32ビット仮想コンソール

システム仕様

パート9:ファイルフォーマット

資料作成日2023.01.08年Carra著

これは何だ?

このドキュメントは、Vircon32システム仕様の部品番号9です。この一連のドキュメントは、Vircon32システムを定義し、その機能と動作を詳細に説明する完全な仕様を提供します。

この仕様の主な目標は、Vircon32システムとは何か、および準拠していると見なされるためにゲームシステムがどのように実装される必要があるかについての標準を定義することです。また、Vircon32は仮想システムであるため、これらの文書の重要な第2の目標は、独自のVircon32実装を作成するための知識を誰にでも提供することです。

Vircon32について

Vircon32プロジェクトはCarraによって独自に作成されました。Vircon32システムとその関連資料(ドキュメント、ソフトウェア、ソースコード、アート、およびその他の関連要素を含む)は、元の作成者が所有しています。

Vircon32は無料のオープンソースプロジェクトで、誰でもゲーム機をプレイし、そのためのソフトウェアを開発できるようにすることを目的としている。この詳細については、使用可能な各ソフトウェアに含まれているライセンステキストを参照してください。

このドキュメントについて

このドキュメントは、Creative Commons Attribution 4.0 License(CC BY 4.0)に基づいて提供されています。ライセンスの全文は、Creative CommonsのWebサイトで読むことができます。

<https://creativecommons.org/licenses/by/4.0/>

まとめ

仕様のパート9では、Vircon32の実装が動作中にロードおよび/または保存する必要があるすべての情報に使用されるファイル形式について説明している。これには、ファイルヘッダーとレイアウト、データ形式、およびファイルの正確性に関する基本的な検証が含まれる。

[1 Introduction 3](#_30j0zll)

[2 Types of Vircon32 files 3](#_1fob9te)

[3 Used data formats 5](#_3znysh7)

[4 Program binary files 7](#_2et92p0)

[5 GPU texture files 7](#_tyjcwt)

[6 SPU sound files 9](#_3dy6vkm)

[7 Vircon32 ROM files 10](#_1t3h5sf)

[8 Vircon32 memory card files 13](#_4d34og8)

## 1はじめに

Vircon32システムが動作するためには、外部情報を処理できる必要があります。具体的には、BIOSとカートリッジの内容を読み取る必要があり、メモリカードの内容を読み書きする必要があります。これらの外部コンテンツは「ファイル」として表現されると考えます。

すべての実装は、異なるファイルシステムでこれらのファイルを処理し、異なる物理的サポートを使用してそれらを保存する可能性があります。このため、私たちはこれらの詳細を抽象化します:私たちの目的のために、「ファイル」は単に順序付けられたバイトのシーケンスで構成されます。オプションで、ファイルには名前を関連付けることができます。また、ファイルは永続的であると考えます:その内容は、特定のVircon32システムがオフになっても保存され、アクセス可能なままになります。

Vircon32システム(BIOS、カートリッジ、メモリカード)で使用される外部コンテンツを格納するファイルは、異なるVircon32コンソール間で交換可能である必要があります。異なる実装間での互換性を確保するために、このドキュメントでは、これらのファイルタイプのそれぞれが情報を表すために使用するフォーマットについて説明します。

このドキュメントで説明されている表現形式は、情報交換(つまり、Vircon32システムの境界を越える情報)のためにのみ必須であることに注意してください。ただし、実装では、内部表現または処理に他の形式を使用することが依然として許可されています。たとえば、ハードウェア実装では、ビデオハードウェアからのピクセルへのより簡単なアクセスを可能にするために、GPUテクスチャに別の内部表現を使用する必要がある場合があります。

## 2種類のVircon32ファイル

まず、Vircon32システムで使用されるさまざまなファイルタイプとその基本的な特性を列挙します。これらの各ファイルタイプについては、次のセクションで詳細に説明します。

#### ROMファイルROMふぁいる

ROMは、すべてのVircon32システムで使用される主要なファイルです。ROMは、プログラムと必要なテクスチャおよびサウンドを1つのファイルにまとめた実行可能ファイルです。Vircon32では、ROMファイルにファイル拡張子\*.v32を使用します。

BIOSとカートリッジはどちらも実行可能ファイルであるため、両方とも同じファイル形式で保存されます。唯一の違いは、BIOSの場合、ファイルの署名が変更され、いくつかの追加の制限が適用されることです。これについては、セクション7で詳しく説明します。

#### アセットファイル

前述したように、ROMには、プログラムバイナリ、GPUテクスチャ、SPUサウンドの3種類のアセットが含まれています。この各アセットは、スタンドアロンファイルとして個別に表現することもでき、その結果、次の3つのファイル形式が得られます。

* **プログラムバイナリファイル**(拡張子\*.vbin)

これらには、1つのプログラムの完全なバイナリ(プログラム+データ)が含まれています。

* **GPUテクスチャファイル**(拡張子\*.vtex)

これらには、グラフィックスチップ(GPU)で使用できる単一のテクスチャが含まれています。

* **SPUサウンドファイル**(内線\*.vsnd)

サウンドチップ(SPU)で使用可能な単一のサウンドが含まれています。

これらのアセットは、開発ツールまたはその他のサポートプログラムによってスタンドアロンファイルとして処理されることのみを目的としています。Vircon32システム自体は、通常、この方法でアセットを使用しません。ただし、これらのフォーマットはROMファイル自体の一部として存在するため、ここで説明する必要があります。

これらのすべての形式について考慮すべき重要な点は、アセットが常に圧縮されずに保存されることです。これにより、ファイルのサイズが大きくなりますが、これらのファイルの読み取り/書き込みがはるかに簡単になり、特定のイメージまたはサウンド形式を処理するための追加のオーディオおよびビデオライブラリからの依存を防ぐことができます。

#### メモリカードファイル

メモリカードは基本的に、Vircon32プログラムが読み書きできるメモリ領域です。実行終了後、そのメモリの内容はファイルに格納されます。Vircon32はメモリカードファイルにファイル拡張子\*.memcを使用します。

### 2.1ファイルの署名

ファイル拡張子を使用してファイルの種類を区別することは便利ですが、信頼性は十分ではありません。ユーザーがファイル拡張子を変更する可能性があり、ファイルシステムによっては拡張子がまったくサポートされていない場合があります。このため、ファイルの種類をファイルの内容自体から識別できる必要があります。

この目的のために、すべてのVircon32ファイルは、そのコンテンツ形式を識別するために、最初の8バイトに既知の「ファイル署名」を含んでいます。次のセクションでは、ファイルヘッダーの一部として、さまざまな可能なファイル署名を示します。ファイル署名は、すべてのファイルヘッダーの最初のフィールドとして常に赤で表示されます。

ファイルシグニチャは、標準ASCIIでエンコードされた8文字のシーケンスとして保存されます。ただし、特定のシグニチャを認識するには完全一致が必要であるため、64ビット値の形式に依存しない比較を実行するには十分です。たとえば、有効なBIOSファイルでは、ファイルの最初の8バイトが次のように一致することが予想されます。

ファイル署名を使用すると、サイズが8バイト以下のファイルは、無効なVircon32ファイルとしてすぐに破棄されます。

## 使用する3つのデータ形式

これらのファイルにはすべて、フィールドまたはシーケンスに配置された情報が含まれています。すべてのフィールドまたはシーケンス要素は、次のデータ形式のいずれかを使用してエンコードされます。すべての4バイトデータ形式では、バイトはリトルエンディアンで配置されることに注意してください。

### 3.1文字

文字は、ASCIIでエンコードされた1バイトです。これらのファイル・フォーマットのデータ・フィールドに文字が含まれている場合は、それらを固定容量の文字配列にグループ化して文字列を形成します。文字は、次の2種類の文字列で使用されます:

* **ファイルの署名**これらは常に正確に8文字であるため、NULL終端ではありません。有効なファイル署名に使用される文字は、標準ASCII(0～127の数値)に制限されています。
* **ROMタイトル**これらは可変長であるため、NULLで終了します。値が0のバイトは、最後に使用された文字の後に配置され、文字列が終了する場所を示します。ROMタイトルの場合、拡張ASCIIの使用が許可され、Latin-1としても知られるコードページ1252に従って解釈されます。

### 3.2整数

これらのファイルは、常に4バイトを使用して整数を表します。これらの各整数は、32ビットの符号なし整数として解釈されます。これは、Cのデータ型「uint32\_t」と同じです。

### 3.3 CPUワード

プログラムバイナリデータを含むファイルは、格納された各ワードを単一の32ビットバイナリ値として表します。プログラムの実行中、Vircon32 CPUは各ワードを命令または他のデータ形式として解釈できますが、これらのファイルのコンテキスト内では、格納されたワードは32ビットのシーケンス以外の解釈を必要としません。

### 3.4 GPUピクセル

GPUテクスチャを含むファイルは、格納された各ピクセルを単一の32ビットRGBAカラーとして表します。これについては、仕様のパート2ですでに説明されています。

### 3.5 SPUサンプル

SPUサウンドを含むファイルは、仕様のパート2ですでに説明されているように、LおよびRチャンネルの16ビット値をエンコードする単一の32ビット値としてそれぞれのサンプルを表します。

## 4プログラムバイナリファイル

ファイルの署名とは別に、プログラムバイナリファイルのヘッダには、このバイナリを構成する32ビットCPUワードの数を示す1つの整数フィールドのみが含まれる。

ヘッダーの後、ファイルにはこれらのCPUワードのすべてが順番に含まれます。

### 4.1プログラムバイナリファイルの検証

正しいファイル・シグニチャーを持つこととは別に、プログラム・バイナリー・ファイルが有効であることを確認するために、以下のアサーションをテストすることができます。

1. 宣言されたワード数は有効です。範囲は、1から適用可能なプログラムROMサイズ制限までです(セクション7のカートリッジおよびBIOSの制限を参照)。
2. バイト単位のファイルサイズは、12+4 x{データワード数}でなければなりません。

## 5 GPUテクスチャファイル

GPUテクスチャの内部サイズは1024 x 1024ピクセルに固定されていますが、ほとんどのゲームテクスチャはその領域の一部のみを使用します。そのため、テクスチャファイルには、指定された幅と高さの長方形のみが格納されます。したがって、ファイルシグネチャとは別に、GPUテクスチャファイルのヘッダーには、格納されたテクスチャ領域のピクセル単位の寸法を示す2つの整数フィールドのみが含まれます。

ヘッダーの後、ファイルにはこれらのピクセルのすべてが順番に含まれます。

### 5.1テクスチャピクセルの格納順序

格納されたテクスチャ領域の矩形は、フルテクスチャの左上の部分とみなされます。フルサイズまでのテクスチャピクセルの残りは、ゼロに等しい4つのRGBAコンポーネントすべてを持つとみなされます(つまり、完全に透明)。

ピクセルは左上隅から格納され、最初に左から右に、次に上から下に順に格納されます。たとえば、サイズが4 x 3ピクセルの格納されたテクスチャの場合、これら12ピクセルの格納順序は次のようになります。

### 5.2 GPUテクスチャファイルの検証

GPUテクスチャファイルが有効であることを確認するために、正しいファイルシグネチャのほかに、次のアサーションをテストできます。

1. 宣言されたテクスチャサイズは有効です。幅と高さの両方で、範囲は1から1024までです。幅と高さは同じである必要はありません。
2. ファイルサイズ(バイト単位)は、16+4 x{幅(ピクセル単位)}x{高さ(ピクセル単位)}でなければなりません。

## 6 SPUサウンドファイル

ファイル・シグニチャーとは別に、SPUサウンド・ファイルのヘッダーには、このサウンドを構成する32ビット・サンプルの数を示す1つの整数フィールドしか含まれていません。

ヘッダーの後、ファイルにはこれらのサンプルのすべてが順番に含まれています。注意事項として、Vircon32は、すべてのSPUサンプルを44100サンプル/秒の再生レートを有するものとして解釈する。

### 6.1 SPUサウンドファイルの検証

正しいファイル・シグニチャーを持つこととは別に、SPUサウンド・ファイルが有効であることを確認するために、以下のアサーションをテストすることができます。

1. 宣言されたサンプル数は有効です。範囲は、1から適用可能なオーディオROMサイズ制限までです(セクション7のカートリッジとBIOSの制限を参照)。
2. ファイルサイズ(バイト単位)は、12+4 x{サンプル数}でなければなりません。

## 7 Vircon32 ROMファイル

ROMファイルの構造は、他の形式よりも複雑です。これはヘッダーにも反映されます。サイズは128バイトで、次のフィールドが含まれます:

カートリッジとBIOSの両方が同じファイル形式を共有します。これらは、異なるファイル署名によって区別されます:カートリッジの使用**V 32-カート**、BIOSは**V32-BIOS対応**です。

青で色付けされたフィールド(バージョンとリビジョン)は、<version . revision>形式のバージョン番号として一緒に解釈される整数のペアです。Vircon32標準の現在のバージョンは1.0です(最初のバージョンであり、したがってサポートされているのは1つだけです)。

ROMのタイトルとバージョンは、ROMメタデータとしてのみ提供されます。これらの情報を処理するために、またはそれらを読み取るためにさえ、実装は必要ありません。

オレンジ色のフィールド(オフセットとサイズ)も整数のペアです。これらは、ROMファイルにまとめられた3つのROMに対応するファイル内の3つの領域を区切るために使用されます。これらは、実行可能ファイルの3つのコンポーネントです:プログラムROM、ビデオROM、およびオーディオROM。

ROMファイルのレイアウトは、次の3つのコンポーネントをヘッダーに追加するだけで表示できます。

オフセットとサイズはすべてバイト単位で表されるため、常に4の倍数である必要があります。オフセットはファイルの先頭からのバイト数で表されます。たとえば、プログラムROMオフセットはヘッダーの直後に位置するため、常に128である必要があります。

これらの3つのコンポーネントのうち、必須のものはプログラムROMだけであることに注意してください。宣言されたテクスチャおよびサウンドの数がそれぞれ0の場合、ビデオおよび/またはオーディオROMは存在しません。

### 7.1 3つのコンポーネントROMのレイアウト

ROMファイルの3つの部分は、前のセクションで説明したアセットの3つのファイル形式を再利用します。ここでは、各アセットは、個々のアセットファイルをROMファイルに埋め込むだけで保存されます。そうすることで、プログラムROMのレイアウトは、プログラムバイナリファイルのレイアウトと同じになります(常に1つのバイナリがあるため)。

ビデオROMの場合、そのレイアウトは、次のように順番に格納される埋め込みGPUテクスチャファイルのシーケンスになります。

同様に、Audio ROMレイアウトは、埋め込まれたSPUサウンドファイルのシーケンスで構成されます。

### 7.2カートリッジROMファイルの検証

正しいファイル・シグニチャーを持つこととは別に、カートリッジROMファイルが有効であることを確認するために、以下のアサーションをテストすることができます。

1. 記載されているVirconのバージョンとリビジョンは、バージョン1.0に対応している必要があります。
2. ファイルサイズ(バイト単位)は、128+{プログラムROMサイズ}+{ビデオROMサイズ}+  
   {オーディオROMのサイズ}
3. 宣言されたテクスチャの数は、0から256までの範囲でなければなりません。
4. 申告されたサンプル数は、0から1024までの範囲内でなければならない。
5. 既存のテクスチャはすべて、1～1024ピクセルの有効な幅と高さを持っている必要があります。
6. 既存のプログラムバイナリには、1からカートリッジプログラムROMの制限(128 x 1024 x 1024)ワードまでの有効なワード数が必要です。
7. 既存のサウンドのそれぞれには、1からカートリッジオーディオROMの制限である(256 x 1024 x 1024)サンプルまでの有効なサンプル数が必要です。
8. 既存のサウンドのすべてのサンプルの合計は、カートリッジオーディオROMの制限(256 x 1024 x 1024)サンプルを超えてはなりません。

### 7.3 BIOS ROMファイルの検証

正しいファイル署名を持つこととは別に、BIOS ROMファイルが有効であることを確認するために、次のアサーションをテストできます。

1. 記載されているVirconのバージョンとリビジョンは、バージョン1.0に対応している必要があります。
2. ファイルサイズ(バイト単位)は、128+{プログラムROMサイズ}+{ビデオROMサイズ}+  
   {オーディオROMのサイズ}
3. 宣言されたテクスチャとサウンドの数は両方とも1でなければなりません。
4. 既存のテクスチャには、有効な幅と高さ(1～1024ピクセル)が必要です。
5. 既存のバイナリのワード数は、1からBIOSプログラムROMの制限(1024 x 1024)ワードまでの範囲である必要があります。
6. 既存のサウンドのサンプル数は、1からBIOSオーディオROMの制限(1024 x 1024)サンプルまでの範囲である必要があります。

## 8 Vircon32メモリカードファイル

Vircon32システムは、メモリカードの内容を使用可能なメモリ位置の範囲として取り、そのサイズは常に正確に256 KWords=1 MBです。そのため、メモリカードファイルのヘッダーには、ファイルの署名自体以上の情報は必要ありません。したがって、これらのファイルのヘッダーは、単純に次のようになります。

ヘッダの後のファイルには、カードのすべての32ビットCPUワードが順番に含まれています。

注意事項として、カードの内容の最初の20語は「ゲームの署名」と見なされます。これらの語はカードの残りの部分と同じように扱われますが、ゲームでは通常、これらの語を使用して各カードの内容を識別し、他のゲーム用のデータ(互換性のない保存形式)がロードされないようにします。

### 8.1メモリーカードファイルの検証

正しいファイル署名を持つこととは別に、メモリカードファイルが有効であることを確認するために、次のアサーションをテストできます。

1. ファイルサイズ(バイト単位)は1048584でなければなりません。

(パート9の終わり)