

Nextor 2.1 ユーザーマニュアル

By Konamiman, 2022/03/22 翻訳 : @Lithelia(2025/8/8)

目次

1. はじめに

1.1. 背景

1.2. 目的

1.3. システム要件

2. 特徴

2.1. FAT16ファイルシステムのサポート

2.2. 標準化されたドライバー開発システム

2.3. デバイス/パーティションのドライブ割当管理

2.4. ドライブのロック

2.5. 割当情報縮小/ゼロモード

2.6. Z80アクセスモード

2.7. 高速STROUTモード

2.8. 拡張マッパーサポートルーチン

2.9. ブートキー

2.9.1. ブートキーインバータ

2.9.2. ワンタイムブートキー

2.10. 組込パーティション設定ツール

2.11. 組込 MSX-DOS 1

2.12. 組込 Disk BASIC

2.13. ファイルのマウントとディスクエミュレーションモード

2.14. KILLDSKIO環境変数

3. Nextorの利用

3.1. Nextorのインストール

3.1.1. Sunrise IDE/CF ユーザへの注意

3.2. Nextorのブート

3.2.1. DOS1モードのブート

3.3. メディアの交換管理

3.3.1. DOS1モードのメディア交換

3.4. コマンドラインツール

3.4.1. MAPDRV: ドライブマッピングツール

3.4.2. DRIVERS: ドライバー情報ツール

3.4.3. DEVINFO: デバイス情報ツール

3.4.4. DRVINFO: ドライブ情報ツール

3.4.5. LOCK: ドライブのロック/アンロックツール

3.4.6. RALLOC: 割当情報縮小/ゼロモードツール

3.4.7. Z80MODE: Z80アクセスモードツール

3.4.8. FASTOUT: 高速STROUTモードツール

3.4.9. DELALL: パーティションクイックフォーマットツール

3.4.10. NSYSVER: NEXTOR.SYSバージョンチェンジャー

3.4.11. NEXBOOT: ワンタイムブートキー設定ツール

3.4.12. EMUFILE: ディスクエミュレーションモードツール

3.5. 組込パーティション設定ツール

3.6. 拡張 Disk BASIC

3.6.1. DSKF 関数

3.6.2. DSKI\$ 関数/DSKO\$ 命令

3.6.3 CALL NEXTOR 命令

3.6.4 CALL CHDRV 命令

3.6.5 CALL CURDRV 命令

3.6.6 CALL DRIVERS 命令

3.6.7 CALL DRVINFO 命令

3.6.8 CALL LOCKDRV 命令

3.6.9 CALL MAPDRV 命令

3.6.10. CALL MAPDRVL 命令

3.6.11. CALL USR 命令

3.6.12. CALL SYSTEM2 命令

3.7. 新しい BASIC エラーコード

3.8. ファイルのマウント

3.9. ディスクエミュレーションモード

3.9.1. ディスクエミュレーションモードの開始と終了

3.9.2. イメージファイルの変更

3.9.3. ルールと制限

3.9.4 メモリを解放する方法

3.9.5. 既知のバグ

4. その他の改善点

4.1. F7のload"

4.2. 漢字モードでの英語エラーメッセージ

4.3. 日本語エラーメッセージを削除したNEXTOR.SYS

5. 変更履歴

5.1. v2.1.3

5.2. v2.1.2

5.3. v2.1.1

5.4. v2.1.1 beta 2

5.5. v2.1.1 beta 1

5.6. v2.1.1 alpha 2

5.7. v2.1.1 alpha 1

5.8. v2.1.0

5.9. v2.1.0 RC

5.10. v2.1.0 beta 2

5.11. v2.1.0 beta 1

1. はじめに

Nextorは、MSXコンピュータ用のディスクオペレーティングシステムである MSX-DOS 2 の拡張版です。MSX-DOS 2.31 をベースとしており、100% の互換性があります。このドキュメントは、Nextor が MSX-DOS 2 に追加した機能について説明します。主にエンドユーザーを対象としていますが、開発者にも役立つ基本概念についても説明しています。開発者向けには、他に [Nextor 2.1 プログラマーズリファレンス](#) と [Nextor 2.1 ドライバー開発ガイド](#) という 2 つのドキュメントがあります。このドキュメントの読者は、少なくともユーザーレベルで MSX-DOS 2 の使用経験があることを前提としています。

1.1. 背景

MSX-DOSは、MSXコンピュータ用の唯一の公式ディスクオペレーティングシステムです。最終バージョンである2.31は、1990年にMSX turboRコンピュータに搭載されて登場しました。

MSX-DOSは、MSXコンピュータで大容量ストレージの選択肢がフロッピーディスクしかなかった時代に開発されました。「フロッピーディスク専用オペレーティングシステム」として使用した場合、MSX-DOSは確かに問題なく動作しました。しかし長年利用される間に、アマチュアが作成したハードウェアという形で、より現代的な大容量ストレージオプションが登場しました。90年代初頭のSCSIおよびIDEハードディスクコントローラから、今日のマルチメディアカードリーダーまで、様々なデバイスが存在します。MSX-DOSはこれらのデバイスの管理に使用されてきましたが、いくつかの問題がないわけではありません。

- MSX-DOS はセクター番号を16ビットエンティティとして扱い、サポートするファイルシステムは FAT12 のみです。そのため、単一のファイルシステムボリュームのサイズは 32MB に制限されます。FAT16 ファイルシステムのサポートを追加するための非公式パッチが開発されました。
- 実際のデバイスドライバー（大規模ストレージコントローラハードウェアと相互作用するコード）は、オペレーティングシステムのカーネル ROM 内に組み込まれており、フロッピーディスクドライブ内蔵コンピュータや外付けフロッピーディスクコントローラに搭載されています。カーネル ROM 内にカスタムデバイスドライバーを埋め込む方法については、公式に文書化された方法は存在しません。カスタムストレージコントローラハードウェアの開発者は、カスタムドライバーを埋め込むためにカーネルコードをリバースエンジニアリングする必要があります。
- ユーザーから見たドライブレターと、デバイスドライバー API によって公開されるデバイスユニットの間には、1対1の直接的な対応関係があります。例えば、ドライブ A: にアクセスする場合、MSX-DOS はドライバーに最初のデバイスへのアクセスを要求します。一方、ドライブ B: にアクセスする場合は、2番目のデバイスへのアクセスを問い合わせます。フロッピーディスクの場合は問題ありませんが、1つ以上のパーティションを持つより複雑なデバイスを使用する場合は、ドライブとデバイス、およびパーティションの割り当てをドライバー（通常はドライバー開発者が作成した外部ツールも）が管理する必要があります。
- 非ブロックデバイス（CD-ROMなど）の管理は非常に困難です。カーネルコードのリバースエンジニアリングという大変困難な作業が必要になるためです。

1.2. 目的

Nextor の主な目的は、21世紀のストレージデバイスを搭載したMSXコンピュータに必要な機能を、MSX-DOS 2 をベースとして実装することにより、前述の問題を解決することです。具体的には、Nextor の開発が目指す主な努力目標は以下のとおりです。

- FAT16 ファイルシステムのネイティブサポートを提供する

- カスタムストレージデバイスドライバーを開発し、OSカーネルROMに組み込むか、RAMから動的ロード可能とする、標準化されたシステムを提供する
- (MSX-DOS のドライブベースシステムとは対照的に) デバイスベースドライバーシステムを提供することにより、ドライバー開発者はストレージデバイスの列挙とアクセスのみを配慮すればよく、デバイスとパーティションに対するドライブへの割り当てはオペレーティングシステムが管理する
- 非ブロックデバイス、および FAT12/16 以外のファイルシステムを持つブロックデバイスをサポートする

これらの主な目的に加え、Nextor は MSX-DOS にはない副次的に便利な新機能も提供しています。詳細については以後の章を確認下さい。

1.3. システム要件

Nextorは、少なくとも128KBのメモリマッパーRAMを持つMSXコンピュータ（MSX1以降）で動作します。メモリマッパーRAMを持たないか、最大メモリが128KB未満のコンピュータでは、NextorはMSX-DOS 1モードで起動します（DOSプロンプトはコンピュータに64KBのRAMがある場合のみ使用できます）。

ダミーデバイスドライバーとスタンドアロン版のNextorをROMに書き込むことで、MSX-DOSカーネルに関連付けられたストレージコントローラと組み合わせて使用することもできます。それにより、FAT16ファイルシステムのサポートやZ80アクセスモードなどの機能を利用可能です。ただし、ドライブとデバイス/パーティションの割り当て管理機能には、Nextor専用のデバイスドライバーが必要です。

2. 特徴

この章では、MSX-DOS 2の拡張版となるNextorの機能について説明します。操作の詳細については、以降の章で説明します。

2.1. FAT16ファイルシステムのサポート

NextorはFAT16ファイルシステムを標準サポートしています。パッチをインストールする必要はなく、FAT16ボリュームからシステムを起動することも可能です。最大4GBのボリュームを使用できます。

さらに、標準ブートセクター（フォーマット済み、パーティション分割済みデバイス、あるいはほかのPCコンピュータによるフォーマットやパーティション分割済みのデバイス）も完全にサポートされています。一方、MSX-DOS 2では、それ自身でフォーマットされていないディスクはすべてMSX-DOS 1ディスクとして扱われました。

2.2. 標準化されたドライバー開発システム

カスタムストレージコントローラハードウェアの開発者は、標準化されたカスタムドライバー開発システムを利用できるようになりました。ドライバーの構造、実装するルーチンの詳細、そしてNextorカーネルにドライバーを埋め込むための「レシピ」が提供されているため、リバースエンジニアリングは不要です。

このドライバーの主な目的は、ストレージデバイスの列挙とアクセスですが、カスタムBASICステートメント（CALLコマンド経由）、拡張BIOSコマンド、またはタイマー割り込みサービスルーチンを追加するための拡張ポイントも備えています。

Nextorデバイスドライバー開発者向けに、以下のリソースを用意しています。

- [Nextor 2.1 ドライバー開発ガイド](#) ドキュメント。
- カスタムドライバー開発のスケルトン（訳注：骨格、屋台骨）として使用できるテンプレートドライバーファイル DRIVER.ASM。
- デバイスドライバーを Nextor カーネル ROM に埋め込むためのすべての作業を実行するコマンドラインユーティリティ MKNXEROM。Windows 実行ファイル (MKNEXROM.EXE) と標準 C ソースファイル (MKNEXROM.C) の形式で提供されます。

注) 現時点では、Nextor デバイスベースドライバーが制御するデバイスに割り当てられたドライブに対して、FORMAT コマンドを実行することはできません。これは Nextor の将来のバージョンで変更される予定です。

2.3. デバイス/パーティションのドライブ割当管理

ドライバー開発者は、Nextor デバイスドライバーを開発する際に、ドライブベースドライバー と デバイスベースドライバー という 2 つのドライバースタイルを選択可能です。前者は、OS のドライブレターとドライバーユニットを 1 対 1 で割り当てることで MSX-DOS ドライバーを模倣します。ただし、ドライブとデバイス、およびパーティションの割り当てはドライバー側で管理する必要があります。後者ははるかに興味深いものです。

デバイスベースドライバーは、ドライバーユニットではなく、デバイスを直接操作します。つまり、ドライバーには「ユニット N のセクター X を読み取る」といったルーチンはなく、「デバイス X の情報を返す」や「デバイス X から RAW データを読み取る」といったルーチンが存在します。デバイスベースドライバーは最大 7 個のデバイスを処理でき、各デバイスは 1 個から 7 個の論理ユニットを持つことができます。

最も優れている点は、デバイスベースドライバーを使用する場合、Nextor がデバイスとパーティションのドライブ文字の割り当てを、自動 (ブート時、[3.2. Nextor のブート](#) を参照) と手動 (新しいファンクションコールを呼び出す割り当てユーティリティを使用、[3.4.1. MAPDRV: ドライブマッピングツール](#) と [3.6.9. CALL MAPDRV コマンド](#) を参照) の両方で処理することです。ドライバー開発者は、デバイスへの RAW アクセスを実装するだけで済みます。

2.4. ドライブのロック

Nextor では、ドライブにロック状態のマークを付与できます。ドライブがロックされると、カーネルコードはドライバーに対してドライブ内のメディアが変更されたかどうかを尋ねず、代わりにユーザーがメディアを変更することはないと想定します。これは、マルチメディアカードなどのリムーバブルデバイスをメインストレージデバイスとして使用する場合に便利です。これにより、カーネルがメディア検証コードの実行に時間を浪費するのを防ぐことができます。付属ツール LOCK.COM を使用するか、BASIC 上で CALL LOCKDRV コマンドを呼び出すことで、ドライブをロック可能です。MSX-DOS ドライバーに属するドライブ (フロッピーディスクドライブを含む) も含め、すべてのドライブをロックできます。[3.4.5. LOCK: ドライブのロック/アンロックツール](#) および [3.6.8. CALL LOCKDRV 命令](#) を参照してください。

2.5. 割当情報縮小/ゼロモード

Nextor では、ドライブを割当情報縮小モードに設定できます。このモードでは、ドライブの合計容量と空き容量に関する情報を返す ALLOC ファンクションコールは、必要に応じて偽の情報を返します。これにより、計算される合計容量または空きセクター数が常に 16 ビットに収まるようになります。つまり、割当情報縮小モードが有効なドライブでは、合計容量または空き容量が 32MB を超える場合 (FAT16 ボリュームではこの

可能性あり)、ALLOC は 32MB を返します。 [3.4.6. RALLOC: 割当情報縮小/ゼロモードツール](#) を参照してください。

この機能は、基盤となるファイルシステムが常に FAT12 であり、合計容量または空き容量が最大 32MB であると想定するようなアプリケーションとの互換性の問題を回避することを目的としています。

環境変数 ZALLOC に ON (大文字/小文字の区別なし) という値が設定されている場合 (コマンドプロンプトでコマンド `SET ZALLOC=ON` を実行)、割当情報縮小モードは割当情報ゼロモードになります。この場合 ALLOC は、このモードが有効なドライブに対して空き容量ゼロを返します。これは、デバイスの空き容量の計算 (例えば DIR コマンドの最後) に、大容量デバイスではやや長い時間がかかる場合 (例えば SD カードの Z80 モードでは約 4 秒) があるため便利です。ゼロ割り当て情報モードがアクティブな場合、この時間はゼロに短縮されます。

ゼロ割り当て情報モードは、Nextor 2.0.3 以降で利用可能です。

2.6. Z80アクセスモード

MSX turboR コンピュータでは、MSX-DOS 2 はディスクドライバーにアクセスする際、常に Z80 CPU に切り替えます。Nextor は、Nextor カーネルに統合されたドライバーにアクセスする際、CPU を変更することはありませんが、MSX-DOS カーネルに統合されたドライバーにアクセスする際は、Z80 アクセスモードの有効/無効を切り替えることができます。Z80 アクセスモードが有効になっている場合、Nextor は MSX-DOS と同様に、ドライバーにアクセスする前に Z80 に切り替えます。 [3.4.7. Z80MODE: Z80アクセスモードツール](#) を参照してください。

Z80 アクセスモードは、すべての MSX-DOS ドライバーでデフォルトで有効になっています。ドライバーごとにオン/オフを切り替えることができます (特定のドライブレターに対して変更することはできません)。

2.7. 高速STROUTモード

MSX-DOS のファンクションコール STROUT は、"\$"文字で終了する文字列を出力します。このファンクションコールは実際には、文字列の先頭から順に、CONOUT ファンクションコール (1文字出力) を個別に呼び出します。

Nextorでは 高速STROUT モードが導入されています。このモードが有効な場合、文字列はページ 3 の 512 バイトのバッファにコピーされ、カーネルコードへの 1 回の呼び出しで出力されるため、出力処理が高速化されます。ただし、このモードが有効な場合、文字列の長さが 511 バイトに制限され、それより長い文字列は表示前に切り捨てられます。 [3.4.8. FASTOUT: 高速STROUTモードツール](#) を参照してください。

2.8. 拡張マッパーサポートルーチン

※訳注：以下の2つのルーチンはv2.1.0 RC1で廃止されました。別の拡張マッパーサポートルーチンになっています。

MSX-DOS 2は、アプリケーションが16KのRAMセグメントを割り当てることを可能にするマッパーサポートルーチンを提供しています。Nextorはオリジナルのルーチンを維持しながら、特定のセグメント内に連続したメモリブロック (1バイトから16Kまで) を割り当てる ことができる2つの新しいルーチンを提供しています。詳細については、 [Nextor 2.1 プログラマーズリファレンス](#) を参照してください。

2.9. ブートキー

Nextor の起動時設定は、システム起動中に特定のキーを押し続けることで変更できます。これらのキーとその動作は以下のとおりです。

- **0:** パーティションテーブルからエミュレーションデータファイルポインタを削除することで、永続的なディスクエミュレーションモードを無効にします。[3.9. ディスクエミュレーションモード](#) を参照してください。
- **1:** MSX-DOS 1 モードで強制的に起動します。コンピュータが MSX turboR の場合は、CPU を Z80 モードに切り替えます。
- **2:** MSX-DOS 1 モードで強制的に起動します。コンピュータが MSX turboR の場合は、CPU を R800-ROM モードに切り替えます。MSX-DOS 1 モードでは、ディスクドライブへのアクセス時に CPU 切り替えが発生しないことに注意してください。そのため、一部のストレージデバイス、特にフロッピーディスクドライブなどの MSX-DOS ドライバーに制御されているデバイスが正常に動作しない可能性があります。
- **3:** 既存のブートコードを無視して、BASIC を強制的に起動します (つまり、NEXTOR.SYS、AUTOEXEC.BAS、またはブートセクター内のコードをロードして実行しようとしません)。
- **4:** (MSX turboR のみ) R800-ROM モードで起動し、(本体内部マッパーRAMの代わりに) 検出された最大サイズのマッパーRAMをプライマリマッパーとして割り当て、R800-DRAM モード用に割り当てられた 64K を解放します。これは、マッパー RAM が大量に必要で、プライマリマッパーでしか動作しないソフトウェアを使用する場合に便利です。ただし、システム速度が大幅に低下することに注意してください。
- **5:** デバイスベースドライバを持つ各Nextorカーネルには、ドライバが制御するデバイスの数に関係なく、1つのドライブのみを割り当てます。これは、Nextorが検出されたデバイスごとに1つのドライブを割り当てて通常の動作をオーバーライドします([3.2. Nextorのブート](#) を参照)。ただし、これはデフォルトの動作に過ぎず、ドライバは[DRV_CONFIG\(ルーチン\)](#)を実装することでこれをオーバーライドできます。
- **CTRL:** このキーの状態は、初期化時にMSX-DOSカーネルに渡されます。通常、これにより内蔵フロッピーディスクドライブの2つ目の「ゴースト」ドライブが無効化され、特にMSX-DOS 1モードではメモリを解放できるようになります。このキーはデフォルトで反転されているため、Nextor ROMをカスタマイズしない限り、逆の動作になります ([2.9.1. ブートキーインバータ](#) を参照)。
- **SHIFT:** MSX-DOSカーネルの起動をスキップしますが、Nextorカーネルは通常通り起動します。これは、特にMSX-DOS 1モードで、内蔵フロッピーディスクドライブを無効にしてTPAメモリを拡張する場合に便利です。
- **N:** 存在するすべての Nextor カーネルの起動を禁止します。これは、Nextor 以外のコントローラによって制御されるストレージデバイス (例: 内蔵フロッピーディスクドライブ) からカーネル ROM をアップデートする必要がある場合に便利です。
- **スロットキー:** 指定したスロットキーに関連付けられた Nextor カーネルの起動をスキップします。これは、別の Nextor コントローラによって制御されているデバイスからカーネル ROM を更新する必要がある場合に便利です。各スロットに関連付けられたキーは次のとおりです。
 - Q はプライマリスロット 1
 - A はプライマリスロット 2

- QWER はそれぞれスロット 1-0 から 1-3
- ASDF はそれぞれスロット 2-0 から 2-3
- ZXCV はそれぞれスロット 3-0 から 3-3
- まれに、スロット 0 に Nextor カーネルがある場合、それぞれ 0-0 から 0-3 に UIOP キーを使用

例: Nextor カーネルがプライマリスロット 1 にある場合、Q キーを押して起動をスキップします。スロット 2-3 にある場合は F を押します。

2.9.1. ブートキーインバータ

Nextorカーネルには、ROMのオフセット512と513に、ブートキーインバータとして機能する2バイトがあります。ブートプロセスに影響するキー（スロットキーは除く）にはそれぞれ1ビットが割り当てられており、そのビットがセットされるとキーの意味が反転します。例えば、Shiftキーのビットが押されると、ブート中にShiftキーを押さない限り、MSX-DOSドライバーは無効になります。

これらの値はハードコードされているため、カスタマイズするには、デバイスに書き込む前にNextor ROMファイルを変更するしかありません。MKNEXROM.EXEツールを使用するか、16進エディタを使って手動で変更することもできます。

各キーへのビットの割り当て方法は次のとおりです。

- 第 1 バイト (オフセット 512):
 - Bit 1～5: キー 1～5
- 第 2 バイト (オフセット 513):
 - Bit 5: CTRL キー
 - Bit 4: SHIFT キー

その他のビットは現在未使用であり、将来の拡張との互換性を確保するため、常に 0 に設定する必要があります。

MKNEXROM.EXE を使用する場合は、`/k` パラメータに 16 ビットの 16 進数値を指定する必要があります。この数値は、各キーの値を以下のように加算して作成してください。

- 1: 0002
- 2: 0004
- 3: 0008
- 4: 0010
- 5: 0020
- Ctrl: 2000
- Shift: 1000

例: `/k:3002` は、1、CTRL、SHIFT キーを反転します。

デフォルトでは（「公式」ROM ファイルを使用する場合）、CTRL キーのみが反転されるため、内蔵フロッピーディスクドライブはデフォルトで 1 つのドライブ文字のみが割り当てられて起動します。

2.9.2. ワンタイムブートキー

Nextor の起動手順を変更する別の方法として、ワンタイムブートキー があります。起動時に Nextor が RAM 内の特定の位置に特定の署名を見つけると、その署名に続く数バイトを読み取り、キーボードを無視してブートキー（スロットキーを含む）の値として使用します。使用される RAM 領域はページ 2 であるため、この方法は 32K 以上の RAM を搭載したコンピューターでのみ機能します。

RAM ベースのメカニズムであるため、署名とキーデータがメモリに再度保存されない限り、次のコンピューターのリセット時には再び機能しないという意味で「ワンタイム」です。この動作を一定に保つため、署名は Nextor によって読み取られた後に明示的に消去されます。

[NEXBOOT.COM ツール](#) を使用すると、このデータを簡単に設定してコンピューターをリセットできますが、このツールは RAM への書き込みのみを行うため、他のツールを使用することもできます。このメカニズムで使用されるデータの場合と形式の詳細については、[Nextor 2.1 プログラマーズリファレンス](#) ドキュメントに記載されています。

2.10. 組込パーティション設定ツール

Nextorカーネルには、BASICプロンプトでCALL FDISKを実行するだけで起動できるデバイスパーティションツールが組み込まれています。このツールを使用すると、Nextorデバイスベースドライバで制御されるデバイス上に、100KBから4GBまでの任意のサイズのパーティションを作成できます。[3.5. 組込パーティション設定ツール](#) を参照してください。

2.11. 組込 MSX-DOS 1

NextorカーネルにはMSX-DOS 1カーネルが含まれているため、必要に応じてこの環境で起動できます。Nextor版のMSX-DOS 1は、オリジナルバージョンと比較してユーザーや開発者向けの追加機能は提供していませんが、Nextorドライバーに接続されたデバイスにアクセスできるように内部的に変更されています。[3.2.1. DOS1モードのブート](#) を参照してください。

2.12. 組込 Disk BASIC

Disk BASICは新しい命令が拡張されました。また、既存命令の一部も改良されました。[3.6. 拡張 Disk BASIC](#) を参照してください。

2.13. ファイルのマウントとディスクエミュレーションモード

バージョン 2.1 以降、Nextor ではディスクイメージファイルを 2 つの方法でマウント可能です。

- 通常のブートと、ディスクイメージファイルをドライブにマウントする方法。[3.8. ファイルのマウント](#) を参照してください。
- ディスクエミュレーションモードでブートする方法。システムは MSX-DOS 1 モードでブートし、指定されたディスクイメージファイルを（一度に 1 つずつ）ブートデバイスとして使用します。[3.9. ディスクエミュレーションモード](#) を参照してください。

2.14. KILLDSKIO環境変数

Nextor バージョン 2.1.1 以降、BASIC 命令 `DSKI$` および `DSKO$` を無効にするメカニズムが提供され、BASIC 環境で 512 バイトの空きメモリを増加可能となります。これは、`KILLDSKIO` という環境変数を作成し、値を `ON` (大文字小文字は区別されません) に設定することで有効となります。

上記の設定によって **DSKI\$** および **DSK0\$** を無効にすると、これらの命令がセクターを読み書きする 512 バイトのバッファのアドレスを保持する **DIRBUF** ワークエリア (&HF351) の値が、Nextor が内部で使用する汎用セクターバッファである **SECBUF** (&HF34D) と同じ値になることに注意してください。また、BASIC が **FILES** などのコマンドのパス名を解析するために使用するバッファである **PATHNAM** (&HF33B) の値も同様です。ほとんどの場合、これは問題にはなりませんが、堅牢性を確保するため、追加のメモリが絶対に必要な場合にのみこの機能を使用することをお勧めします。

3. Nextorの利用

この章では、Nextor および関連ユーティリティの運用について説明します。

3.1. Nextorのインストール

Nextor は以下のコンポーネントで構成されています。

- Nextor カーネル ROM。デバイスドライバが含まれている必要がありますが、デバイスを公開しないダミードライバを含む「スタンドアロン」バージョンも提供されています。
- NEXTOR.SYS ファイル。DOS プロンプトが起動するために必要なファイルです。このファイルは、MSX-DOS 2 における MSXDOS2.SYS と同じ役割を果たします (実際、NEXTOR.SYS は MSXDOS2.SYS の拡張版です)。
- COMMAND2.COMファイル。Nextor 専用のコマンドインタープリタは用意されていません。代わりに、MSX-DOS 2 と同じコマンドインタープリタが使用されます (COMMAND2.COMのバージョンは 2.20 以降であればどれでも使用できます)。新しい機能は外部コマンドを使用して処理されます。

注意: NEXTOR.SYS ファイルには 2 つのバージョンがあります。_4.3. [日本語エラーメッセージを削除した NEXTOR.SYS](#)

注意: Nextor 2.1.0 beta 2 以降、NEXTOR.SYS が見つからない場合、カーネルは MSXDOS2.SYS のロードを試みます。ただし、この場合、Nextor のコマンドラインツールは動作しません。

注意: Nextor 2.1.0 beta 2 以降、BASIC で **CALL SYSTEM2 命令** を使用することで、NEXTOR.SYS が存在する場合でも、MSXDOS2.SYS を使用して DOS 環境に強制的にリブートできます。

MSX-DOS 1 プロンプトで起動するには、通常の MSXDOS.SYSファイルと COMMAND.COMファイルが必要です。また、カーネルのみでNEXTOR.SYSやMSXDOS.SYSファイルがない場合、NextorはBASICプロンプトで起動します (AUTOEXEC.BASが存在する場合はそれを実行します)。

したがって、Nextorを「インストール」するには、次の2つの方法があります。

1. 適切なNextorドライバが入ったROMをストレージデバイスコントローラに直接書き込む。
2. スタンドアロン版をフラッシュROMカートリッジに書き込み、別のスロットに挿入したストレージデバイスコントローラと組み合わせて使用する。

また、BASICの使用だけで満足できない場合は、少なくともNEXTOR.SYSとCOMMAND2.COMをブートデバイスにコピーする必要があります (関連ユーティリティも用意しておくことをお勧めします)。ブート手順の詳細は後述します。

3.1.1. Sunrise IDE/CF ユーザへの注意

Sunrise IDEカートリッジまたはSunrise CFリーダーカートリッジにNextorカーネルROMを書き込む場合、IDELOAD.COMプログラムは使用できません。このプログラムは、書き込むファイルのサイズを64KBと想定しており、Nextorカーネルはそれよりも大きいためです（現在のバージョンではドライバーを含めて128KB）。代わりに、128KBのファイルを書き込むように修正されたIDEFL128.COMを使用してください。このプログラムは[Konamimanのウェブサイト](#)からも入手できます。

また、Nextor 2.0に付属のSunrise IDEドライバーは試験的なドライバーであり、いくつかの制限事項があります。具体的には以下のとおりです。

- LBAモードのブロックデバイスのみがサポートされており、CHSまたはATAPIデバイスはサポートされていません。
- デバイスはドライバーによって固定デバイスとして報告されます。CF カードリーダーを使用しており、カードを交換する場合は、コンピューターの電源をオフにする必要があります。カードのホットスワップはサポートされていません。

3.2. Nextorのブート

Nextor のブート手順は MSX-DOS 2 のものと似ています。ただし、Nextor デバイスベースドライバーが存在する場合は、Nextor ドライバーに接続されたすべてのドライブに対して、ドライブとデバイス、およびパーティションの割り当てを実行する必要があるため、手順が少し異なります（デバイスベースドライバーが接続された Nextor カーネルを使用していない場合は、ブート手順は MSX-DOS 2 と同じです）。

ブート時に、Nextor は利用可能なすべてのデバイスベースドライバーに対してクエリを実行し、これらのドライバーによって制御されているデバイスの数（実際には論理ユニットの数ですが、デバイスは通常 1 つの論理ユニットを持ちます）を調べ、各ドライバーに、ドライバーによって制御されるデバイスの数と同じ数のドライブを割り当てます。ブート時に 5 が押された場合、各ドライバーには 1 つのドライブのみが割り当てられます（[2.9. ブートキー](#) を参照）。

例えば、デバイスベースドライバーが接続された 2 つの Nextor カーネルがあるとします。スロット1のカーネルは1つのデバイスを制御し、スロット2のカーネルは3つのデバイスを制御します。この場合、初期のドライブ割り当ては次のようになります。

A: スロット1のドライバー用
B:, C:, D: スロット2のドライバー用
E:, F: 内蔵ディスクドライブ用

5 を押しながら起動すると、割り当ては次のようになります。

A: スロット1のドライバー用
B: スロット2のドライバー用
C:, D: 内蔵ディスクドライブ用

起動中にSHIFTキーを押した場合、内蔵ディスクドライブにはドライブが接続されません（[2.9. ブートキー](#) を参照）。

すべてのドライブにドライバーが割り当てられると、各ドライブに対してデバイスとパーティションからドライブへの自動割り当て処理が実行されます。各ドライブは、以下の条件を満たす最初のパーティションに割り当てられます。

1. デバイスに「自動割当しない」フラグが設定されていない（このフラグはドライバーによって設定されます）
2. 有効な FAT12 または FAT16 パーティションである（MSX-DOS 1 モードで起動する場合は FAT12 のみ）
3. パーティションテーブルに「アクティブ」フラグが設定されている（パーティションテーブルエントリの先頭バイトの最上位ビット。FDISK を使用して設定できます）
4. 同じデバイス内のパーティションに既に割り当てられているドライブがない

3つの条件すべてを満たすパーティションが見つからない場合、検索を最初からやり直しますが、今回は「アクティブ」フラグのチェックはスキップされます。それでも見つからない場合は、デバイスの物理セクター 0 がチェックされます（デバイスにパーティションは存在しないが、有効な FAT ファイルシステムが存在するかを確認するため）。最終的に有効なボリュームが存在しない場合、該当デバイスへのドライブの割り当ては行われません。

ブート処理を高速化するため、この処理では各デバイスの最初の9つのパーティションのみがスキャンされます。そのため、FDISK では、これらの最初の9つのパーティションの「アクティブ」フラグのみを変更できます。

Nextor 2.0.5以降、デバイスベースドライバーは、起動時に必要なドライブ数と、これらのドライバーに割り当てるデバイスをNextorに指定できるようになりました。これにより、この自動処理の一部が省略されます（パーティションは引き続き自動的に選択されます）。この機能はオプションであり、ドライバー開発者が実装する必要があります。

自動割り当てが完了すると、ブート手順は次の手順に進みます。

1. 「3」キーが押されている場合、システムはBASICプロンプトを表示します。
2. 上記以外の場合、ブートドライブ（割り当て済みの最初のドライブ）にNEXTOR.SYS（またはMSXDOS2.SYS）およびCOMMAND2.COMファイルが存在する場合、AUTOEXEC.BAT（存在する場合）の実行後にDOSプロンプトが表示されます。
3. 上記以外の場合、ブートドライブにMSX-DOS 1またはMSX-DOS 2のブートセクターが存在する場合、MSX-DOSの場合と同様に、そのブートコードが実行されます。まず、キャリーフラグがリセットされたBASIC環境が実行され、次にキャリーフラグがセットされたDOS環境が実行されます。これにより通常の、MSXDOS.SYSと COMMAND.COMが存在する際のロードが実行されます。
4. （訳注：MSXDOS.SYSのロードが出来ないなどで）前の手順に戻る場合、BASIC環境が起動され、AUTOEXEC.BASが存在する場合は実行されます。

ディスクに標準ブートセクター（MSX-DOS 1 または MSX-DOS 2 以外で作成）がある場合は、手順 3 は実行されません。内蔵のディスクパーティションツールは、32MB 以下のパーティションにはすべて MSX-DOS 2 ブートセクターを作成し、それより大きいパーティションには標準ブートセクターを作成します。

Nextor 2.1.0 beta 2 以降、Nextor カーネルは NEXTOR.SYS が見つからない場合でも MSXDOS2.SYS があればそれをロードします。これにより、古い MSX-DOS 2 ディスクからの起動が可能になります。ただし、この場合、Nextor のコマンドラインツールは動作しませんのでご注意ください。

3.2.1. DOS1モードのブート

Nextor カーネルは MSX-DOS 1 モードでの起動が可能です。これは、以下の条件のいずれかが満たされた場合となります。

- コンピュータにマップパーRAMがないか、最大メモリが 128K 未満である。
- ブートドライブに MSX-DOS 1 ブートセクターがある（標準フォーマットまたは MSX-DOS 2 フォーマットでないブートセクターは MSX-DOS 1 ブートセクターとみなされます）。
- 起動中に「1」キーまたは「2」キーが押されたままである。

MSX-DOS 1 モードのブート手順は、通常モード（MSX-DOS 2 互換モード）と同じですが、以下の違いがあります。

- 自動割り当て手順では、MSX-DOS 1 互換パーティションのみが検査されます。これらは、FAT あたり 3 セクター以下の FAT12 パーティションです。
- 自動割り当て手順後、NEXTOR.SYSとCOMMAND2.COMの検索手順は省略されます。

内蔵ディスクパーティションツールで作成された16MB以下のパーティションは、FATあたり3セクター以下となるため、MSX-DOS 1モードで使用できます。

MSX-DOS 1では、コンピュータに64KBのRAMが搭載されている場合、DOS環境（MSXDOS.SYS と COMMAND.COM）を起動できます。それ以外の場合は、Disk BASICのみ使用できます。

MSX turboRコンピュータでは、起動中に2キーが押されない限り、MSX-DOS 1モードで起動するとCPUモードがZ80に切り替わります（[2.9. ブートキー](#) を参照）。

注：MSX-DOS 1モードでBASICプロンプトから直接起動する場合は、CALL SYSTEMの前に「POKE &HF346,1」を実行する必要はなくなりました。

3.3. メディアの交換管理

MSX-DOS は、デバイスからデータを読み書きする前に、デバイスドライバーにメディア交換が行われたかどうかを問い合わせ、アクセス先のファイルシステムに関する内部情報を更新します。Nextor も同様の処理を行いますが、アクセス先のドライブがデバイスベースドライバーに割り当てられている場合は、ディスクのパーティション分割が関係するため、処理が少し複雑になります。

Nextor がリムーバブルデバイスベースドライバーに割り当てられたドライブでメディアの交換を検出すると、以下の手順が実行されます。

- ドライブは、デバイス上で最初に見つかった有効なプライマリパーティションに割り当てられます。有効なパーティションは FAT12 および FAT16 です。デバイスにパーティションテーブルがない場合、ドライブは物理セクター 0 に割り当てられます。
- 同じデバイス内で、パーティションに割り当てられていた他のすべてのドライブは、割り当てが解除されます。

デバイスドライバーは、デバイス交換状態を尋ねられた際に「not sure」と応答する場合もあります。その場合の手順は以下のとおりです。

- Nextor は、デバイスベースドライバー上のデバイスに割り当てられたドライブのブートセクターを最初に読み取る際に、ブートセクターの内容の 16 ビットチェックサムを計算し、残りのディスクパラメータと共に保存します。
- Nextor がドライバーにデバイス交換状態を問い合わせ、その応答が「not sure」だった場合、ドライブのブートセクターを再度読み取り、チェックサムを再度計算します。前回保存したチェックサムと一致する場合、Nextor はデバイスが交換されていないと見なします。一致しない場合は、デバイスが交換されたと見なして、ドライバーがデバイス交換を報告したときと同じ割り当て手順を実行します。

不要なメディアチェック（および不要なブートセクターの読み取りとチェックサム計算）を回避するため、リムーバブルデバイスに割り当てられたドライブをロックすることをお勧めします。

3.3.1. DOS1モードのメディア交換

Nextor が MSX-DOS 1 モードで動作している場合、デバイスベースドライバーに割り当てられたドライブではメディアの交換は管理されません。これらのドライブの場合、Nextor はメディアが交換されないものと想定し、ドライバーに交換状態情報を要求することはありません。メディアが交換された場合は、BASIC 上で CALL MAPDRV 命令を実行して、Nextor に手動で交換を通知する必要があります。

3.4. コマンドラインツール

Nextor には、新機能を管理するためのツールセットが付属しています。これらのツールはすべて、DOS プロンプトから実行するための .COM ファイルです。

この章では、これらのツールの使い方について説明します。ただし、各ツールをパラメータなしで起動すると、そのパラメータの概要が表示されます。TYPE コマンドを使用して目的のファイルを直接表示することで、より詳細なヘルプも利用できます (例: TYPE MAPDRV.COM)。

すべてのツールは、Nextor が提供する新しいファンクションコールで動作します。開発者の方で詳細を知りたい場合は、[Nextor 2.1 プログラマーズリファレンス](#) ドキュメントを参照してください。

これらのツールはいずれも MSX-DOS 1 モードでは動作しません。ただし、ほとんどのツールに同等の機能を提供する、同等の BASIC CALL コマンドがあります。

一部のツールでは、`<driver slot>` パラメータを使用できます。これらすべてにおいて、スロット番号の代わりに「プライマリ コントローラ」を意味する 0 番を指定できます。

3.4.1. MAPDRV: ドライブマッピングツール

MAPDRV.COM は、Nextor デバイスベースドライバーによって制御されるデバイス上のパーティションにドライブレターを割り当てを行うツールです。最初は割り当てられていないドライブや、MSX-DOS ドライバーまたは Nextor ドライブベースドライバーに関連付けられているドライブも含め、あらゆるドライブを割り当て可能です。

MAPDRV の使用構文は以下のとおりです。

```
MAPDRV [/L] <drive>: <partition>|d|u [<device index>-[<LUN index>]  
[<driver slot>[<driver subslot>]]]
```


パーティション番号 1 は、デバイス上の最初のプライマリパーティションを参照します。パーティション 2 ～ 4 は、デバイスのパーティション 2 が拡張されている場合は拡張パーティション 2-1 ～ 2-4 を参照し、そうでない場合はプライマリパーティション 2 ～ 4 を参照します。パーティション 5 以降は常に拡張パーティション 2-(P-1) を参照します。

パーティション番号 0 を指定した場合、ドライブはデバイスの物理セクター 0 に割り当てられます。

パーティションが配置されているデバイスを指定するには、次の 3 つの方法があります。

koumoku* パーティション番号の後にパラメータを指定しない。この場合、パーティションはドライブに既に割り当てられているデバイスと同じデバイス内にあると想定されます（これは、ドライブが現在デバイスベースドライバーに割り当てられている場合にのみ機能します）。

- デバイスインデックスを指定し、ドライバースロット番号は指定しない。この場合、パーティションは指定されたデバイス内にあると想定され、デバイスは現在割り当てられているデバイスと同じスロット上のカーネルによって制御されていると想定されます（これは、ドライブが現在デバイスベースドライバーにマップされている場合にのみ機能します）。
- デバイスインデックスとドライバースロット番号を指定する。この場合、スロットは、デバイス进行处理するドライバーを含む Nextor カーネルに対応します。

デバイス番号を指定する場合、論理ユニット番号も指定できます。論理ユニット番号のデフォルト値は 1 です。

パーティション番号の代わりに「d」を指定した場合、ドライブはデフォルトの状態に割り当てられます。デフォルトの状態は以下のいずれかです。

- 起動時にドライブが割り当てられていない場合は、割り当てられないままになります。
- 起動時にドライブが MSX-DOS ドライバーユニットまたは Nextor ドライブベースユニットに割り当てられていた場合は、同じユニットに割り当てられます。
- 起動時にドライブが Nextor デバイスベースドライバーに割り当てられていた場合は、自動割り当て手順（起動時に実行されるものと同じですが、「アクティブ」パーティションフラグがチェックされない点が異なります）が実行されます。この結果、他のドライブの割り当て状態に応じて、ドライブが起動時と同じ割り当てになる場合とならない場合があります。

パーティション番号の代わりに「u」を指定した場合、ドライブの割り当ては解除となります。

オプションパラメータ「/L」は、割り当て後すぐにドライブをロックします（変更されないリムーバブルデバイスに推奨されます）。

Nextor 2.1以降では、MAPDRVツールを使用してディスクイメージファイルをドライブにマウントすることもできます。この場合の構文は以下のとおりです。

```
MAPDRV <drive> <file> [/ro]
```

/ro パラメータを指定すると、ファイルは読み取り専用モードでマウントされます。ただし、ファイルに読み取り専用属性が設定されている場合は、/ro パラメータが指定されていなくても、常に読み取り専用モー

ドでマウントされます。

ファイルをドライブにマウントする際には、いくつかの制限があります。詳細は [3.8. ファイルのマウント](#) を参照してください。

3.4.2. DRIVERS: ドライバー情報ツール

DRIVERS.COMユーティリティは、パラメータなしで実行すると、利用可能な MSX-DOS および Nextor ドライバーに関する情報を表示します。ドライバーの名前とバージョン（Nextor ドライバーのみ）、スロット番号、起動時に割り当てられたドライブが表示されます。MSX-DOS ドライバーは「Legacy driver」（レガシードライバー）と識別されます。

このツールは主に、ドライバーのスロット番号を取得し、他のツールにパラメータとして渡すために役立ちます。

3.4.3. DEVINFO: デバイス情報ツール

DEVINFO.COMユーティリティは、指定された Nextor デバイスベース ドライバーによって制御されるデバイスに関する情報を表示します。表示される情報には、デバイス名と製造元（利用可能な場合）、デバイス インデックス、および関連する論理ユニットの種類とサイズが含まれます。

DEVINFO の使用構文は以下のとおりです。

```
DEVINFO <driver slot>[-<driver subslot>]
```

このツールは主に、デバイスと論理ユニットのインデックスを取得し、それらを MAPDRV ツールにパラメータとして提供するのに役立ちます。

3.4.4. DRVINFO: ドライブ情報ツール

DRVINFO.COMユーティリティはパラメータなしで実行され、利用可能なすべてのドライブレター（割り当て済み）に関する情報を表示します。表示される情報には、関連付けられたドライバースロットと、関連付けられたドライバーの種類に応じたその他の情報（Nextor ドライバーの場合はドライバー名とバージョン、Nextor デバイスベースドライバーの場合はデバイス番号と論理ユニット番号、MSX-DOS および Nextor ドライブベースドライバーの場合は相対ユニット番号）が含まれます。MSX-DOS ドライバーは「Legacy driver」（レガシードライバー）と識別されます。

3.4.5. LOCK: ドライブのロック/アンロックツール

LOCK.COMユーティリティは、ドライブレターのロックとロック解除を行います。LOCKの使用構文は以下のとおりです。

```
LOCK [<drive letter>: [ON|OFF]]
```

パラメータなしで実行すると、現在ロックされているドライブレターの一覧が表示されます。ドライブレターのみを指定した場合は、ドライブの現在のロック状態が表示されます。

ドライブがロック中とマークされている場合、Nextor はドライブのメディア交換状態を確認しません。代わりに、挿入されたメディアは交換されないものとみなされます。これによりメディアへのアクセス速度は向上しますが、ロックされている間にメディアを交換するとデータが破損する可能性があるため注意が必要です。

ディスクエラーによる中止(Abort)が発生すると、関連するドライブのロックが自動的に解除されます。それ以外では、ドライブのロックは LOCK ユーティリティを OFF パラメータ付きで実行した場合にのみ解除されます。Nextor はドライブを自動的にロックすることはありません。

3.4.6. RALLOC: 割当情報縮小/ゼロモードツール

RALLOC.COMユーティリティは、ドライブの割当情報縮小モードを有効または無効にすることができます。RALLOCの使用構文は以下のとおりです。

```
RALLOC [<drive letter>: ON|OFF]
```

パラメータが指定されていない場合は、現在割当情報縮小モードになっているドライブの一覧が表示されません。

ドライブがこのモードになっている場合、ドライブの合計容量と空き容量に関する情報を返す ALLOC 関数は、計算された合計容量または空きセクター数が常に 16 ビットに収まるように、必要に応じて偽の情報を返します。つまり、割当情報縮小モードが有効なドライブでは、合計容量または空き容量が 32MB を超える場合（FAT16 ボリュームではこの可能性があります）、ALLOC は 32MB を返します。

環境変数 ZALLOC が存在し、その値が ON になっている場合（コマンドプロンプトでコマンド **SET ZALLOC=ON** を実行）、割当情報縮小モードは割当情報ゼロモードになります（Nextor 2.0.3 以降で可能）。ALLOC 関数は、このモードが有効なドライブの空き容量としてゼロを返します。これにより関数は即座に制御を戻すため、非常に大容量または低速なデバイスで役立ちます。

Nextor はドライブの割当情報縮小モードのステータスを自動的に変更することはありません。この動作は常にユーザーが制御します。ディスクエラーやメディアの交換があっても、割当情報縮小モードのステータスは変更されません。

3.4.7. Z80MODE: Z80アクセスモードツール

MSX turboR コンピュータでのみ動作するZ80MODE.COMユーティリティは、MSX-DOSドライバーのZ80アクセスモードを有効または無効にします。Z80MODEの使用法は以下のとおりです。

```
Z80MODE <driver slot>[-<driver subslot>]] [ON|OFF]
```

ドライバースロットのみを指定した場合、ドライバーの現在の Z80 アクセスモードの状態が表示されます。Z80 アクセスモードはドライバーごとに設定または解除されます（特定のドライブレターに対して変更することはできません）。

Z80 アクセスモードは、MSX-DOS ドライバーに対してのみ設定または解除できます（Nextor は、Nextor ドライバーにアクセスする際に現在の CPU を切り替えることはありません）。有効に設定されている場合、

Nextor はドライバーで何らかの操作を実行する前に、現在の CPU を Z80 に切り替えます。無効に設定されている場合、Nextor はドライバーにアクセスする際に現在の CPU を変更しません。

特定の MSX-DOS ドライバーで Z80 アクセスモードの設定が必要かどうかは、ドライバーによって異なります。不明な場合は、ドライバーのドキュメントを参照するか、可能であればドライバーの開発者に問い合わせてください。フロッピーディスクドライブでは、Z80 アクセスモードを有効にする必要がある可能性があります。

Nextor は起動時にすべての MSX-DOS ドライバーに対して Z80 アクセスモードを有効にします。それ以外では、Nextor はどのドライバーに対しても Z80 アクセスモードを自動的に変更することはなく、この動作を常に制御するのはユーザーです。

3.4.8. FASTOUT: 高速STROUTモードツール

FASTOUT.COMユーティリティを使用すると、高速STROUTモードのオン/オフを切り替えることができます。FASTOUTの使用構文は以下のとおりです。

```
FASTOUT [ON|OFF]
```

パラメータなしで呼び出すと、FASTOUT モードの現在の状態が表示されます。

MSX-DOS のファンクションコール STROUT は、"\$" 文字で終了する文字列を出力します。このファンクションコールは実際には、文字列の先頭から順に、CONOUT ファンクションコール（1文字出力）を個別に呼び出します。

高速 STROUT モードが有効な場合、文字列はページ 3 の 512 バイトのバッファにコピーされ、カーネルコードへの 1 回の呼び出しで出力されるため、出力処理が高速化されます。ただし、このモードが有効な場合、文字列の長さが 511 バイトに制限され、それより長い文字列は切り捨てられ、最初の 511 文字のみが表示されます。

3.4.9. DELALL: パーティションクイックフォーマットツール

DELALL.COMユーティリティは、指定されたドライブレータで表示されるファイルシステムをクイックフォーマットします。DELALLの使用構文は次のとおりです。

```
DELALL <drive letter>:
```

このツールは、ファイルシステムのFATおよびルートディレクトリ領域をクリーンアップし、ファイルシステム上のすべての情報を実質的に削除します。この操作を元に戻すことはできません。ファイルは永久に失われるため、慎重に使用してください。

このツールは、MSX-DOSドライバーに接続されているドライブも含め、あらゆるドライブで使用できます。ただし、ドライブに有効なFAT12またはFAT16ファイルシステムが割り当てられている必要があります。そうでない場合、このツールは動作しません。

3.4.10. NSYSVER: NEXTOR.SYSバージョンチェンジャー

一部のMSX-DOSコマンドラインアプリケーションは、MSXDOS2.SYS（Nextorの場合はNEXTOR.SYS）のバージョン番号をチェックし、その番号が特定の値（通常は2.20）より小さい場合は動作を停止することが知られています。これは、現在のNEXTOR.SYSのバージョン番号が2.1であるため、問題となります。

この問題の回避方法として、バージョン2.0 beta 2 以降では、DOSVERファンクションコールによって返されるNEXTOR.SYSのバージョン番号がRAMに保存され、簡単に変更できるようになりました（詳細については、[Nextor 2.1 プログラマーズリファレンス](#) ドキュメントを参照してください）。この変更を簡単に行うことができるコマンドラインツールも開発されており、NSYSVER.COMという名前で、以下のように使用できます。

```
NSYSVER <major version number>.<secondary version number>
```

例: **NSYSVER 2.20**。これは、DOSVER ファンクションコールによって返されるNEXTOR.SYSのバージョン番号の値のみを変更することに注意してください。VERコマンドは、実際のファイルバージョン番号を引き続き表示します。

注: このツールによるバージョン番号の変更は一時的なものであり、CALL SYSTEMによるBASICプロンプトの起動と終了、またはコンピュータの再起動によってNEXTOR.SYSが再ロードされると、変更は無効になります（つまり、NEXTOR.SYSのバージョン番号は実際の値に戻ります）。

注: このツールは、NEXTOR.SYSのバージョン2.0 beta 2 より古いバージョンでは使用しないでください。

3.4.11. NEXBOOT: ワンタイムブートキー設定ツール

NEXBOOT.COMツールを使えば、次のリセット時に**ワンタイムブートキー**として使用するキーを簡単に設定できます。構文は以下のとおりです。

```
NEXBOOT <boot keys>|. [*|<slot> [<slot>... ]]
```

ここで、ブートキーは数字キー、C は CTRL キー、S は SHIFT キーです。**<slot>** は無効化する Nextor カーネルのスロット番号です。例えば、**NEXBOOT 1C** は Ctrl キーと 1 キーを反転し、**NEXBOOT S 1 23** は SHIFT キーを反転し、スロット 1 と 2-3 の Nextor カーネルを無効化します。**NEXBOOT . 2** はスロット 2 の Nextor カーネルのみを無効化します。

NEXBOOT.COM バージョン 1.1 以降を使用している場合は、***** を指定してすべての Nextor カーネルを無効にすることもできます。これは、起動中に **N** キーを押すのと同じです。ただし、これはバージョン 2.1 以降の Nextor カーネルでのみ有効です。

いずれの場合も、ツールは RAM にキー情報を適切に設定した後、直ちにコンピューターをリセットします。

3.4.12. EMUFILE: ディスクエミュレーションモードツール

EMUFILE.COMツールは、ディスクエミュレーションモードのデータファイルを作成し、ディスクエミュレーションモードに入ることを可能にします。エミュレーションデータファイルを作成するための構文は以下のとおりです。


```
emufile [<options>] <output file> <files> [<files> ...]
```

<output file> は作成されるエミュレーションデータファイルの名前（デフォルトの拡張子は .EMU）で、<files> はエミュレーションに使用するディスクイメージファイルです（ワイルドカードを使用できます）。ディスクイメージファイルには、指定された順序で番号（ディスク変更用）が割り当てられます。ワイルドカードを使用する場合は、ワイルドカードを含むストレージデバイス内での検出順序（DIR コマンド実行時に表示される順序と同じ）で番号が割り当てられます。

-b <number> オプションを使用すると、エミュレーションセッション開始時に起動に使用するディスクイメージファイルの番号を指定できます。デフォルトは 1 です。

-a <address> オプションを使用すると、エミュレーションセッション中に Nextor が作業領域（約 16 バイト）として使用するページ 3 のアドレスを指定できます。アドレスはページ 3 内の 16 進数（C000 以上）で指定する必要があります。指定しない場合、この領域はエミュレーションセッション開始前に Nextor によって割り当てられます。

-p オプションは、データファイルの作成後、すべてのファイル名と関連キーを出力します。ただし、エミュレーションデータファイルを TYPE /B で実行すると、後で同じ情報を確認できます。

ディスク エミュレーション セッションを開始するための構文は次のとおりです。

```
emufile set <data file> [o|p[<device index>[<LUN index>]]]
```

o はワンタイム方式（デフォルト）を使用してエミュレーションを開始し、p は永続方式を使用してエミュレーションを開始します。後者の場合、デフォルトではエミュレーションファイルのデータポインタは <data file> が保存されているデバイスに書き込まれますが、別の <device index> を指定したり、オプションで <LUN index> を指定したりすることもできます。デフォルトの LUN index は 1 です（つまり、p3 は p31 と同じです）。

どちらの方式でも、EMUFILE.COM がエミュレーションデータファイルポインタを適切な場所に書き込んだ直後にコンピューターがリセットされることに注意してください。

3.5. 組込パーティション設定ツール

Nextor カーネルには、Nextor デバイスベースドライバーに接続されたストレージデバイスをパーティション分割するためのユーティリティが組み込まれています。起動するには、BASIC プロンプトから CALL FDISK を呼び出すだけです。40 桁モードと 80 桁モードの両方で正常に動作します。FDISK ツールを起動すると、現在の BASIC プログラムがメモリから削除されることに注意してください。

このツールはメニュー形式のユーザーインターフェースを備えているため、画面に表示される指示に従うだけで誰でも使用できます（不明な場合は、画面下部に表示される次の操作方法を確認してください）。ただし、ツール自体には記載されていない、考慮すべき点がいくつかあります。

- このツールを使用すると、Nextor デバイスベースドライバーに接続された任意のブロックデバイス上に、最大 256 個の FAT12 および FAT16 パーティションを作成できます。MSX-DOS ドライバーおよび Nextor ドライブベースドライバーはサポートされていません。

- このツールでは、既にパーティション分割されているデバイスに新しいパーティションを追加することはできません。新しいパーティションを定義する前に、既存のパーティションをすべて削除する必要があります。
- 100KB（サポートされる最小パーティションサイズ）から32MBまでのパーティションはFAT12、33MBから4GB（サポートされる最大パーティションサイズ）までのパーティションはFAT16になります。
- 16MB以下のパーティションはFATあたり3セクター以下となるため、MSX-DOS 1モードで使用できません。
- 32MBまでのパーティションにはMSX-DOS 2ブートセクターが、33MB以上のパーティションには標準ブートセクターが設定されます。
- 最適なクラスタサイズを得るには、パーティションサイズを2の累乗で定義することをお勧めします（FAT12パーティションの場合は1MB、2MB、4MB、8MB、16MB、32MB、FAT16パーティションの場合は64MB、128MB、256MB、512MB、1GB、2GB、4GB）。これが不可能な場合は、最も近い2の累乗より少し大きいサイズよりも、少し小さいサイズを選択することをお勧めします（例えば、33MBより31MBの方が適しています）。

Nextorは、FAT16パーティションと標準ブートセクタを持つデバイスに対応しています。工場出荷時にパーティションが設定されている2GB以下のデバイスを使用する場合は、MSX-DOS 1互換パーティションを作成する場合を除き、パーティションを設定する必要はありません（4GBデバイスは通常FAT32パーティションで出荷されるため、いずれにしてもFDISKでパーティションを設定する必要があります）。

新しいパーティションを作成する際に、どのパーティションに「アクティブ」フラグを設定するかを選択できます。これにより、起動時に自動的にマッピングされるようになります（[3.2. Nextorのブート](#)を参照）。また、既存のパーティションのフラグを変更することも可能です。

パーティション作成ツールはMSX-DOS 1モードでも動作します。ただし、このツールでは常に16MBを超えるパーティションを作成でき、これはMSX-DOS 1と互換性がないのでご注意ください。

3.6. 拡張 Disk BASIC

Nextorは、Disk BASICにいくつかの新しいコマンドを追加し、主にこの環境からのデバイスとパーティションの管理を容易にします。また、MSX-DOSに既に存在していたコマンドの一部も拡張または改良されています。

新しいCALL命令の中にはパラメータを取るものがあります。これらの命令はパラメータなしで実行することで、使用方法に関するヘルプを表示できます。

特に明記されていない限り、既存のDisk BASICに対するNextorによる変更はMSX-DOS 1モードでは使用できませんが、新しい命令は使用できます。

3.6.1. DSKF 関数

ドライブの空き容量を返すDSKF関数は、MSX-DOSでは空きクラスタ数を返します。Nextorでは、このコマンドの動作が変更され、空きKB数を返すようになりました。

この動作はMSX-DOSからの互換性を著しく損なう変更点です。しかし、この関数を使用する既存のプログラムのほとんどは、実際には空き容量をKB単位で計算せず、代わりに生のクラスタ数をユーザーに表示します。また、長年にわたり、MSXコンピュータで最も普及していたストレージメディアはクラスタサイズが

1KB の 2DD フロッピーディスクであったため、多くのユーザーが DSKF コマンドが KB 数を返すものと誤って認識していました。

この変更は MSX-DOS 1 モードには適用されません。このモードでは、空きクラスタ数は引き続きクラスタ数として返されます。

DSKF コマンドは、ドライブで縮小割り当て情報モードがアクティブになっている場合でも、常に実際の空き容量を返します。ただし、ドライブで割り当て情報ゼロモードが有効になっている場合は、返される値は 0 になります。

3.6.2. DSKI\$ 関数/DSKO\$ 命令

ディスクセクター 1 つ分の読み取りと書き込みを実行する DSKI\$ 関数と DSKO\$ 命令は、32 ビットのセクター番号に対応しました。これにより、最初の 65536 セクターだけでなく、任意のドライブセクターにアクセスできるようになりました。

32767 を超えるセクター番号を持つセクターにアクセスするには、セクター番号を単精度または倍精度の定数、式、または変数で指定する必要があります。単精度の値を指定し、その数値が大きすぎて切り捨てにより最下位桁の 1 桁以上が失われる場合、これらのコマンドは「オーバーフロー」エラーで失敗します。これは、誤ったセクターへの誤ったアクセスを防ぐためです。例：

```
10 DEFSNG S
20 S=12345678
30 PRINT S 'Prints "12345700"
40 PRINT DSKI$(0, S) 'Throws "Overflow"
```

上記の例は、10 行目を DEFDBL S に変更すれば動作します（ただし、デバイス内にセクターが存在する必要があります）。BASIC コード内で任意のセクター番号にアクセスする場合は、必ず倍精度変数を使用してください。

指定されたセクター番号が 32 ビットに収まらない場合、つまり 4,294,967,295 より大きい場合も、「オーバーフロー」エラーが発生します。

MSX-DOS の同等の命令/関数との互換性を維持するため、負のセクター番号（実際のセクター番号を取得するには 65536 を加算します）も受け付けますが、セクター番号が整数式（16 ビット）として評価できる場合に限りです。したがって、以下の関数は同等であり、デバイス内にセクターが存在する場合に動作します。

```
PRINT DSKI$(0, 65535)
PRINT DSKI$(0, &HFFFF)
PRINT DSKI$(0, -1)
DEFINT S: S=-1: PRINT DSKI$(0, S)
```

ただし、以下の場合は「ディスク I/O エラー」が発生します。（訳注：16 ビット整数式ではないため）

```
PRINT DSKI$(0, CDBL(-1))
DEFDBL S: S=-1: PRINT DSKI$(0, S)
```

これらはいずれも MSX-DOS 1 モードには適用されません。このモードでは整数 (16 ビット) セクター番号のみが受け入れられます。

3.6.3 CALL NEXTOR 命令

この命令は、Nextor が BASIC 環境に提供する新しい CALL 命令のリストを表示します。

3.6.4 CALL CHDRV 命令

この命令はカレントドライブを変更します。これはMSX-DOS 2 Disk BASICに既に存在します。しかし、Nextorでは以下の2つの方法で拡張されています。

- この命令はMSX-DOS 1モードでも使用できるようになりました。
- ドライブ番号は、ドライブレターの代わりに数字も指定できます（1はA:、8はH:）。例えば、`_CHDRV(3)`は`_CHDRV("C:")`と同じです。

3.6.5 CALL CURDRV 命令

この命令は、カレントドライブを表示します。

3.6.6. CALL DRIVERS 命令

この命令は、利用可能なMSX-DOSおよびNextorドライバーの情報を表示するDRIVERS.COMツールと同等です。名前とバージョン（Nextorドライバーのみ）、スロット番号、起動時に割り当てられたドライブが表示されます。MSX-DOSドライバーは「Legacy driver」（レガシードライバー）と識別されます。

3.6.7. CALL DRVINFO 命令

この命令は、利用可能なすべてのドライブレター（割り当て済み）に関する情報を表示するDRVINFO.COMユーティリティと同等です。表示される情報には、関連付けられたドライバースロットと、関連付けられたドライバーの種類に応じたその他の情報（Nextorドライバーの場合はドライバー名とバージョン、Nextorデバイスベースドライバーの場合はデバイス番号と論理ユニット番号、MSX-DOSおよびNextorドライブベースドライバーの場合は相対ユニット）が含まれます。MSX-DOSドライバーは「Legacy driver」（レガシードライバー）と識別されます。

3.6.8. CALL LOCKDRV 命令

この命令はドライブのロックとアンロックを行います（[2.4. ドライブのロック](#)および[3.4.5. LOCK: ドライブのロック/アンロックツール](#)を参照）。この命令は以下のように使用します。

```
CALL LOCKDRV(<drive>)
```

ドライブの現在のロック状態を表示します。

```
CALL LOCKDRV(<drive>, 0)
```

ドライブのアンロックを行います。

```
CALL LOCKDRV(<drive>, <any non-0 number>)
```

ドライブをロックします。

<drive> は、ドライブ文字の後にコロン (📁) が付いた文字列 (例: "A:") または数字 (A: ドライブから H: ドライブの場合は 1 から 8、現在のドライブの場合は 0) です。

このコマンドは、ドライブロックの概念が存在しない MSX-DOS 1 モードでは使用できません。

3.6.9. CALL MAPDRV 命令

この命令は、BASIC環境からドライブとデバイス、およびパーティションの割り当てを変更できます。MAPDRV.COMツールと同等です。

CALL MAPDRVの構文については以下で説明します。一部のパラメータはオプションであるため、最も完全な（すべてのパラメータを使用）ものから順に、考えられるすべてのバリエーションについて説明します。各パラメータの可能な値の詳細については、後述します。

```
CALL MAPDRV(<drive>, <partition>, <device>, <slot>|0)
```

指定されたドライブに、スロットのドライバーによって制御されるデバイスのパーティションを割り当てます。スロット番号の代わりに0を指定した場合、プライマリコントローラのスロットが使用されます。

```
CALL MAPDRV(<drive>, <partition>, <device>)
```

指定されたドライブに、デバイスのパーティションを割り当てます。ドライバースロットは、ドライブに既に割り当てられているパーティションを含むデバイスと同じであると想定されます。ドライブが現在デバイススペースドライバーに割り当てられていない場合は、「Invalid device driver」（無効なデバイスドライバー）エラーが発生します。

```
CALL MAPDRV(<drive>, <partition>)
```

指定されたドライブに、パーティションを割り当てます。デバイスは、ドライブに既に割り当てられているパーティションを含むデバイスと同じであると想定されます。ドライブが現在デバイススペースドライバーに割り当てられていない場合は、「Invalid device driver」（無効なデバイスドライバー）エラーが発生します。

```
CALL MAPDRV(<drive>, -1)
```

指定されたドライブの割り当てを解除します。このドライブに再度アクセスしようとする時、「Bad drive name」（ドライブ名不正）（MSX-DOS 1 モードでは「Disk I/O error」）エラーが発生します。

```
CALL MAPDRV(<drive>, -2)
CALL MAPDRV(<drive>)
```

指定されたドライブにデフォルト値を割り当てます。起動時にドライブが割り当てられていない場合、またはMSX-DOSドライバーもしくはNextorドライブベースドライバーにマッピングされていた場合、ドライブは元の割り当て状態に戻ります。それ以外の場合は、自動割り当て手順が実行されます（この手順は、アクティブパーティションフラグがチェックされない点を除けば、起動時に実行される手順と同じです。詳細は[3.2. Nextorのブート](#)を参照してください）。この結果、利用可能なデバイスや他のドライブの割り当て状況に応じて、ドライブが起動時と同じ割り当て状態になる場合とならない場合があります。

コマンドパラメータの構文は以下のとおりです。

- **<drive>** は、ドライブ文字にコロン（例："A:"）または数字（ドライブA:~Hの場合は1~8、現在のドライブの場合は0）の文字列です。
- **<partition>** は 0~255 の範囲の数値で、以下のように解釈されます。
 - 0: デバイスにパーティションがないものとみなします。ドライブはデバイスの物理セクタ 0 にマッピングされます。
 - 1: デバイスの最初のプライマリパーティション。
 - 2、3、または 4: デバイスパージション 2 が拡張されている場合、この数値はそれぞれ 1 番目、2 番目、または 3 番目の拡張パーティションとして解釈されます。それ以外の場合、この数値はそれぞれデバイスの 2 番目、3 番目、または 4 番目のプライマリパーティションとして解釈されます。
 - 5 以上: この数値はデバイスの (n-1) 番目の拡張パーティションとして解釈されます。
- **<device>** は 1~7 の範囲のデバイスインデックスです。デバイスに複数の論理ユニットがある場合は、式 **<device>+16*<logical unit>** を使用します。論理ユニットに指定できる値も 1~7 です（0 も指定可能で、1 として解釈されます）。
- **<slot>** は 0~3 の範囲のスロット番号です。スロットが拡張されている場合は、**<main slot>+4*<subslot>** という式を使用します。特別なケースとして、スロット番号に 0 を指定し、サブスロット番号を指定しない場合は、プライマリコントローラのスロットが使用されます。

MSX-DOS 1 モードでは、Nextor アーキテクチャによっていくつかの追加制限が課せられます。

- 指定されたドライブは、起動時にデバイスベースドライバーに割り当てられている必要があります。割り当てられていない、または起動時に MSX-DOS ドライバーまたは Nextor ドライブベースドライバーに割り当てられているドライブの割り当てを変更することはできません。
- 新しい割り当て情報では異なるパーティションやデバイスを指定できますが、ドライバースロットは起動時にドライブに割り当てられたスロットと同じである必要があります。システムに Nextor カーネルが 1 つしかない場合は、これは問題になりません。また、MSX-DOS 1 モードでは、ドライブをサポートされていないパーティションタイプ（FAT16 パーティション、または FAT が 3 セクターを超え

る FAT12 パーティション) にマッピングすると、そのドライブにアクセスすると必ず「Disk I/O error」 (ディスク I/O エラー) が発生しますのでご注意ください。これはデバイスに実際に障害があるという意味ではなく、Nextor がアクセスを拒否しているという意味です。

Nextor 2.1 以降では、CALL MAPDRV コマンドを使用してディスクイメージファイルをドライブにマウントすることもできます。この場合の構文は次のとおりです。

```
CALL MAPDRV(<drive>, <file> [,0|1])
```

,1 パラメータを指定すると、ファイルは読み取り専用モードでマウントされます。ただし、ファイルに読み取り専用属性が設定されている場合は、**,1** パラメータが指定されていなくても、常に読み取り専用モードでマウントされます。

ファイルをドライブにマウントする際には、いくつかの制限があります。詳細は [3.8. ファイルのマウント](#) を参照してください。

3.6.10. CALL MAPDRVL 命令

CALL MAPDRVL 命令は CALL MAPDRV コマンドと同じですが、ドライブ割り当てを変更した直後にドライブロック ([2.4. ドライブのロック](#) および [3.4.5. LOCK: ドライブのロック/アンロックツール](#) を参照) を実行する点が異なります。

このコマンドは MSX-DOS 1 モードでは使用できないことに注意してください。MSX-DOS 1 モードには「ドライブのロック」という概念が存在しません。

3.6.11. CALL USR 命令

CALL USR 命令は、BASICコードからアセンブラコードを実行できるようにします。これは、MSX-BASIC標準のDEF USR命令およびUSR関数と同等ですが、実行コードにZ80レジスタへの入力値を指定し、実行後に出力値を読み取る機能が追加されています。

CALL USRコマンドの構文は次のとおりです。

```
CALL USR(<code address> [,<registers address>])
```

<code address> は、実行するアセンブラコードのアドレスです。値 -1 は特別なケースとして扱われます。**_USR(-1)** は何も行いませんが、エラーはスローされません。この機能を ON ERROR GOTO コマンドと組み合わせて使用することで、BASIC プログラム内から Nextor の存在を検出できます。

<registers address> は、Z80 レジスタ値を格納する 12 バイトのバッファのアドレスです。このパラメータが指定されている場合、コードが呼び出される前に、この領域の内容がレジスタにロードされます。コード実行後は、逆の処理が実行され、バッファはレジスタに保持されている値で更新されます。バッファ内のレジスタの順序は、F、A、C、B、E、D、L、H、IXL、IXh、IYL、IYh です。

CALL USR コマンドをテストするための簡単な BASIC プログラムを以下に示します。別のコードを呼び出したい場合は、40 行目から 90 行目のレジスタ割り当てと、100 行目で呼び出されるコードのアドレスを適切

に変更します (MSX BIOS 自体は、試してみるのに適したルーチンのソースです)。

```
10 ON ERROR GOTO 20: _USR(-1): ON ERROR GOTO 0: GOTO 30
20 PRINT "Nextor not found!": END
30 DEFINT R: DIM R(12)
40 R(0)=&H2100 'AF
50 R(1)=&H3040 'BC
60 R(2)=&H5060 'DE
70 R(3)=&H7080 'HL
80 R(4)=&H90A0 'IX
90 R(5)=&HB0C0 'IY
100 CALL USR(&H00A2, VARPTR(R(0))) 'Prints a "!" (passed in A as &H21)
110 PRINT "AF=&H";HEX$(R(0))
120 PRINT "BC=&H";HEX$(R(1))
130 PRINT "DE=&H";HEX$(R(2))
140 PRINT "HL=&H";HEX$(R(3))
150 PRINT "IX=&H";HEX$(R(4))
160 PRINT "IY=&H";HEX$(R(5))
```

3.6.12 CALL SYSTEM2 命令

Nextor 2.1.1 では、新しい **CALL SYSTEM2** コマンドが導入されました。このコマンドは **CALL SYSTEM** と同じように動作しますが、NEXTOR.SYS というファイルが存在する場合でも、常に MSXDOS2.SYS をロードします。MSXDOS2.SYS は NEXTOR.SYS よりもサイズが小さいため、これはアプリケーションの TPA 領域を節約するのに役立ちます。

ただし、Nextor で導入された新しい関数呼び出しは、NEXTOR.SYS がロードされていない場合は動作しません。つまり、**CALL SYSTEM2** コマンドで DOS 環境に入った場合、Nextor 固有のコマンドラインツール (例: **MAPDRV.COM**) は動作しません。

3.7. 新しい BASIC エラーコード

新しい BASIC コマンドで発生する可能性のあるエラーを処理するために、以下の新しい BASIC エラーコードが定義されています。これらのエラーは、MSX-DOS 1 モードでも、この環境で動作するコマンドで使用できます。括弧内の数字はエラーコードです。

- Invalid device driver (無効なデバイスドライバです) (76)。CALL MAPDRV 命令が、以下のいずれかのイベントでスローします。
 - 指定されたスロット番号に Nextor デバイスベースドライバーが含まれていません。
 - スロット番号が指定されていませんが、ドライブは現在 Nextor デバイスベースドライバーにマップされていません。
 - MSX-DOS 1 モードにおいて、ドライブは元々 Nextor デバイスベースドライバーに割り当てられていなかったか、別のドライバーに割り当てられていました。
- Invalid device or LUN (無効なデバイスまたは論理ユニットです) (77)。CALL MAPDRV 命令が、以下のいずれかのイベントでスローします。

- 指定されたインデックスのデバイスまたは LUN は、指定されたドライバーまたは暗黙のドライバーでは利用できません。
- 指定されたインデックスを持つデバイスと論理ユニット番号は、指定されたドライバーまたは暗黙のドライバーに存在しますが、ブロックデバイスではありません。

- Invalid partition number (無効なパーティション番号) (78)

指定されたパーティションが、指定されたデバイスまたは暗黙のデバイス上に存在しない場合、CALL MAPDRV 命令によってこのエラーがスローされます。

- Partition is already in use (パーティションが使用中です) (79)

このエラーは、別のドライブに既にマップされているパーティション、デバイス、およびドライバーの組み合わせを割り当てようとした場合に、CALL MAPDRV 命令によってスローされます。ただし、同じ組み合わせを同じドライブに再度割り当てることは可能です。

- File is mounted (ファイルがマウントされています) (80)

マウントされたファイルを開こうとしたり、変更しようとしたり、あるいはマウントされたファイルに対して許可されていない操作を実行しようとした。

- Bad file size (無効なファイルサイズ) (81)

512バイト未満、または32MBを超えるファイルをマウントしようとした場合に、CALL MAPDRV コマンドによってスローされます。

- Invalid cluster number or sequence (無効なクラスタ番号またはシーケンス) (82)

ホストファイルシステム内の連続したセクタに格納されていないファイルをマウントしようとした場合に、CALL MAPDRV コマンドによってスローされます。

3.8. ファイルのマウント

Nextor 2.1 では、ディスクイメージファイルをドライブレターにマウントする機能が導入されました。ディスクイメージファイルをマウントすると、通常の MSX-DOS/MSX BASIC コマンドとツールを使用して、そのファイルやディレクトリにアクセスできます。

ファイルをマウントするには、[MAPDRV ドライブマッピングツール](#) で `MAPDRV <drive> <file> [/ro]` 構文を使用します。または、BASIC 環境では、[CALL MAPDRV 命令](#) で `CALL MAPDRV(<drive>, <file> [,0|1])` 構文を使用します。ファイルをアンマウントするには、ドライブの割り当てを別のものに変更するか、ドライブを見割り当ての状態にします (`MAPDRV <drive> U` または `CALL MAPDRV(<drive>, -1)`)。

この機能にはいくつかの制限があります。

- ファイルのサイズは512バイト以上32MB以下である必要があります。
- ファイルはホストファイルシステム内の連続したセクタに格納されている必要があります (Nextor 2.1.1以降)。
- ファイルには適切なFATファイルシステムが既に含まれていなければなりません。マウントされたドライブにFORMATコマンドを適用することはできません。

- ファイルにはパーティションを含めることはできません。含まれるファイルシステムはファイルの先頭から始まる必要があります。
- ファイル自体が配置されているドライブにファイルをマウントすることはできません。

```
MAPDRV A: A:TOOLS.DSK --> Error
```

- 同じファイルを同時に 2 つのドライブにマウントすることはできません。

```
MAPDRV B: TOOLS.DSK
MAPDRV C: TOOLS.DSK --> Error
```

- 再帰的なファイルマウント (マウントされたディスク イメージ ファイル内にあるファイルのマウント) は実行できません。

```
MAPDRV B: TOOLS.DSK
MAPDRV C: B:FILE.DSK --> Error
```

- 別のドライブへのマウントに使用しているファイルがある場合、ドライブの割り当て状態を変更することはできません。

```
MAPDRV B: A:TOOLS.DSK
MAPDRV A: U --> Error
```

- マウント元のファイルを開いたり、変更(名前の変更、移動、削除、上書き、属性の変更)したりすることはできません。

```
MAPDRV B: TOOLS.DSK
TYPE TOOLS.DSK --> Error
ECHO HELLO > TOOLS.DSK --> Error
REN TOOLS.DSK X.DSK --> Error
MOVE TOOLS.DSK SOMEDIR\ --> Error
DEL TOOLS.DSK --> Error
ATTRIB +R TOOLS.DSK --> Error
```

注: 現在はバグのため、`ECHO HELLO > TOOLS.DSK` は実際にはエラーをスローしません。

警告: ファイルをマウントした後、ファイルが格納されているメディアの排出や交換をしないでください。これを行うと Nextor の動作は未定義であるため、データが失われる可能性があります。

Warning: After mounting a file do not extract or swap the medium where the file is contained. The behavior of Nextor if this is done is undefined and you could lose data.

3.9. ディスクエミュレーションモード

Nextor バージョン 2.1 以降では、ディスクエミュレーションモードでの起動が可能になりました。このモードでは、Nextor は通常のデバイスではなく、ディスクイメージファイル（または交換可能なファイルセット）を起動デバイスとして使用します。これは、ファイルシステムがない場合や、MSX-DOS 1 モードで実行する必要がある、フロッピーディスクでリリースされ、最新のストレージデバイスから実行できないディスクを再生する場合に最適です。

[EMUFILE.COM](#) を使用する代わりに独自のツールを作成することに興味がある場合は、ディスクエミュレーションモードの仕組みに関する技術的な詳細は、[Nextor 2.1 プログラマーズリファレンス](#) ドキュメントに記載されています。

3.9.1. ディスクエミュレーションモードの開始と終了

まず最初に、ディスクエミュレーションモードセッション中に必要なデータ（使用されるディスクイメージファイルとその保存場所）は、特定の形式のファイル（ディスクエミュレーションデータファイル）に存在する必要があります。これらのファイルは、[EMUFILE.COM ツール](#) を使用して作成できます。これらのファイルの名前は任意で、通常は拡張子が .EMU ですが、必須ではありません。

次に、Nextor をディスクエミュレーションモードで起動するには、コンピュータの起動時に適切なディスクエミュレーションデータファイルへのポインタが特定の場所に存在する必要があります。エミュレーションモードには2つの種類があり、それぞれエミュレーションデータファイルポインタの位置が異なります。

- **ワンタイム:** Nextor は一度だけディスクエミュレーションモードに入ります。つまり、コンピュータをリセットした後、Nextor は通常通り起動します。このモードでは、エミュレーションデータファイルへのポインタは RAM 上に設定されます。
- **永続的:** Nextor は、コンピュータをリセットするたびにディスクエミュレーションモードに入ります。起動時に 0 キーを押して手動で無効にするまで、このモードは有効です。このモードでは、エミュレーションデータファイルへのポインタが、Nextor が制御するデバイスのいずれかのパーティションテーブルに設定されます（通常はエミュレーションデータファイルとディスクイメージファイルが格納されているデバイスと同じですが、必須ではありません）。

[EMUFILE.COM](#) ツールで `set` パラメータを使用することで、どちらのタイプのディスクエミュレーションモードにも入れることができます。デフォルトは、1 回限りのモードです。

3.9.2. イメージファイルの変更

エミュレーションセッションでは最大32個のディスクイメージファイルを指定できますが、一度にアクティブになるのはそのうちの1つだけです。別のファイルに切り替えるには、コンピュータがファイルを読み取ろうとする前に適切なキーを押す必要があります。これにより、ディスクの交換がエミュレートされます。キーは、最初の9個のイメージファイルでは1~9、残りのイメージファイルではアルファベット順にA~Wです。

例えば、2枚のディスクを使用するゲームをプレイしているとします。ディスク1で起動すると、ある時点でゲームがディスク2を挿入してスペースキーを押すように要求します。2（2番目のイメージファイルに割り当てられたキー）とスペースキーを同時に押すだけで、準備完了です。

あるいは、コンピュータがファイルを読み取ろうとしているときにGRAPHキーを押すこともできます。CAPS LEDが点灯し、GRAPHキーを離して適切なファイルキーを押すまでコンピュータがフリーズになります（ま

たは、気が変わって同じディスクを使い続けたい場合は、もう一度GRAPHキーを押すこともできます）。これは、ディスク アクセスの実行中に英数字キーを直接押す必要がある場合に問題が発生する場合に便利です (たとえば、BASIC プromptで FILES コマンドを実行するときにファイルの変更をトリガーする場合、押されたキーが "FILES" に追加され、構文エラーが発生します)。

3.9.3. ルールと制限

ディスクエミュレーションモードには、以下のルールと制限が適用されます。

- プライマリコントローラは、デバイスベースドライバを備えた Nextor カーネルである必要があります。
- エミュレーションデータファイルとすべてのディスクイメージファイルは、プライマリコントローラが制御するデバイスに配置する必要があります (ただし、異なるパーティションや異なるデバイスに配置することもできます)。
- エミュレーションデータファイルには物理デバイスセクターに関する情報が格納されるため、ディスクイメージファイルが移動すると使用できなくなり、ファイル名を変更しても効果がありません。ファイルを使用する直前に生成するか、ディスクイメージファイルとそれに対応するエミュレーションデータファイル専用のパーティション (つまり、通常はファイルの作成や移動を行わないパーティション) を用意することをお勧めします。
- ディスクイメージファイルのサイズは 512 バイト以上 32 MB 以下で、パーティションを含んではならず (含まれるファイルシステムはファイルの先頭から始まると想定されます)、適切な FAT12 ファイルシステムでなければなりません (ディスクエミュレーションモードでは FORMAT コマンドは機能しません)。
- ディスクイメージファイルは断片化されてはいけません。つまり、その内容はデバイス内の連続したセクタにまたがって配置されている必要があります。
- ディスクエミュレーションモードは、DOS 1 モードと Z80 モードで常に起動されます。ゲームを R800 モードで起動したい場合は、以下の手順に従ってください。コンピュータの起動中に GRAPH キーと 2 キーを押し続け、CAPS LED が点灯したら両方のキーを放し、1 キーを押します。
- ディスクエミュレーションモードに入ると、プライマリコントローラを除くすべての Nextor コントローラが無効になります。MSX-DOS カーネル (内蔵フロッピーディスクドライブなど) は無効になりませんが、起動中に SHIFT キーを押すことで強制的に無効にすることができます。これはメモリを解放するのに役立ちます。

3.9.4 メモリを解放する方法

一部のゲームは、システムにフロッピーディスクドライブのみが存在することを前提としているため、そのままでは動作しません。しかし、Nextor とフロッピーディスクドライブの両方にドライブが割り当てられているため、空きメモリが少なくなっています。ゲームで使えるメモリ量を増やすには、以下の手順を実行してください。

- 起動中に SHIFT キーを押すと、内蔵フロッピーディスクドライブ (およびその他の MSX-DOS カーネル) が無効になります。

- 起動中に 5 キーを押すと、Nextor が自身にドライブを 1 つだけ割り当てるように強制できます（Nextor コントローラに複数のデバイスが接続されている場合にのみ有効です）。エミュレーションセッションに 5 つ以上のディスクイメージがある場合は、代わりに以下の手順を実行してください。GRAPH キーと 5 キーを同時に押して CAPS LED が点灯したら、両方のキーを放し、1 キーを押します。

3.9.5. 既知のバグ

- 現在のバージョンの [EMUFILE.COM](https://emufile.com) ツールでは、ディスクイメージファイルが断片化されていないかどうかを確認できません。
- プライマリ Nextor コントローラに複数のデバイスがある場合（例えば、MegaFlashROM SCC+ SD の場合では、2枚のSDカード、または1枚か2枚のカードと ROM ディスク）、Nextor は追加デバイスごとにダミーのドライブ文字を割り当てます。MSX-DOS デバイス（存在する場合）には、これらのドライブ文字の後にドライブ文字が割り当てられます。例えば、デバイスが 3 つある場合、A: はエミュレートされたディスクイメージファイルがマウントされているドライブ、B: と C: はダミードライブ、D: は内蔵フロッピーディスクドライブです。これらのダミードライブには、FAT バッファ用のメモリは割り当てられません。

4. その他の改善点

4.1. F7のload"

Nextor は、デフォルトで F7 キーに `cload` が割り当てられている MSX1 および MSX2 コンピュータであっても、`load` 文字列を割り当ててコンピュータを強制的に起動します。ただし、INIFNK BIOS ルーチンと呼び出すコードを実行すると、キーが再び `cload` に割り当てられることに注意してください（自分で試すことができます: `_USR(&H3E)`）。

4.2. 漢字モードでの英語エラーメッセージ

ERRLANGという環境項目をENの値で作成しており（コマンドプロンプトでコマンド「SET ERRLANG=EN」を実行）、漢字モードが有効な場合（BASICインタープリタで「CALL KANJI」を実行していた場合）、コマンドインタープリタのエラーメッセージは日本語ではなく英語で表示されます。この機能はNextor 2.0.4以降で利用可能です。

4.3. 日本語エラーメッセージを削除したNEXTOR.SYS

NEXTOR.SYSファイルには2つのバリエーションが提供されています。完全版にはエラーメッセージの一部（「読み/書き」部分や「中止(A)/再試行(R)/無視(I)?」文字列など）で日本語文字列が含まれていますが、縮小版には英語文字列のみが含まれています。縮小版の利点はサイズが小さく、完全版と比較してTPA領域を256バイト節約できることです。

これら2つのバリエーションは、NEXTOR.SYSバージョン2.01（カーネルバージョン2.0.4と同時にリリース）以降で提供されています。バージョン2.00は既に縮小版として提供されていましたが、漢字モードで適切なエラーメッセージではなく、文字化けしたエラーメッセージが表示されるバグがありました。

ERRLANG環境項目がENの値で存在する場合、使用するバリエーションに関わらず、エラーメッセージは英語で表示されます。（[4.2. 漢字モードでの英語エラーメッセージ](#)を参照）。

5. 変更履歴

この章では、Nextor のさまざまなバージョンの変更履歴について説明します。アプリケーションやドライバー開発に影響する変更はここには記載されていません。そのタイプの変更の一覧については、[Nextor 2.1 プログラマーズリファレンス](#) および [Nextor 2.1 ドライバー開発ガイド](#) を参照してください。

この一覧には、2.1 ブランチの変更のみが記載されています。2.0 ブランチの変更履歴については、「Nextor 2.0 ユーザーマニュアル」を参照してください。

※訳注：v2.1.0 RC 以降の履歴は、Githubの変更履歴(<https://github.com/Konamiman/Nextor/tags>)を参考にしました。

5.1. v2.1.3

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.3>

- CALL MAPDRV の「デバイスセクタ 0 に割り当て」オプションを修正 ([#151](#))
- ZAP_ALL を修正し、デバイスベースドライバーに割り当てられたドライブをスキップするようにしました ([#147](#))
- デバイスへの書き込み後にセクタバッファが無効化されるとクラッシュする問題を修正 ([#148](#))
- _GETCLUS が 16 バイトではなく 64 バイトのデータをコピーする問題を修正 (NEXTOR.SYS 内) ([#149](#))
- [2025-06-08 更新] printf ルーチンで長整数型のサポートを有効にしました ([#155](#))

5.2. v2.1.2

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.2>

- Cプログラムの[printf.c](#)と[crt0_msxdos_advanced](#)に小さな修正を加えました([#135](#))。
- MegaFlashROM SCC+ SD ROMは、MSX2以降で起動時に保存された画面設定を復元するようになりました([#132](#))。
- [_DOSVER](#)関数は、MSX-DOS 1モードでレジスタCを変更しないようになりました([#128](#))。
- [SunriseIDE.emulators.ROM](#)ファイルの名前を[SunriseIDE.blueMSX.ROM](#)に変更しました([#129](#))。
- [MAPDRV](#)のバグと、MegaFlashROM SCC+ SDドライバーの[DEV_STATUS](#)ルーチンに関するバグを修正しました([#119](#))。
- カーネルおよび[MAPDRV.COM](#)でパーティションタイプコード15（「拡張LBA」）を「拡張パーティション」として認識するサポートを追加しました。また、FDISKでパーティションタイプコード15の拡張パーティションを作成できるようになりました。また、既存のパーティションを修正するためのコマンドラインツールも提供されています([#121](#))。
- MSX-DOS 1のページ1転送におけるDSKIOドライバーのエラーチェックを修正しました([#134](#))。

5.3. v2.1.1

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.1>

- 韓国語版MSXコンピュータでFDISKがクラッシュするバグを修正しました(#106)。
- EXTPIO処理中のスロット管理に関するバグを修正しました。このバグにより、特定のスロット/サブスロットの組み合わせで2つのNextorカーネルとマップRAMが存在する場合に、COMMAND2.COMがクラッシュしていました(#103)。
 - この修正は@uniabisによるものです。ありがとうございます！
- MSX-DOS 1モードで誤って実装された割り込み処理を修正しました。このバグにより、MSX-DOSカーネルの割り込みハンドラが場合によっては実行されず、例えばディスクドライブのモーターが停止しなくなるという問題がありました(#107)。
- Sunrise IDE ROMファミリーに新しいフレーバー「マスターデバイスのみ」(ファイル名の一部に.MasterOnlyが含まれるファイル)を追加しました。これは、Carnivore2や、1つのIDEデバイスのみを扱う同様のデバイスをお持ちの方に役立ちます(#105)。

5.4. v2.1.1 beta 2

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.1-beta-2>

- マウントされたファイルへのアクセス方法をファイルベースからセクタベースに変更しました。これにより、ディスクイメージファイルをドライブにマウントするには、連続するディスククラスタにまたがって保存する必要が生じます。(#93)
 - また、新しいBASICエラーコード「無効なクラスタシーケンス」(コード82)が追加されました。
- FATチェーンの解放時にUD_ACLUの設定が誤っていた問題を修正しました。この設定により、初期クラスタ番号が小さいファイルが作成され、それより大きいクラスタ番号のファイルにアクセスした後に削除されると、コンピュータがクラッシュする可能性がありました(#96)。
- BASICに新しいCALL SYSTEM2コマンドを追加しました。これはCALL SYSTEMと同じように動作しますが、NEXTOR.SYSが利用可能な場合でも常にMSXDOS2.SYSをロードします。(#97)
- 新しい環境変数KILLDSKIOを追加しました。この変数がonに設定されている場合、BASICで512バイトの空き領域が利用可能になりますが、DSKI\$関数とDSKO\$命令は無効になります。 (#98)

5.5. v2.1.1 beta 1

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.1-beta-1>

- 修正: 拡張BIOS処理コードによってDEレジスタペアが破損していましたが、ドライバー自身のEXTPIOフック処理コードによって破損した場合でも、常に保持されるようになりました。(#88)

- 修正: `_GPART`関数呼び出しが0005hエントリポイント経由で呼び出された場合、レジスタIXとIYに誤った情報を返していました(F37Dhエントリポイントを使用した場合は正常に動作していました)。修正はNEXTOR.SYSで行われました。(#89 ですが、この問題は元々vipooによって #84 で報告されていました)。
- 新しいカーネルROMファイル、`Nextor-2.1.1-beta1.OCM.ROM`が追加されました。これは1チップMSXおよびSX|2などの互換デバイス用です。ドライバーはMegaFlashROM SCC+ SD用のものと基本的に同じですが、DiskROM関連のコードが削除されています。(#91)

5.6. v2.1.1 alpha 2

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.1-alpha-2>

クラスタチェーン割り当てプロセスのパフォーマンス向上 (v2.1.1 alpha 1) により、ディスクへのデータ書き込み時に深刻な問題 (奇妙なエラー、コンピュータのクラッシュ、さらにはデータ破損) を引き起こすことがある重大なバグが発生しました。今回のリリースではこのバグを修正しました。

技術的な詳細については、[修正プルリクエスト](#)をご覧ください。

注意事項

v2.1.1 アルファ1と同様：このリリースでは、ファイルシステムへのデータの書き込み方法に大幅な変更が加えられています。このリリースを試す前に、データのバックアップを作成するか、テストデータで試してみることをお勧めします。

5.7. v2.1.1 alpha 1

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.1-alpha-1>

- [#68 クラスタチェーン割り当てのパフォーマンス向上](#)

割り当てられた最後のクラスタ番号をドライブごとにキャッシュする新しい機能により、空き容量の少ないFAT16パーティションへの書き込みが大幅に高速化されました。

- [#70 FDISKにおけるFAT16ファイルシステムのクラスタ数計算の修正](#)

以前のバージョンのNextorでFDISKを使用して作成されたFAT16ファイルシステムでは、FDISKでクラスタ数を計算するルーチンのバグにより、ブートセクターのクラスタ数がファイルシステムの実際のサイズよりも大きくなっていました。この問題は本バージョンで修正されました。

注意事項

このリリースでは、ファイルシステムへのデータの書き込み方法に大幅な変更が加えられています。このリリースを試す前に、データのバックアップを作成するか、テストデータで試してみることをお勧めします。

5.8. v2.1.0

訳注：以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.0>

注意事項

- .SunriseIDE.ROM バージョンは、エミュレータ上のマスターデバイスを認識しません。互換性を高めるに実機で使用してください。エミュレータでは .SunriseIDE.emulators.ROM バージョンを使用してください。
 - Carnivore2 コントローラーを使用している場合は、使用するカードに応じてどちらか一方を使用する必要があります。詳細については [#60](#) を参照してください。
 - ドライバー開発者の方は、変更ログに記載されている重大な変更にご注意ください。

更新 - 2020-08-05

- Flahsjacks 用ドライバーを含むカーネルが更新され、ドライバー v1.7.1 になりました。

変更点と新機能

注: 記載されている変更点は [v2.1.0 beta 2](#) に対するものです。

ユーザー向け

- 新しいツールCONCLUS.COMを追加しました。このツールは、新しい `_GETCLUS` 関数を使用して、ディスク内の連続したセクターに保存されているかどうかを判断します。
- EMUFILE.COMツールの新バージョンでは、新しい `_GETCLUS` 関数を使用して、ディスクイメージファイルがディスク内の連続したセクタに保存されているかどうかを確認します。
- EMUFILE.COMツールは、SET コマンドのパラメータとしてディレクトリを指定できるようになりました。`\DSKS\GAME` がディレクトリの場合、`EMUFILE SET \DSKS\GAME` は `EMUFILE SET \DSKS\GAME\GAME.EMU` と同等です。
- 新しいブートキーを追加しました。起動時に `N` キーを押すと、存在するすべての Nextor カーネルが無効になります。
- NEXBOOT.COMツールが更新され、起動時に `N` キーをエミュレートできるようになりました。
- NEXTOR.SYSがバージョン 2.11 にアップグレードされ、新しい `_GETCLUS` 関数呼び出しがサポートされました。

開発者向け

- FAT12 または FAT16 ボリューム内のクラスタに関する情報を取得するための新しい関数呼び出しを追加しました ([#35](#))。
- セグメント番号とスロット番号の両方を指定して、マッパー RAM にアクセスするための、UNAPI RAM ヘルパー互換のマッパーサポートルーチンを追加しました ([#34](#))。
- LUN_INFO によって返される新しいフラグにより、Nextor がデバイスとドライブの自動割り当てのためのデバイス検索時にデバイスを無視するように指示できます ([#54](#))。

- デフォルトの DPB はバンク 0 と 3 の固定アドレス 7BAAh を取得するため、確実にカスタマイズできます。
- **PROMPT** ルーチンがアドレス 41E8h でドライバーから利用可能になりました (#42)。
- C で記述されたすべてのツールが再びコンパイルされるようになりました (#36)。
- **重大な変更** (アプリケーション開発者向け): **ALL_BK** および **FRE_BK** ルーチンは利用できなくなりました。

バグ修正

- 修正: BASIC の古い MSX-DOS 1 ファイル関数でディスクエラーが発生し、コンピュータがクラッシュする問題を修正しました (#59)。
- 修正: DOS 1 モードでファイルの拡張子が「3」か「4」のファイルを利用できません (#55)。
- 修正: 3 を押して直接 BASIC を起動した後、CALL SYSTEM が動作しない。
- 修正: 起動時に DOS 1 モードに切り替えた際に、DOS 1 変数「データバッファ変更」および「コンソール出力をプリンタにリダイレクト」がゼロに設定されていなかった。
- 修正: ルートディレクトリに長いファイル名エントリがある場合、ボリューム名を変更できない (#57)。
- 修正: ファイルをドライブにマウントした際に、物理セクターの読み書き機能が正しく動作しない (#43)。
- 修正: マウントされたファイルへの書き込み時に不良セクターバッファ管理が行われ、マウントされたファイルのデータが破損する (#58)。
- 修正: ドライバーが DRV_CONFIG 経由でドライブマッピングを提供している場合、メディア変更後にドライブに初めてアクセスした際にドライブパラメータが更新されず、不正なデータ読み取りが発生する (#45)。
- 修正: ディスクエミュレーションモードでエミュレーション作業領域の不正なアドレスを使用すると、一部のゲームで起動エラーが発生する。

5.9. v2.1.0 RC

訳注 : 以下から引用

<https://github.com/Konamiman/Nextor/releases/tag/v2.1.0-RC>

- セグメント番号とスロット番号の両方を指定してマップされた RAM にアクセスするための、[#34 UNAPI RAM ヘルパー互換のマッパーサポートルーチン](#)を追加しました。
- FAT12 または FAT16 ボリューム内のクラスタ情報を取得するための新しい関数呼び出しを追加しました。
 - ディスク内の連続したセクタに保存されているかどうかを判定する新しいツール CONCLUS を追加しました。

- ディスクイメージファイルがディスク内の連続したセクタに保存されているかどうかをチェックする EMUFILE ツールの新バージョンを追加しました。
- EMUFILE ツールは、SET コマンドのパラメータとしてディレクトリを受け入れるようになりました。
\\DSKS\\GAME がディレクトリの場合、EMUFILE SET \\DSKS\\GAME は EMUFILE SET \\DSKS\\GAME\\GAME.EMU と同等です。
- [#36 C](#) で記述されたすべてのツールが再びコンパイルできるようになりました。
- アプリケーション開発者向けの重大な変更: ALL_BK および FRE_BK ルーチンは使用できなくなりました。
- NEXTOR.SYS は、新しい _GETCLUS 関数呼び出しをサポートするために、v2.11 にアップグレードされました。

5.10. v2.1.0 beta 2

- ブートドライブに「NEXTOR.SYS」が見つからない場合、Nextor は「MSXDOS2.SYS」のロードを試みるようになりました。
- 自動割り当ての対象となるパーティションの選択方法が、ルートディレクトリに「NEXTOR.DAT」ファイルが存在することを要求する方法から、パーティションテーブルに「アクティブ」フラグが設定されていることを条件とする方法に変更されました。
- 自動割り当て処理中に、デバイスの最初の 9 つのパーティション（拡張パーティションを含む）がスキャンされるようになりました。
- FDISK で、新規および既存のパーティションの「アクティブ」フラグを変更できます。
- FDISK は、定義されているパーティションが 4 つ以下の場合でも、常に拡張パーティションを作成するようになりました。
- FDISK は、パーティションタイプコードが 6 (FAT16 CHS) ではなく 14 (FAT16 LBA) の FAT16 パーティションを作成するようになりました。
- 起動時とディスクエミュレーションモードでのディスク交換時に、テンキーを使用できるようになりました。
- ロシア語キーボードが正しく認識されるようになりました（数字キーのみ）。
- [ブートキーインバータ](#) を導入しました。
- [ワнтаイムブートキー](#) を導入しました。
- RAMベースのワнтаイムブートキーを設定するための [NEXBOOT.COM ツール](#) を導入しました。
- RAMベースのワнтаイムディスクエミュレーションモードを導入しました。
- 従来のディスクエミュレーションモード（現在は「persistent」という名称）に入る方法が、ルートディレクトリに [NEXT_DSK.DAT](#) ファイルを置く方法から、デバイスのパーティションテーブルにエミュレーションデータファイルへのポインタを保存する方法に変更されました。

- 起動時に 0 キーを押すと、パーティションテーブル内のエミュレーションデータファイルへのポインタが削除され、永続的エミュレーションモードが無効になります。手動で他に操作する必要はありません。
- ディスクエミュレーションモードで GRAPH キーを押した後、Nextor がディスクキーの押下を待機しているときに、もう一度 GRAPH キーを押すとディスク交換がキャンセルされるようになりました。
- 最初に起動する Nextor カーネルは、ドライバーの初期化を呼び出す前に画面をクリアするようになりました。
- ARG が Nextor カーネルによって一時的な作業領域として使用されなくなりました。これにより、ディスクエミュレーションモードでのゲームの互換性が向上します。
- 修正: エラーが発生した場合、ドライブが再割り当てされていました (再試行に成功した場合でも)。
- 修正: ブートセクタのチェックサム計算にバグがあり、「Wrong Disk (間違ったディスク)」エラーが発生していました。
- 修正: #1 Nextor が NEXTOR.SYS をロードしようとしているときに Ctrl+Stop キーを押すと、コンピューターがハングする問題。
- 修正: #23 単一のドライブレーターで起動するとコンピュータがハングする問題 (例: 内蔵ディスクドライブのないコンピュータでシングルデバイスコントローラを使用している場合)。
- 修正: #29 スタック管理の誤りにより、ファイルハンドルの読み取りまたは書き込みが複数回行われた場合にコンピュータがハングする問題。
- 修正: 拡張BIOSフックが呼び出された直後に複数のNextorカーネルが存在する場合に (例: COMMAND2.COMのロード時)、コンピュータがクラッシュする問題。

5.11. v2.1.0 beta 1

- [Nextor 2.0.5] のすべての変更と修正。
- ファイルマウントおよびディスクエミュレーションモードを導入しました。
- ブートキーの変更: ドライバーごとに1つのドライブを要求するキーが 5 キーに変更され、MSX-DOS でブートする場合と同様に、CTRL キーが MSX-DOS カーネルに渡されるようになりました (2.9. ブートキー を参照)。
- #3 _RDBLK で読み取り専用ファイルにアクセスできない を修正しました。