコンテキストおよびキャッシュコンテキストを CPU が保持したまま割り込み等を止め、低消費電力状態に移行する状態である。タイマの復帰などを電源管理イベントとして扱い処理する程度で矛盾無く復帰可能である
S2	キャッシュコンテキストが失われる以外は S1 と同様である。
S3	Suspend to RAM または <u>スリープ</u> と呼ばれる状態で、メモリは保持されているが、レジスタコンテキストが失われる。サスペンドに入る前に OS はレジスタコンテキストをメモリに書き出し、16 ビットコードの復帰ベクタを FACS の然るべき場所に書いておく。復帰はリセット状態から復帰し、BIOS がサスペンド状態であったことを検知して初期化を行った後復帰ベクタへ移行する。その後復帰ベクタから <u>プロテクトモード</u> への復帰などを行って最終的にレジスタを書き戻して運用状態に復帰する。 【筆者注】 XP の「サスペンド」、Vista/7 の「スリープ」に相当 なお Vista/7 の「ハイブリッドスリープ」は、S3 と S4 の組み合わせ
S4	Suspend to Disk または <u>ハイバネーション</u> と呼ばれる状態で、メモリ内容も失われる。メモリを HDD に書き出し電源断の状態にするのと同じである。初期には書き出し復帰は BIOS が行う事もあり、その場合の処理は S1 や S3 等と同等だったが、OS が行う場合には復帰時はブートローダや OS が、ハイバネーション内容が存在することを検出してメモリ内容を書き戻すことになる。 【筆者注】 XP/Vista/7 の「休止状態」に相当
S5	単なる電源断である 【筆者注】 XP/Vista/7 の「シャットダウン」に相当

↑ [http://ja.wikipedia.org/wiki/Advanced Configuration and Power Interface](http://ja.wikipedia.org/wiki/Advanced_Configuration_and_Power_Interface)

2.2.1. Windows 7 の省電力機能

Windows 7には、「スリープ」、「休止状態」、「ハイブリッドスリープ」の3つの省電力機能がある。

(注) 「ハイブリッドスリープ」を使用できるように設定すると「休止状態」は使用できない。

そのため「休止状態」を使用する場合は、「ハイブリッドスリープ」を使用しないように設定する必要がある。

(参考) MSの「Windows 7 ヘルプ」によると、「休止状態は、主としてノート PC 用に設計された省電力の状態」であり、「ハイブリッドスリープは、主としてデスクトップ PC 用に設計されている」とのことである。

 <http://windows.microsoft.com/ja-JP/windows7/Sleep-and-hibernation-frequently-asked-questions>)

(1) スリープ

- ▶ スリープは、パソコンの作業状況（データ、プログラム、設定等）をメモリに保持するために、わずかな電力（1W/H 弱⇨ナツメ電球の数分の1程度）を使用する省電力機能である。
- ▶ スリープでは、高速なメモリに作業状況を保持しているので素早く作業を再開できるが、電源トラブル（停電、コンセント抜け）等によりメモリに保持している作業状況が消えることがある。

(注) スリープに移行して 18 時間（変更可能）が経過すると、自動的に休止状態に移行する。

またノート PC では、バッテリーの充電量が極度に少なくなると、自動的に休止状態に移行する

(補足) スリープでは、メモリのリフレッシュ（データ保持のための一定間隔で再書き込み）に、わずかな電力（1W/H、ナツメ電球の数分の1程度）を消費する。

しかし上記の注書きに示したように、長時間のスリープでは自動的に休止状態に移行するので、スリープだけ使用した場合の電力料金は、月 10 円前後と推察される。

【スリープへの移行手順】

- ① [スタート] の Windows 終了の [スリープ] をクリックして、スリープに移行する

【作業の再開手順】

- ① 電源スイッチを押すと、数秒でスリープ直前の状態に戻って作業を再開できる

(注) 電源スイッチを押してスリープを解除するパソコンが多いが、詳細は取説等で確認すること。

(2) 休止状態

- ▶ 休止状態は、パソコンの作業状態（データ、プログラム、設定等）を HDD に保存した後、電源を切る省電力機能である。
- ▶ 休止状態では、低速の HDD に保存している作業状態をメモリに読み込み作業を再開するので、作業再開までにやや時間を要するが、万一電源トラブル（停電、コンセント抜け）等があっても保存している作業状況は保たれる。


【休止状態への移行手順】

- ① [スタート] の Windows 終了の [休止状態] をクリックして、休止状態に移行する

(注) Windows 終了に [休止状態] が表示されない場合は「ハイブリッドスリープ」を使用する設定になっている。この場合は、次の手順で「ハイブリッドスリープ」を使用し

ない設定にすると、「休止状態」を表示するようになる。

【手順】

- ① [スタート] → [コントロールパネル] → [システムとセキュリティ] → [電源オプション] とクリックし、[電源オプション (電源プランの選択)] 画面を表示
- ② ラジオボタンがオンの電源プラン (例: 高パフォーマンス) の [プラン設定の変更] をクリックし、[プラン設定の編集] 画面を表示
- ③ [詳細な電源設定の変更] をクリックし、[電源オプション] 画面を表示
- ④ [スリープ] の田をクリックし、[スリープ] を展開表示
- ⑤ [ハイブリッドスリープを許可する] の田をクリックし、[ハイブリッドスリープを許可する] を展開表示
- ⑥ [バッテリー駆動] → [オン ▼] の▼を順にクリックし、メニューを表示
- ⑦ 「オフ」をクリックし、[バッテリー駆動] を「オフ」に設定
- ⑧ [電源接続] → [オン ▼] の▼を順にクリックし、メニューを表示
- ⑨ 「オフ」をクリックし、[電源接続] を「オフ」に設定
- ⑩ [適用] → [OK] を順にクリックした後、 をクリックし [コントロールパネル] を閉じる

【作業の再開手順】

- ① 電源スイッチを押すと、10～30 秒程度 (パソコンで異なる) で直前の状態に戻って作業を再開できる

(注) 電源スイッチを押して休止状態を解除するパソコンが多いが、詳細は取説等で確認すること。

(3) ハイブリッドスリープ

- ▶ ハイブリッドスリープは、作業状態 (データ、プログラム、設定等) をメモリと HDD に保存する省電力機能であり、作業再開が早い「スリープ」と、作業再開がやや遅いが電源トラブルに強い「休止状態」の長所を併せ持っている。このためハイブリッドスリープは、スリープと同様にわずかな電力 (1W/H 弱≒ナツメ電球の数分の1程度) を使用する。
- ▶ ハイブリッドスリープでは、スリープと同様に高速なメモリに作業状況を保持しているので素早く作業を再開できる。

万一電源トラブル (停電、コンセント抜け) 等があっても、やや時間を要するものの HDD に保存している作業状態を用いて作業を再開できる。

【ハイブリッドスリープへの移行手順】

- ① [スタート] の Windows 終了の [スリープ] をクリックして、スリープに移行する

【重要】 [コントロールパネルの] の [電源オプション] の [スリープ] で「ハイブリッドスリープを使用する」ように設定している場合は、Windows 終了の「スリープ」は、「ハイブリッドスリープ」を意味している。

(注) Windows 終了に [休止状態] が表示される場合は「ハイブリッドスリープ」を使用しない設定になっている。その場合は、[休止状態] の表示のための設定変更の手順を参考にして、「ハイブリッドスリープ」を使用する設定にすると、「休止状態」を表示しないようになる。

【作業の再開手順】

- ① 電源スイッチを押すと、数秒後に直前の状態に戻って作業を再開できる。

万一電源トラブル（停電、コンセント抜け）等があった場合でも、10～30秒程度（パソコンで異なる）でハイブリッドスリープ直前の状態に戻って作業を再開できる

（注）電源スイッチを押して休止状態を解除するパソコンが多いが、詳細は取説等で確認すること。

2.2.2. Windows Vista の省電力機能

Windows Vista には、「スリープ」、「休止状態」、「ハイブリッドスリープ」の3つの省電力機能がある。

なお、これらの省電力機能は、2.2.1. [Windows 7 の省電力機能] と同じである。

（補足）Windows Vista の [コントロールパネル] の [システムとメンテナンス] が Windows 7 では [システムとセキュリティ] に変更されている

（参考）「Windows Vista では、シャットダウンよりもスリープを使ってほしい。Vista ロゴのあるマシンなら、スリープ状態から2秒で復帰できる」-----マイクロソフト Windows 本部の中里氏は Windows Vista に搭載されるノート PC 向け機能のプレス向け説明会（2006年9月4日）でこう強調した。

[↑ http://itpro.nikkeibp.co.jp/article/NEWS/20060904/247116/](http://itpro.nikkeibp.co.jp/article/NEWS/20060904/247116/)

（1）スリープ

Windows 7 と同様なため、説明を省略する。

（2）休止状態

Windows 7 と同様なため、説明を省略する。

（3）ハイブリッドスリープ

Windows 7 と同様なため、説明を省略する。

2.2.3. Windows XP の省電力機能

Windows XP には、「スタンバイ」、「休止状態」の2つの省電力機能がある。

（1）スタンバイ

- スタンバイは、パソコンの作業状況（データ、プログラム、設定等）をメモリに保持するために、わずかな電力（1W/H 弱≒ナツメ電球の数分の1程度）を使用する省電力機能である。
- スタンバイでは、高速なメモリに作業状況を保持しているので素早く作業を再開できるが、電源トラブル（停電、コンセント抜け）等によりメモリに保持している作業状況が消えることがある。

【スタンバイへの移行手順】

- ① [スタート] の [終了オプション] の [スタンバイ] をクリックして、スタンバイに移行する

(注意事項) 事前に、通信、印刷、動画や音楽の再生は終了させておくこと

【作業の再開手順】

- ① 電源スイッチを押すと、数秒後に直前の状態に戻って作業を再開できる

(注) 電源スイッチを押してスリープを解除するパソコンが多いが、詳細は取説等で確認すること。

(2) 休止状態

- 休止状態は、パソコンの作業状態（データ、プログラム、設定等）を HDD に保存した後、電源を切る省電力機能である。
- 休止状態では、低速の HDD に保存している作業状態をメモリに読み込み作業を再開するので、作業再開までにやや時間を要するが、万一電源トラブル（停電、コンセント抜け）等があっても保存している作業状況は保たれる。

【休止状態への移行手順】

- ① [スタート] の [終了オプション] で **Shift** キーを押して、[スタンバイ] を [休止状態] に変化させる

(注意事項) 事前に、通信、印刷、動画や音楽の再生は終了させておくこと。

Word、Excel その他のアプリで作成・編集中のファイルは上書き保存しておくこと。

(注) [電源オプション] で **Shift** キーを押しても [スタンバイ] が [休止状態] に変化しない場合は、次の手順で [休止状態] に変化するように設定変更する。

【手順】

- ① [スタート] → [コントロールパネル] → [パフォーマンスとメンテナンス] → [電源オプション] とクリックし、[電源オプションのプロパティ] 画面を表示
 - ② [休止状態] タブで [休止状態] パネルに切り替える
 - ③ [詳細な電源設定の変更] をクリックし、[電源オプション] 画面を表示
 - ④ [休止状態を有効にする] をクリックし、チェック (✓) を入れる
 - ⑤ [適用] → [OK] とクリックした後、**×** をクリックし [コントロールパネル] を閉じる
- ② **Shift** キーを押したまま [休止状態] をクリックして、休止状態に移行する

【作業の再開手順】

- ① 電源スイッチを押すと、10～30 秒程度 (パソコンで異なる) で直前の状態に戻って作業を再開できる

(注) 電源スイッチを押して休止状態を解除するパソコンが多いが、詳細は取説等で確認すること。

3. NTFS はクラッシュ直前の状態にファイルを復元できる

FAT32 までのファイルシステムは HDD のファイル空間を管理するためのものであり、ファイルの安全性のための機能はない。

3.1. FAT32 と NTFS の比較

↓ http://ja.wikipedia.org/wiki/NT_File_System NTFS

↓ http://ja.wikipedia.org/wiki/File_Allocation_Table2 FAT3

Windows で使用できるファイルシステムには FAT32^(※30) と NTFS^(※31) があるが、現在では殆どのユーザが NTFS を使用している。

「FAT32 は USB メモリや小容量の HDD 用のファイルシステムで、しかもファイルサイズに限界があるが、NTFS は大容量の HDD 用のファイルシステムである。」と知っているユーザが多い。しかし、次の比較表で明らかなように FAT32 は単にデータファイルを保存する機能だけの幼稚ともいえるファイルシステムである。一方の NTFS はデータファイルを保護するための様々な機能を備えているので、FAT32 に比べて安心してデータを保存しておくことができるファイルシステムである。

【FAT32、NTFS の仕様】

	FAT32	NTFS
ファイルシステム導入	1996/08 (Windows 95 OSR2)	1993/07 (Windows NT 3.1)
最新バージョン		NTFS 3.1 (XP 以降で使用)
主な対応 Windows	95、98、2000	2000、XP、Vista、7
システム ID (MBR)	0x0B、0x0C	0x07
GUID (GPT) (2.2.1 項参照)	EBD0A0A2-B9E5-4433-87C0 -68B6B72699C7	EBD0A0A2-B9E5-4433-87C0 -68B6B72699C7
最大ファイルサイズ	^{ギガ} 4GB	^{テラ} 16TB (論理上 ^{エクサ} 16EB)
最大ファイル数	1G 個	4G 個
最大ボリュームサイズ	^{テラ} 2TB	^{テラ} 256TB (論理上 ^{エクサ} 16EB クラスタ)
クラスタサイズ (ボリューム)	512B(64MB)~16KB(32GB)	4KB(512MB)~64KB(^{テラ} 256TB)
ファイル名長	8.3 形式 ^(※32) or 256 文字	256 文字
ファイル名の文字	Unicode	Unicode
①POSIX	×	POSIX.1 仕様
②セキュリティの向上	△ (読み取り専用のみ)	パーミッション (ACL)
③ディスククォータ	×	○
④透過的なデータ圧縮	×	LZ77 アルゴリズム
⑤透過的なデータ暗号化	×	AES (XP SP1 以降)
⑥堅牢性の向上	×	ジャーナリング・ファイル・システム
⑦耐障害性	×	○
⑧スナップショット	×	○
⑨変更ジャーナル	×	○

3.2. NTFS にはサーバにも使える高度な機能が多い

3.1 節の表の前半は、主にファイルシステムの規模を示すものであり、表の後半は NTFS で追加された高度な機能である。

(1) POSIX

ポジックス POSIX^(※33)への対応は、米国政府にコンピュータを納入するための必須条件である。MSはWindowsを政府機関に納入するために、Windows NT系サーバ用のNTFSをPOSIXに対応させた。

(2) パーミッション (Permission : 許可)

パーミッション^(※34) (Permission : 許可)とは、HDD等に保存したフォルダやファイルに対するユーザのアクセス権を言い、これには読み込み、書き込み、実行等がある。

(3) ディスククォータ

クォータ^(※35) (Quota : 分け前、割当額)とは、HDDを複数ユーザが使用する場合に、ユーザ毎にディスク使用量の上限を割り当てることを言う。

(4) 透過的なデータ圧縮

圧縮に関する知識が無くても、簡単な操作で圧縮できる方法を透過的な圧縮^(※36)と言う。

NTFSでは、LZ77アルゴリズムと呼ばれる「辞書型符号化」方式の透過的なデータ圧縮を使用して、ドライブ、フォルダ、ファイルを圧縮することができる。

圧縮されたフォルダやファイルは、圧縮の有無を意識することなくそのまま使用できる。

(参考) XP以降では、NTFSの持つ透過的な圧縮機能の他に「圧縮 (Zip形式) フォルダ」という機能を持っている。この圧縮 (Zip形式) フォルダはNTFS以外のファイルシステムに圧縮したままコピーでき、しかもNTFSの圧縮より圧縮率が高いといわれている。

(5) 透過的なデータ暗号化

暗号化に関する知識が無くても、簡単な操作で暗号化できる方法を透過的な暗号化^(※37)と言う。

Windows XP SP1以降のNTFSでは、エーイーエス アドバンスド エンクリプション スタンダード AES (Advanced Encryption Standard) 方式の透過的なデータ暗号化を使用して、フォルダ、ファイルを暗号化することができる。

暗号化されたフォルダ、ファイルは、暗号化の有無を意識することなくそのまま使用できる。

(6) 堅牢性の向上 (ロールバック)

クラッシュ等が発生した場合、トランザクション・ログを用い、実行した処理をロールバック^(※38)してファイルシステムの不整合を防ぐ、ジャーナリング・ファイルシステムを取り入れている。

(7) 耐障害性 (代替クラスタ)

障害性は一般的には可用性 (アベイラビリティ Availability) と呼ばれていて、コンピュータの一部に障害が発生した場合でも、自動修正、予備・代替切り替え等の機能や手段によって、故障の影響を最小限に止めて継続して運用できる能力を言う。

NTFSでは、PCで最も障害が起きやすいといわれるHDDに対して、耐障害性向上のための代替クラスタ機能をサポートしている。

これは、不良セクタが発生した場合、不良セクタを含むクラスタ (通常は、4KB/セクタ) を不良クラスタとしてMTF (Master File Table) の\$BadClusファイルに記録すると共に、そのクラスタ内のデータを代替クラスタ (代替用の予備クラスタ) にコピーする。以降そのクラスタに対するアクセスは代替クラスタに対して行われるようになる。

(補足) 不良クラスタの内容を代替クラスタにコピーする際には、不良セクタ内のデータは回復され

ない。これの対策には、RAID1 (2 台の HDD でミラーリング)、RAID5 (3 台以上の HDD で RAID) 等の冗長構成にしておく必要がある。

(8) スナップショット (VSS)

スナップショットは、VSS (Volume Shadow Copy Service) ボリューム シャドウ コピー サービス と言われていて、ある時点のボリュームの状態を正確にバックアップする機能である。Windows に付属するバックアップユーティリティはこの VSS 機能を利用している。Vista 以降では VSS によってファイルの世代管理を実現している。

(9) 変更ジャーナル

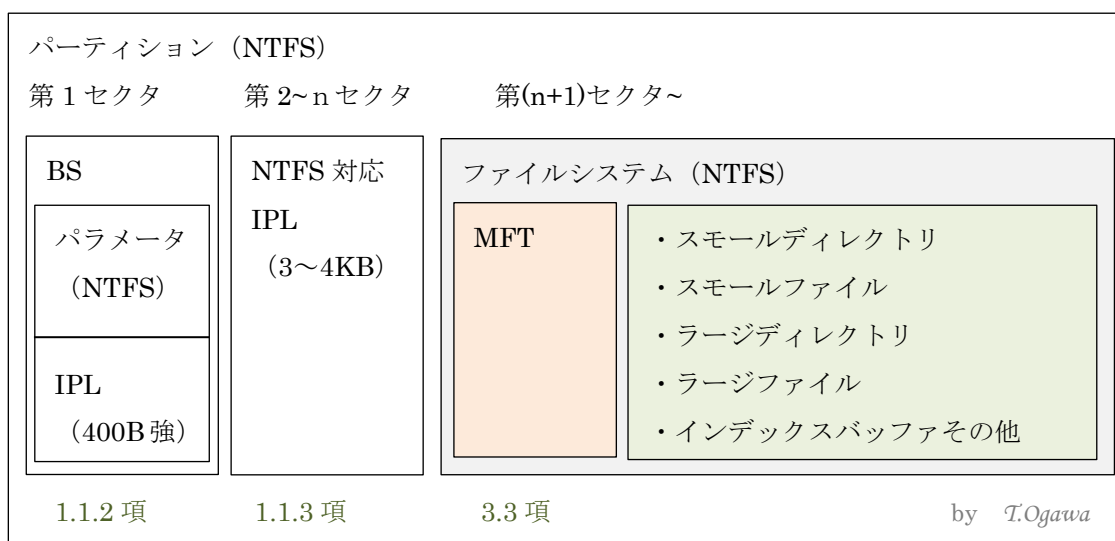
ファイルの変更 (追加、削除、変更) をログ (変更ジャーナル) として記録して、ファイルの変更を追跡できるようにした機能。アプリは API を利用して簡単にファイルの変更履歴を知ることができる。例えば、バックアッププログラムは、増分バックアップの前に、変更ジャーナルを利用して、バックアップすべきファイルの一覧を作成できる。

3.3. NTFS の主役は MFT である

Microsoft が NTFS の仕様を明らかにしていないため、NTFS の構造はネット上の記事を基に推定する他にないが、それらの記事に共通する内容を総合すると以下に示す構造と想定される。

(1) パーティション (NTFS) の構造

↓ <http://www.exceedone.co.jp/tabid/112/Default.aspx>



(2) MTF ファイルの構造

MFT (Master File Table) マスター ファイル テーブル ファイルは、ファイルシステム (NTFS) を管理するための情報を格納しているファイルであり、ブートセクタ (BS) のディスクパラメータからリンクされている。

↓ <http://support.microsoft.com/kb/174619/ja>

- ・ MFT のエントリ (レコード) は、1KB (1,024B) の固定長
- ・ 先頭の 16 レコード (0~15 レコード) は、ファイルシステムを管理するメタデータ
- ・ 続く 8 レコード (16~23 レコード) は、将来の拡張に備えた予約領域
- ・ その後 (24 レコード~) に、NTFS ボリュームの 12.5% を占める MFT ファイル専用の予約領域 (※39) が続く。

この MFT ファイル専用の予備領域は、ユーザのファイルやディレクトリ用には使用しない。ただし、この後に続くユーザ領域をすべて使い切った場合に限り使用できる。

(参考) 次表に示した MFT ファイルの各レコードの役割その他は、主に次のページを基にして説明した。

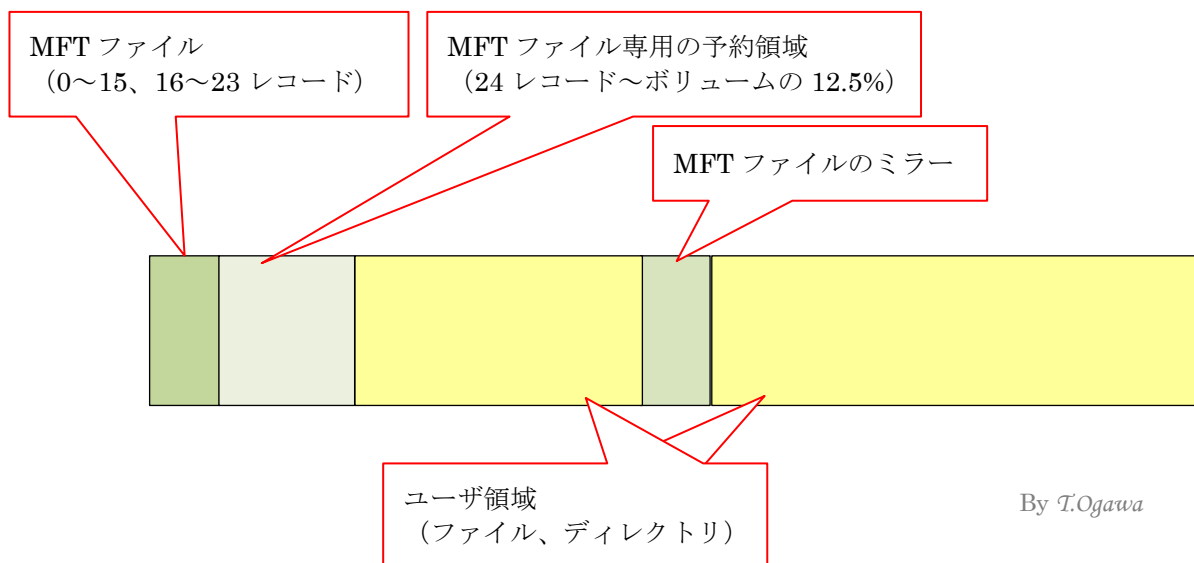
黒字：Microsoft のサポートサイト (<http://support.microsoft.com/kb/103657/en-us?fr=1> より、英語です)

青字：EaseUS (パーティション・マネージャその他のソフト)

(<http://www.easeus.com/data-recovery-ebook/ntfs-file-system-metafiles.htm> より、英語です)

レコード	ファイル名	役割その他
0	\$MFT	<small>マ ス タ ファイル テーブル</small> Master File Table (MFT) : NTFS ボリューム上のすべてのファイルレコードのデータ属性を含む
1	\$MFTMirr	<small>マ ス タ ファイル テーブル</small> Master File Table2 (MFT2) : 回復の目的で使用する MFT のミラーファイル
2	\$LogFile	<small>ロ グ ファイル</small> Log File : 回復の目的で使用する
3	\$Volume	<small>ボ リ ュ ム</small> Volume : ボリューム名やバージョン等のボリューム情報を含む
4	\$AttrDef	<small>ア ト リ ビ ュ ッ ト デ ィ フ ィ ニ ッ シ ョ ン ズ テ ー ブ ル</small> Attribute Definitions Table : すべてのシステム定義とユーザ定義の属性を含む
5	\$Root	<small>ル ー ト ファイル ネ ー ム イ ン デ ッ ク ス</small> Root File Name Index : 全てのファイルとディレクトリのインデックスをルート・ディレクトリに保存
6	\$Bitmap	<small>ク ラ ス タ ア ロ ケ ー シ ョ ン ビ ッ ト マ ッ プ</small> Cluster Allocation Bitmap : 使用中クラスタを表すボリューム全体のビットマップを含む
7	\$Boot	<small>ブ ー ト ファイル</small> Boot File : 起動可能ボリュームの場合ブートストラップを含む
8	\$BadClus	<small>バ ッ ド ク ラ ス タ ファイル</small> Bad Cluster File : ボリュームのすべての不良クラスタを含む
9	\$Secure	<small>セ キ ュ ア ファイル</small> Secure file : ボリューム全体のセキュリティ記述子のデータベースを保存している。 NTFS ファイルとディレクトリはそれぞれのセキュリティ記述子を持っている。
10	\$UpCase	<small>ア ッ プ ケ ー ス テ ー ブ ル</small> Upcase Table : 小文字 Unicode を大文字 Unicode に変換するために使用する表
11	\$Extend	<small>エ ク ス テ ン デ ィ ッ ド メ タ デ ー タ デ ィ レ ク ト リ</small> Extended Metadata directory : 拡張メタデータディレクトリ
12	\$Reparse	<small>リ パ ー ス ポ イ ン ト ファイル</small> Reparse Points File : 再解析ポイント
13	\$UsnJrnl	<small>ロ グ キ ャ ン キ ン グ ファイル</small> Log Changing File : 変更ファイルのログ (ジャーナル)
14	\$Quota	<small>ク オ ー タ テ ー ブ ル</small> Quota Table : 各ユーザのディスククォータ使用状況を表示するための表
15	\$ObjID	<small>オ ブ ジ ェ ク ト アイディ ファイル</small> Object I D File : オブジェクト ID を登録
16~23		<small>リ ザ ー ブ</small> Reserve : 将来の拡張用の予約
24~		MFT ファイル専用の予約領域 (NTFS ボリュームの 12.5%まで)

(3) MFT の領域の割り当て



3.4. クラッシュ直前までロールバックしてファイルを復元

3.4.1. クラッシュによるファイル破壊

ファイル更新では、ディスクへの書き込みが1回だけと思われがちだが、実際は実データの書き込みの他に、アロケーション情報、属性情報その他の管理情報の更新のための書き込みが必要であり、複数の書き込みが行われている。

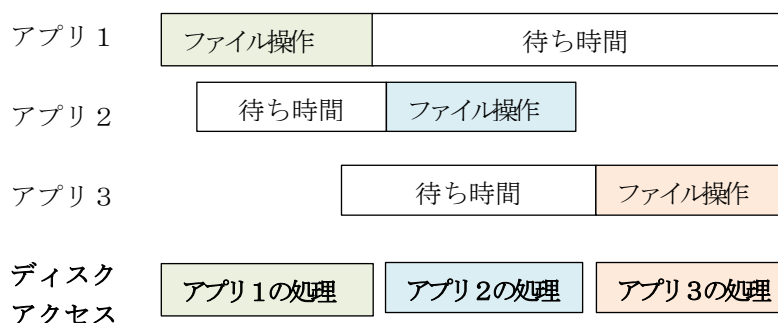
このため、クラッシュ^(※40)が発生すると1回の書き込みに見えたディスクへの書き込みのうち、一部だけが書き込まれ、残りは書き込まれず、管理情報と実データに不整合が生じる。あるいは書き込み途中のデータの後半部分が書き込まれず、途中まで書き込まれたデータがごみになることがある。

3.4.2. ファイル更新を効率的かつ安全に行うための手法

XP、Vista、7等で採用されているプリエンプティブ・マルチタスク^(※41)では、複数のアプリケーションが時分割で並行処理を行っている。このため複数のアプリケーションから複数のファイル更新が発生して、ディスク書き込みの競合が起こる。

NTFSでは、競合するファイル更新を安全に効率よく行うために、ケアフル・ライト^(※42)、遅延書き込み^(※43)、トランザクション処理^(※44)等の手法を採用している。

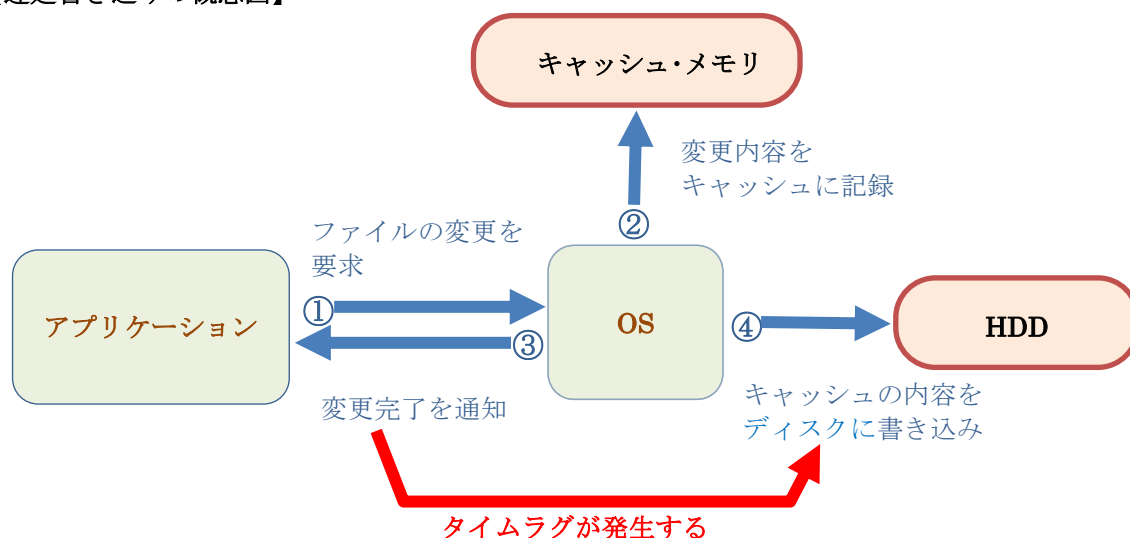
【ケアフルライトの概念図】



アプリケーションからのファイル更新を1個ずつ順番にディスクに書き込んで安全性を高める。

欠点はこのためアプリケーションにディスクアクセスの待ち時間が生じること。

【遅延書き込みの概念図】



アプリケーションからのファイル更新をメモリ上の遅延書き込みキャッシュに記録しておき、ディスクが空いた時点でディスクに書き込むことでケアフル・ライトに伴う待ち時間の問題を低減する。欠点はキャッシュへの記録とディスクへの書き込みにタイムラグが生じること。

3.4.3. NTFS にはファイルを復元できるトランザクション機能がある

ビジネス用に開発された Windows NT のファイルシステム (NTFS) は、トランザクション機能を持っている。Windows NT (NT3.1~NT4.0) の後継でもあり、ビジネス用 OS と消費者用 OS を一本化した Windows XP (NT5.1) は、Windows NT の技術の延長線上にあり、プリエンプティブ・マルチタスクや NTFS を採用している。

NTFS では、更新レコードをロギングデータとして自動的に記録しているので、クラッシュ後の起動時には、ロギングデータを用いて自動的にクラッシュ直前の状態までロールバックし、不整合のない状態にファイルを復元することができる。

一方、古いファイルシステムの FAT32 ではトランザクション機能がなくロギングデータがないので、書き込み途中の不完全なデータが残ったり、管理情報と実データに不整合が生じたりすることがある。この不整合が原因で、最悪の場合はファイルシステム全体にアクセスできなくなる危険性さえある。

3.4.4. ファイル復元のためにログデータを記録とチェックポイント処理

アプリケーションからファイル更新^(※45)の要求を受け取った NTFS は、更新レコードを作成して、ログファイル^(※46)に記録 (保存) する。

また定期的にチェックポイント処理を行い、ログレコードの状況等を編集してチェックポイント情報として記録する。

【ログ・ファイルのイメージ図】

リスタート・エリア	ロギング・エリア					
	n	n+1	n+2	n+3	n+4	...
回復処理用	レコード	レコード	レコード	レコード	レコード	...

チェックポイント
レコードの
論理シーケンス番
号

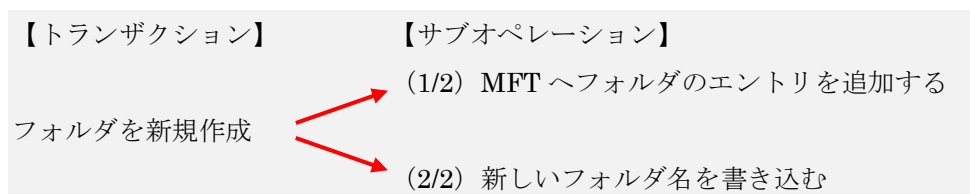
- ・トランザクションの手順を [更新レコード] として記録する
更新レコードは、[リドゥ情報] と [アンドゥ情報] を持ち、論理シーケンス番号 (通し番号) が付けられる
- ・定期的に、チェックポイントデータを記録する
チェックポイントデータは次の 3 種類のレコードを 1 組として記録され、各レコードには論理シーケンスが付けられる
 - ① ダーティページ・テーブル、
 - ② トランザクション・テーブル、
 - ③ チェックポイントレコード

(1) ファイル更新要求を元に作った更新レコードを記録 (保存) する

アプリケーションからファイル更新を要求された NTFS は、次の手順により更新レコードをロギング・エリアに記録した後、ファイル更新の完了をアプリケーションに通知する。

- ① NTFS は、アプリケーションから要求されたファイル更新処理 (トランザクション) をディスク書き込み単位 (サブオペレーション) に分割し、それぞれを書き込みデータ情報として遅延書き込みキャッシュ (メモリ) に記録する

(参考) トランザクションをサブオペレーションに分割する例



- ② サブオペレーションの「リドゥ情報」と「アンドゥ情報」を編集して、論理シーケンス番号を付与した更新レコードをロギング・エリアに追加記録 (保存) する

(補足) 「リドゥ情報」はディスクへ書き込むデータ情報であり、クラッシュ等でトランザクションが中断した場合に、ディスクへの変更を再実行する時に使用する。

「アンドゥ情報」はディスクから読み込んだデータ情報であり、トランザクションをロールバックする場合に、ディスクへの変更を取り消すために使用する

- ③ キャッシュに記録された書き込みデータ情報をキューに接続して、HDD への書き込みを待つ

(補足) キューとは、ファースト イン ファースト アウト FIFO (First In First Out、先入れ先出し) 方式で処理要求等のデータを接続する待ち行列を言い、キューに接続されたデータを古いものから順に取り出して処理を行う。逆に新しく接続したデータから順に取り出す方式をスタックと言う。

- ④ トランザクションの全ての書き込みデータ情報をキューに接続し終わったら、アプリケーションに更新完了を通知する。

(2) トランザクションが終了したらコミットする

トランザクションのサブオペレーションが全て正常に完了したら、その結果を確定するコミットを行う。

- ① NTFS は、書き込みキューに接続された書き込みデータ情報を先頭から順次取り出して、ディスクへの書き込みを行う
- ② トランザクションの全てのサブオペレーションが正常に完了(キャッシュからディスクへの書き込みが完了)した場合は、該当トランザクションを終了させコミット^(※47)する
- ③ コミットされたトランザクションの更新レコードが、ログ・エリアから削除されるので、ロギング・エリアの空きが増える。また同時に、更新レコードに対応するキャッシュ上の書き込みデータ情報も削除される。

(3) ディスクへの書き込みが追いつかない場合のエラー処理

キャッシュからディスクへの書き出しに時間を要するので、ファイル更新が頻発するとロギング・エリアに更新レコードを追加する余裕がなくなり、エリア不足のエラーになることがある。

このような場合は、次のエラー処理を行う。

- ① ロギング・エリアに余裕がなく、トランザクションの更新レコードが追加できなくなった場合は、「ログファイル・フル」エラーを返した後、新たなトランザクションの受付を一時停止する
- ② 「ログファイル・フル」エラーになったトランザクションをロールバックして、トランザクション開始前の状態に戻す。
なお、このトランザクションは措置終了後に再実行するので、書き込みデータ情報をキューに残しておく
- ③ キューに残っている書き込みデータ情報を順次取り出して、処理しディスクに書き込んでゆく
- ④ このディスクへの書き込み処理を通じて、トランザクションを順次コミットして行く。
- ⑤ 全てのトランザクションがコミットされると、キャッシュが空になり、ロギング・エリアに余裕ができる
- ⑥ 手順②でロールバックしたトランザクションをキューから取り出し再処理した後、手順②で一時停止したトランザクションの受け付けを再開する。

(4) 定期的にチェックポイント処理を実行する

NTFS は定期的にチェックポイント処理を実行する。

このチェックポイント処理は、ファイル回復の処理に必要な 3 種類のチェックポイント情報をロギング・エリアに記録し、記録したチェックポイント情報の位置をリスタート・エリアに記録する処理である。

なお、チェックポイント処理でチェックポイント情報を記録している間にも、新たな更新レコードをロギング・エリアに記録し続けている。

【チェックポイント処理が書き込むチェックポイント情報】

チェックポイント処理では、①～④の順に4種類のチェックポイント情報を書き込む。

チェックポイント処理中にも、更新レコードを追加記録している

リスタート・エリア	ロギング・エリア						
チェック・ポイント・レコードの論理シーケンス番号 ④	...	ダーティ・ページ・テーブル ①	...	トランザクション・テーブル ②	...	チェック・ポイント・レコード ③	...

論理シーケンス番号でリンクする

- ① 1 個目のレコード (ダーティ・ページ・テーブル)
ファイル処理のワークエリア (メモリ上) のダーティ・ページ・テーブルをロギング・エリアに書き込む。このテーブルには、HDD への書き込みが終わっていない遅延書き込みキャッシュ領域ページ番号が書かれている
- ② 2 個目のレコード (トランザクション・テーブル)
ファイル処理のワークエリア (メモリ上) のトランザクション・テーブルをロギング・エリアに書き込む。このテーブルには、未コミットのトランザクションに含まれる更新レコードの論理シーケンス番号が書かれている
- ③ 3 番目のレコード (チェックポイント・レコード)
1 個目のレコード (ダーティ・ページ・テーブル) の論理シーケンス番号、2 個目のレコード (トランザクション・テーブル) の論理シーケンス番号が書かれている
- ④ チェックポイントレコード (③) を書き込んだ後に、その論理シーケンス番号を「リスタートエリア」に保存する。
またリスタート・エリアへのアクセスが不能になる場合に備えて、リスタートエリアのコピーを作成する

3.4.5. クラッシュ後の起動時にログデータを用いてファイルを復元する

(1) ファイル復元用のテーブル読み込み (準備段階)

クラッシュした後で、Windows を起動するとファイル復元の準備をする。

- ① クラッシュ後に、Windows を起動すると、自動的に [ログファイル] の [リスタート・エリア] をチェックする
- ② [リスタート・エリア] が初期化されず [チェックポイント・レコード] の論理シーケンス番号が書き込まれている場合は、論理シーケンス番号を取り込む
(初期化されている場合は、ファイル復元が不要なので通常の起動処理を行う)
- ③ [チェックポイントレコード] を読み込んで、最後に書き込んだ [トランザクションテーブル] と [ダーティページテーブル] の論理シーケンス番号を取り込む

- ④ 論理シーケンス番号を用いて、[トランザクションテーブル] と [ダーティページテーブル] を読み込む

(2) クラッシュ時の処理状況を調査 (分析パス)

ログファイルの更新レコードを用いて、メモリ上に読み込んだ [ダーティページテーブル] と [トランザクションテーブル] を最新状態に更新して、クラッシュ時にどこまで処理されていたかを分析する。

(補足) チェックポイントオペレーションの最中でも、新しい更新レコードの追加が行われているので、HDD に書き込まれた [ダーティページテーブル]、[トランザクションテーブル] にタイムラグが発生して最新状態ではない。

- ① [ダーティページテーブル] の直前の更新レコードを読み出して [リドウ情報] のデータと、それに対応する HDD のデータを読み出してデータが同じかどうかをチェックする
- ② データが同じである場合は、HDD に反映 (書き込み) 済の更新レコードと判断する。
同じでない場合は、HDD に未反映の更新レコードと判断して、反映済の更新レコードが見つかるまで、1 個ずつ前の更新レコードについてチェックを行う
- ③ 最初に見つけた HDD に反映済の更新レコードが最も新しい反映済の更新レコードであり、次のログレコードからチェックポイントオペレーションが始まったと判断する
- ④ チェックポイントオペレーションが始まった個所から、ロギング・エリアの終端に向けて順に更新レコードを読み込んで行き、そのデータをメモリ上の [ダーティページテーブル]、[トランザクションテーブル] に反映させて行き、各テーブルを最新の状態にする
- ⑤ ③項で見つけた HDD に反映済みの更新レコードの後ろにある更新レコードが、最も古い未反映 (未書き込み) の更新レコードであり、その論理シーケンス番号を取り込む

(3) 未反映の更新レコードの再実行 (リドウ・パス)

HDD に未反映 (未書き込み) の更新レコードを用いて、再実行 (リドウ) を行い、クラッシュ直前の可能な限り最新状態の遅延書き込みキャッシュを再現する

- ① 最も古い HDD に未反映の更新レコードの [リドウ情報] を再実行して、遅延書き込みキャッシュに書き込む。
- ② ロギング・エリアの終端までにある HDD に未反映の更新レコードを順番に再実行して行くと、遅延書き込みキャッシュは、クラッシュ直前までの (更新レコードで復元できる) 最新の状態になる
- ③ 遅延書き込みキャッシュに記憶されたデータを HDD に書き込むことで、HDD に未反映の更新レコードの再実行を終わる
なお、この HDD への反映 (書き込み) は、ログファイルに記録されない。

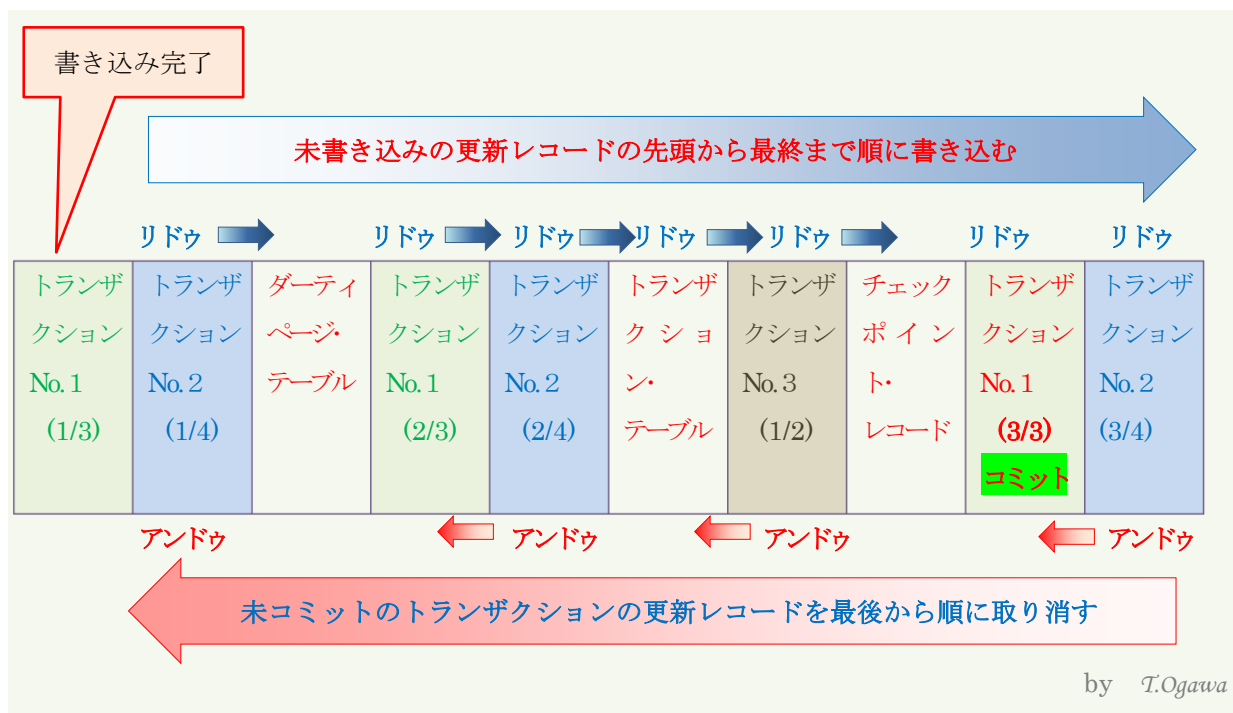
(4) 未コミットのレコードの取り消し (アンドゥ・パス)

HDD に未反映 (未書き込み) の更新レコードを用いて、ファイルを最新状態に復元した後、未コミット (トランザクションが未完了) の更新レコードをロールバックして、クラッシュで中断されたトランザクションによる変更を取り消して行く

- ① [トランザクションテーブル] を参照して、未コミットのトランザクションを探す
- ② 未コミットのトランザクションの更新レコードのうちで、最も新しい更新レコードを探し、論理

シーケンス番号を取り込む

- ③ ログインエリアからその更新レコードを探し、[アンドゥ情報] を実行する。
- ④ 同様の処理をログインエリアの先頭に向けて行い、そのトランザクションのロールバックを終わる
- ⑤ 未コミットの全てのトランザクションのロールバックが終わると、リスタートエリアを初期化して回復処理を終わる



4. PCの買い替えは Windows 8 搭載パソコンを待っては・・・

恒例のバージョンアップの時期がやって来た。Windows のバージョンアップである。

次期 Windows 8 (開発コード名) の内部バージョンは NT6.2 であり、Windows 7 (NT6.1) と同一のソースコードからビルドされている。

Windows 8 では、OS 本体は殆ど変更されず、添付されているユーティリティやアプリケーションが変更され、それらの画面や操作方法が大幅に変更されている。

ユーザが努力し苦勞して習得した操作技術はユーザの資産である。機能や能力に直接関係しない操作方法の変更は最少にすべきである。ましてや、競合他社のタブレット端末やスマホに対抗するためだけの操作方法の大幅変更などは実質的な独占企業の横暴であり、弊害である。

4.1. Windows 8 の発売時期は？

次期 Windows 8 (開発コード名) の発売時期について、春先頃からネット上で 2011 年～2012 年に発売などと諸説が入り乱れていたが、以下に示すように 2012 年の発売に収束してきた。

- 2011.5.23 に日本 MS が開催した開発者向け「MS Developer Forum 2011」で米 MS バルマー OEO が「次期 Windows は来年に登場する」と述べ、2012 年中に発売することを明言した。

- 現行 Windows 7 の場合は、2008.10.28 に開発者向けイベント「PDC 2008」の会場で、「PRE-RELEASE PREVIEW」と書かれた DVD を配布してから、開発者向けベータ版公開 (2009.1.13) → RC 版公開 (2009.4.3) → RTM 版リリース (2009.7.末) の過程を経て、1 年後の (2009.10.22) に製品版が発売された。

次期 Windows 8 は、2011.9.13~16 の開発者向けカンファレンス「BUILD」の初日に、開発者向けプレビュー版がネット (<http://msdn.microsoft.com/en-us/windows/home/>) に公開された。もし Windows 8 が Windows 7 と同じような過程を経て発売されるなら、上記プレビュー版の公開の 1 年後の 2012.9~10 に発売されると思われる。

- 筆者は、次表に示すコンシューマ向け Windows の発売周期 (2 年半~3 年) から推定して、Windows 8 (NT6.2) の発売を、Windows 7 (NT6.1) 発売 (2009.10.22) の 2 年半~3 年後である 2012 年 4 月末~2012 年 10 月末 と考えていた。

【Windows のバージョンと発売日】 ……フリー百科事典「ウィキペディア」より

製品名	バージョン	系	発売日 (日本語版)	次期までの期間
Windows NT3.5	NT3.5	NT	1994 年	
Windows 95	4.0	9×	1995 年 11 月 23 日	2 年 8 カ月間
Windows NT4.0	NT4.0	NT	1996 年	
Windows 98	4.1	9×	1998 年 7 月 25 日	2 年 8 カ月間
(Windows 2000)	NT 5.0	NT	2000 年 2 月 18 日	(1 年 11 カ月間)
Windows ME	4.9	9×	2000 年 9 月 22 日	1 年 2 カ月間
Windows XP	NT 5.1	NT	2001 年 12 月 31 日	5 年 1 カ月間
Windows Vista	NT 6.0	Vista	2007 年 1 月 30 日	2 年 9 カ月間
Windows 7	NT 6.1	Vista	2009 年 10 月 22 日	2 年~3 年 ?
Windows 8 (仮称)	NT 6.2 ?	Vista	2012 年 5 月 ~2012 年 11 月	

4.2. Windows 8 はユーザ・インタフェースが激変する？

↓ <http://pc.nikkeibp.co.jp/special/windows8/?set=ml>

ネット上の情報によると、Windows 8 は パソコンとタブレット端末をターゲットにして、インテルその他の x86 系 CPU と英アーム社その他の ARM 系 CPU に対応するといわれている。

そのため、スマートフォンやタブレット端末で採用している、タッチ操作、タイル操作、スライド操作、ソフトキーボード操作等も取り入れられている。

勿論、次期 Windows 8 は既存のハードウェアにインストールして使うこともできるが、激変するユーザ・インタフェースを十分に活用するには、Windows 8 用に作成されたパソコンが望ましいと考える (人に教える立場では、教わる人と同じ環境が望ましいから)。

4.3. Windows 8 の新機能

MS は次期 Windows に搭載される機能については公表していないが、各種の情報が数週間ほど前から聞こえてくるようになった。

現時点で Windows 8 に関して明らかになっている情報には次のものがある。

↓ <http://headlines.yahoo.co.jp/hl?a=20110530-00000051-zdn tt-sci> (9/23 リンク切れを確認)

- クロスプラットフォーム OS :

x86 系 CPU 以外に、タブレット端末で採用されている ARM 系 CPU にも対応する

- ユーザ・インタフェースの刷新（改善） :
 - キネクト
Kinect（ジェスチャーや音声認識でプレイできる MS 製のゲームシステム）で使用、
 - メトロ
Metro UI（Windows Phone 7 で採用されたタイルベースの UI）の採用
- Internet Explorer 10 :
 - HTML5 標準のサポートを強化、CSS（Cascading Style Sheets）との互換性を改善、タッチインタフェースも搭載
- セキュリティの強化 :
 - スマート スクリーン
SmartScreen（インテリジェント型ファイルフィルタ）が Windows のファイルシステムに直接組み込まれ、潜在的に危険なファイルをユーザが不用意に起動するのを防止
- History Vault :
 - ヒストリ ヴォールト
History Vault 機能で、外付け HDD、NAS 等にデータを自動的にバックアップし、任意の時点で復元
 - この機能は、Mac OS X の^{タイム マシン}Time Machine機能に似ているといわれている
- リセット :
- 仮想化機構（Hyper-V）:
 - これまでサーバ OS 向けに提供されていた仮想化機構であり、Windows 7用の Microsoft Virtual PC を更に発展させた仮想マシン機能である。（MS の公式ブログより）
- ハイブリッドブート（Hybrid Boot）:
 - 「シャットダウン」と「休止状態」の中間に位置するものであり、従来の「休止状態」からの再開に加えて、ドライバの初期化を行い安定した状態で起動できる（開発者向けプレビュー版より）

4.4. Windows 8 でパソコンの仕様は変わる？

ネット上で見られる情報から推測すると、Windows 8 対応として作られるパソコンは次のようなものになるのではないかと考えられる。

- CPU 現行 Windows 7 マシンと同等の CPU
- メモリ 現行 Windows 7 マシンと同等の容量
- ディスプレイ アスペクト比 16 : 9（1366×768 以上）、タッチセンサー付
- ファームウェア UEFI の搭載（2.2.2 項を参照）
- インタフェース USB3（現時点では高級機⇒中級機以下にも実装？）

本文中に出てきた用語の説明等

- (※1) ROM (Read Only Memory) : 読み出し専用メモリを言い、電源を供給しなくてもデータが消えないメモリの総称である。ROMには書き換えが不可能なマスク ROMの他に、書き換えが可能なROM (PROM、EPROM、EEPROM)がある。また、ROMではないがフラッシュメモリをROM的に使うことも多い。
- (※2) BIOS (Basic Input Output System) : Firmwareの一つであり、システム起動の前段階の機能と最低限の入出力機能を実行するプログラム群を言う。
現在は入出力機能の殆どをOSのシステムコールを経由してデバイスドライバが行っているため、BIOSの入出力機能の出番は少ない。
Firmには、硬い、固定的などといった意味があり、ファームウェアとは、ハードウェアの制御を行うためにハードウェアに埋め込まれたソフトウェアで、むやみに書き換えしない媒体(例:ROM)に記録されたソフトウェアを言う。
- (※3) POST (Power On Self-Test) : 電源投入後やハードウェアリセットの直後に行なわれるハードウェアの診断テストのこと言う。PC/AT互換機では、このためのPOSTプログラムがBIOS ROMに記録されている。電源を入れると自動的に、CPUやメモリ、グラフィックスコントローラ、ディスクコントローラ、シリアル/パラレルポート、キーボードコントローラなど、マザーボード上の各デバイスに関する診断が実行され、障害が発見された場合は、その結果がディスプレイ表示されたり、ディスプレイ表示が行えない場合には、ビープ音を鳴らしたりして、障害の原因を報告する。
- (※4) MBR (Master Boot Record) : PC/AT互換機では、HDDの先頭のセクタ(512B)を言う。
- (※5) BIOS入出力サブルーチンには、次の例に示すものがある。
INT(0x10) : ビデオ関係 INT(0x11) : 周辺機器リストの取得
INT(0x12) : 使用可能メモリのサイズ取得 INT(0x13) : ディスク関係
INT(0x14) : シリアルポート初期化 INT(0x15) : カセット関係
- (※6) Boot Strap Loader : PC/AT互換機では、MBRの先頭のプログラム領域(446B)にあるBS読み込み用のプログラムを言う。
- (※7) CHS (Cylinder/Head/Sector)方式はHDDセクタのアドレスの表示方式である。CHR方式では、フラッタ(円盤)の外周から内周へのシリンダ番号(0~1023、10bit)、磁気面のヘッド番号(0~254、8bit)、回転方向のセクタ番号(1~63、6bit)でセクタアドレスを表す。24bit(3B)を使用して約8.4GB(約16.45Mセクタ)のアドレスを表示できるが、HDDの大容量化により既に無意味になっている。
- (※8) LBA (Logical Block Address、論理セクタアドレス)方式はHDDセクタのアドレスの表示方式である。LBA方式では、HDDセクタのアドレスを、Cylinder=0、Head=0、Sector=1を“0”とした通し番号で表す。32bit(4B)を使用して約2.19TB(約4.29Gセクタ)まで対応できる。
なお近年のHDDは、内蔵コントローラでLBA(通し番号)をCylinder番号、Head番号、Sector番号に変換し目的のセクタにアクセスしている。
- (※9) BS (Boot Sector) : PC/AT互換機では、各パーティションの先頭セクタ(512B)を言い、MBRに対してPBR (Partition Boot Record)と呼ぶこともある。
- (※10) IPL (Initial Program Loader) : 1964年に発売したIBM System/360ではOS起動用の24BのファームウェアをIPLと呼んだ。PC/AT互換機もこの名前を踏襲している。

- (※11) M F T (Master File Table) は、NTFS の中核をなすテーブルであり、NTFS 上のすべてのファイル进行管理している。(詳細は、3.3 節を参照のこと)
- (※12) BOOTMGRは、複数 OS を切り替えながら起動する機能を持つ OS ロードである。これは NT 系 Windows (2000/XP 等) のNTLDRに相当するものである。
- (※13) B C D (Boot Configuration Data、ブート構成データ) は、マルチブート (複数 OS を切り替えて起動) の場合に、起動できる OS 情報 (OS 名、保存場所、起動パラメータ) をバイナリ形式で記述したファイルであり、OS を選択するための画面でこのデータを使用する。これは NT 系 Windows (2000/XP 等) のテキスト形式のBOOT.INIに相当する。
- (※14) HALは、OS カーネルとハードウェア (CPU) との間にあり、ハードウェアの違いを吸収するプログラム部品であり、インストール時にシステムの種類 (例: シングルプロセッサ、マルチプロセッサ、ACPI 対応、ACPI 非対応等) を自動的に識別して該当する種類の HAL を組み込む。
- (※15) レジストリとは、ハードウェア、インストールしたアプリケーションの設定、各ユーザアカウントのプロファイルに関する情報等を格納している Windows 内のデータベースである。Windows は、絶えずレジストリ内の情報を参照している。
通常、プログラムはレジストリの変更をすべて自動的に行うため、自分で責任を採れる上級者以外は手動でレジストリを変更してはならない。
- (※16) ドライバは、アプリケーションと OS の間に位置して、PC に接続した周辺装置を制御・操作するためのソフトウェアである。製品に添付された CD-R、製造元の Web サイト、Windows Update で入手してインストールする。
- (※17) SMSS.EXE (Session Manager Sub System) は、ユーザセッションの起動 (ページファイルの追加、ファイル名の変更と削除処理、システム環境変数の作成)、必要に応じてチェックディスクを実行、Windows Update その他のインストールの後処理、テキストモード⇒グラフィックモードの画面モード切り替えなどを行う。
- ↑ http://www.atmarkit.co.jp/fwin2k/bookpreview/insidewin4/insidewin4_07.html
↑ http://blog.goo.ne.jp/limited_terra/e/3e6dc5125a2693005b84bf55fe0a4405
- (※18) WININIT.EXE (Windows スタートアップアプリケーション) は、Windows のバックグラウンドサービスの一部 (例: LSASS.EXE、SCM.EXE、LSM) を開始する。
なお、Windows サービスとは、Windows 配下で長時間動作し、ユーザとのやりとり無しで特定機能を実行するものをいう。[コントロールパネル] → [システムとセキュリティ] → [管理ツール] → [サービス] で「サービス (ローカル) の内容」、[サービス (ローカル) の状況」等が確認できる。
- (※19) LSASS.EXE (Local Security Authority Service) は、ユーザアカウントの管理、ユーザログインの認証に関するサービスを行う。
- ↑ <http://www.processlibrary.com/ja/directory/files/lsass/24751/>
- (※20) SERVICES.EXE (Service Control Manager) は、Windows の開始と停止の操作を管理すると共に、ブート処理中にサービスを自動開始させ、シャットダウン中にサービスを停止する。
- ↑ <http://www.processlibrary.com/ja/directory/files/services/24770/>
- (※21) MBR の管理限界の 2 TB は、セクタアドレスが 4B (32bit) の LBA のためである。
このため MBR で管理可能なセクタ数は、4G ($2^{32}=2^2 \times 2^{30} \doteq 4 \times 10^9 = 4\text{G}$) となり、
MBR で管理可能な HDD 容量は、2TB ($4 \times 10^9 \text{セクタ} \times 512\text{B/セクタ} \doteq 2 \times 10^{12}\text{B} = 2 \text{ TB}$)

=2兆バイト) である

(参考) 数の単位 (表の5行目は、 $2^{10}=1024$ を 10^3 とみなした単位である)

10進数	10^3	10^4	10^6	10^8	10^9	10^{12}	10^{15}	10^{16}	10^{18}	10^{20}	10^{21}	10^{24}
単位	キロ K	—	メガ M	—	ギガ G	テラ T	ペタ P	—	エクサ E	—	ゼタ Z	ヨタ Y
		まん 万		おく 億		ちょう 兆		けい 京		がい 垓		秭/穄
2進数	2^{10}		2^{20}		2^{30}	2^{40}	2^{50}		2^{60}		2^{70}	2^{80}
$1024 \doteq 10^3$	キビ Ki		メビ Mi		ギビ Gi	テビ Ti	ペビ Pi		エクスビ Ei		ゼビ Zi	ヨビ Yi

$10^{24} = \text{秭 or 穄}$ (字形が似ている穄を代用することがある)

- (※22) 8ZBの管理限界を持つGPTでは、セクタアドレスに8B(64bit)のLBAを用いている。
 このためGPTで管理可能なセクタ数は、 $16E$ ($2^{64}=2^4 \times 2^{60} \doteq 16 \times 10^{18}$) となり、
 GPTで管理可能なHDD容量は、 $8Z$ (16×10^{18} セクタ \times 512B/セクタ $\doteq 8 \times 10^{21}$ B) である。
 これは、 $2TB$ (MBRでの管理限界) の40億倍の容量である
- (※23) PC/AT互換機：1984(S59)に発売したIBM PC/AT (16bitCPUの80286、メモリ256~512KB、16bitのISAバス、5.25"のFDD)の回路図やBIOSソースコード等の仕様を公開したため、PC/AT互換機市場が形成された。これには、大ヒットしたMS-DOS用の表計算ソフトのLotus 1-2-3が大きく寄与したといわれている。
 PC/AT互換機は、技術進歩に伴って規格競争と標準化を繰り返しながら変身を続けて、27年後の現在では、BIOSだけがオリジナルのPT/AT仕様を残すのみとなっている。そもそもBIOS (Basic Input Output System) は、MS-DOSの時代には名の示すとおりの入出力処理という重要な機能を提供していたが、現在ではブート処理での役割が主な機能になっている。
- (※24) サービスは、Windowsの起動時に自動的に起動されバックグラウンドで走行して、ユーザとのやりとり無しで、特定機能を実行するプロセスである。
 なおVista/7の場合は、実行中のサービスを「タスク・マネージャ」の「サービス」で確認することができる。
- (※25) 構成マネージャは、レジストリ (PCの構成情報等を保存するデータベース) によってシステムの構成を管理している。Windowsのシャットダウン時にメモリ上のレジストリをHDDに保存して、次の起動時にレジストリの矛盾が出ないようにしている
- (※26) メモリマネージャは、ユーザモード、カーネルモードで使用するすべてのメモリを管理する。
 Windowsのシャットダウン時に、キャッシュ (遅延書き込みメモリ) に書き込まれたデータを正しくHDDに保存して、次の起動時にNTFSに矛盾が出ないようにする。
- (※27) I/Oマネージャは、ファイル、ネットワーク、デバイスとの入出力を管理する
- (※28) エグゼクティブは、各種のカーネルサービスを提供するコンポーネントの集合体である。
- (※29) パワーマネージャは、PCの中断 (スリープ、休止状態)、中断からの再開、自動シャットダウン等の管理をおこなう。
- (※30) FAT32 (File Allocation Table) は、ビル・ゲイツらによって開発されたDISK-BASIC用のファイル管理システムであり、その後MS-DOSに採用された。FATはMicrosoft Disk Basic (1977年) のFAT12 (MRR=0x01、最大ファイル23MB、最大ボリューム32MB)、Compaq DOS 3.1 (1987年) のFAT16 (MBR=0x04/0x06/0x0E、最大ファイル2GB、最大ボリューム2GB)、Windows 95 OSR2 (1996年) のFAT32 (MBR=0x0B/0x0C、最大ファイル4GB、最大ボリューム2TB) へと発展した。

(※31) エヌティエフエス ファイル システム NTFS (NT File System) は、Windows NT 系の OS (NT 3.1~4.0、2000、XP、Vista、7 等) に対応したファイルシステムである。NTFS は大容量 HDD の管理ができるだけでなく、FAT32 には無かった高度なファイル管理機能 (パーミッション、ディスククォータ、圧縮、暗号化、堅牢性、耐障害性、等々) が備わっている。

(※32) 8.3 形式は、初期の MS-DOS のファイル名の命名方式であり、ファイル名 (ベース名) が最大 8 文字、拡張子が最大 3 文字であったことからこの名がついた。

(※33) ポジックス ポータブル オペレーティング システム インタフェース ユニックス アイ・ティ・ピー・イー POSIX (Portable Operating System Interface for UNIX) は、IEEE (米国電気電子学会) の規格であり、移植性の高いアプリケーション開発を容易にするために、UNIX その他の OS に共通の アプリケーション インタフェース API (Application Interface) を定めている。単に POSIX という場合はシステムコール、ライブラリ関数を規定した POSIX.1 (IEEE Std 1003.1) を指す。

(※34) パーミッションは、ファイルやフォルダに対するユーザのアクセス権を言う。

NTFS ではドライブ単位/フォルダ単位/ファイル単位に、[プロパティ] の [セキュリティ] 画面で、「読み取り」、「書き込み」、「読み取りと実行」、「フォルダ内容の一覧表示」等を設定できる。

アクセス権は アクセス コントロール リスト ACL (Access Control List) に設定され、ユーザのアクセス制御に用いられる。

(※35) ディスククォータは、ユーザ毎のディスクの割当量を言う。NTFS では、ドライブの [プロパティ] 画面の [クォータ] をクリックし「クォータエントリ」画面を開いて、[クォータ] メニューの [新規クォータエントリ] で制限をするユーザを指定した後、必要な設定を行ってユーザの上限容量を設定する。

(※36) 透過的な圧縮は、知識なしで簡単な操作で圧縮できる方式を言う。

NTFS のフォルダ単位、ファイル単位の圧縮の手順は次のとおりである。

- ① フォルダあるいはファイルの [プロパティ] の [全般] 画面で、
- ② [詳細設定] をクリックし [属性の詳細] 画面を開き、
- ③ 「内容を圧縮してディスク領域を節約する」にチェックを付けた後、[OK] をクリックして [全般] 画面に戻り、
- ④ [適用] をクリックして [属性変更の確認] 画面を表示させ、[OK] をクリックして圧縮処理を開始させる。
- ⑤ 圧縮が終わったら、[OK] して操作を終了すると、圧縮されたフォルダ、ファイルが青色で表示される。

なお、圧縮したフォルダ、ファイルを元に戻す手順は、前記の圧縮の操作の③で「内容を圧縮してディスク領域を節約する」のチェックを外すとよい。

NTFS のドライブ全体を圧縮する場合は、ドライブの [プロパティ] の [全般] 画面で設定する。

(※37) 透過的な暗号化は、知識なしで簡単な操作で暗号化できる方式を言う。

NTFS のフォルダ単位、ファイル単位の暗号化の手順は次のとおりである。

- ① フォルダあるいはファイルの [プロパティ] の [全般] 画面で、
- ② [詳細設定] をクリックして [属性の詳細] 画面を開き、
- ③ 「内容を暗号化してデータをセキュリティで保護する」にチェックを付けた後、[OK] をクリックして [全般] 画面に戻り、
- ④ [適用] をクリックして [属性変更の確認] 画面を表示させ、[OK] をクリックして暗号化処理を開始させる。
- ⑤ 暗号化が終わったら、[OK] して操作を終了する。

なお、暗号化したフォルダ、ファイルを元に戻す手順は、前記の暗号化の操作の③で「内容を暗号化してデータをセキュリティで保護する」のチェックを外すとよい。

- (※38) ロールバックは、ジャーナル・ログを用いてトランザクション単位でファイル更新を取り消してファイルを復元することを言う。

NTFS では、トランザクション処理によりファイルの書き込み前にログを記録している。HDD への書き込み中にクラッシュが発生した場合でも、システムの再起動時に記録したログを使用して自動的に不完全なデータを消去（ロールバック）して矛盾のない状態に回復する。

FAT32 では、このような機能がないので、書き込み中の不完全なデータが HDD に残り、管理情報と実データに矛盾が生じるので、最悪の場合は HDD にアクセスできなくなることがある。

- (※39) MFT 専用の予約領域：これは MFT ファイルの断片化が進んで、パフォーマンスが低下するのを防止するため NTFS ボリュームの 12.5% を MFT ファイル用に確保しているものである。

なお、「Windows7 では MFT 領域は自動的に確保され、ファイルシステム初期化直後は・・・」と書かれているが、詳細は確認できていない。

↑ http://ja.wikipedia.org/wiki/%E3%83%9E%E3%82%B9%E3%82%BF%E3%83%BC_%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB_%E3%83%86%E3%83%BC%E3%83%96%E3%83%AB

- (※40) クラッシュとは、アプリケーションやシステムが突然停止することや、ハードウェアの不具合などでコンピュータが動作しなくなることを言う。

類似した言葉にハングアップ（処理を進められない停止状態）、フリーズ（ソフトの動作が停止して操作を受けつけなくなる現象）、ブルースクリーン（重大なエラーが発生し Windows が停止して青い画面に白文字でエラー表示する状態）、暴走（制御不能で勝手に異常動作をする）等がある。

- (※41) プリエンプティブ・マルチタスクは、タイマ割り込みを利用して OS が優先権を得て、タスクを切り替える時分割のマルチタスクである。これは、アプリケーションが空き時間を他のアプリケーション用に開放するノンプリエンプティブ・マルチタスクより安全である。

- (※42) ケアフル・ライトは、アプリケーションからのファイル更新を 1 個ずつ順番にディスクに書き込んで安全性を高めている。しかしディスクへの書き込み時間が CPU の処理時間に比べて桁違いに長いので、処理が込み合うと待ち時間が大きくなり効率が悪くなる。

- (※43) 遅延書き込みは、アプリケーションからのファイル更新をメモリ上の遅延書き込みキャッシュに記録しておき、ディスクが空いた時点でディスクに書き込むことで、ケアフル・ライトに伴う待ち時間の問題を低減する。

しかし、アプリケーションが OS から変更完了を受けてから実際にディスクが更新されるまでにタイムラグが発生するので、この間にクラッシュ等があるとディスクに保存したはずのデータがディスクに存在しない事態が発生する。

- (※44) トランザクション処理は、メインフレーム（企業用の大規模コンピュータ）やサーバ等に用いられてきた「オール・オア・ナッシング」の処理方式である。

トランザクション処理では、ある目的に関連する複数の処理を 1 個のトランザクションとして定義して、トランザクションに含まれるすべての処理が成功した場合にのみ、データベース（ファイル）が更新される仕組みになっている。

NTFS のトランザクションは、データファイルと管理情報ファイルの更新をトランザクション処理してファイルに不整合が発生するのを防ぐ OS レベルの機能である（FAT32 はこの機能を持ってい

ない)。

企業が業務処理などで用いている、データ内容をも保障するアプリケーション・レベルのトランザクション処理とは、次元が異なるものである。

- (※45) ファイル更新とは、ファイルやフォルダの新規作成、削除、更新、属性等変更、コピー、移動等を行う処理を言う。
- (※46) ログファイル (\$LogFile) は、MFT の3番目のレコードであり、回復処理に使用するレコード位置を知るための「リスタート・エリア」と更新レコードやチェックポイント関係のレコードを記録する「ロギング・エリア」に分かれている。ロギング・エリアはサイクリックに使用される。
- (※47) コミット (Commit) は、トランザクションが全て正常に完結したという宣言をいう。

《 END 》