~~ZIPの解凍あたりのプログラムを作ってみた系~~

GZIPファイルやZIPファイルを自分で作ろう

ぽぴ王子@わんくま同盟

アジェンダ

- 自己紹介
- きっかけ
- tar形式のフォーマット解説
- ZIP形式のフォーマット解説
- 実際に圧縮ファイルを作ってみるデモ
- まとめ

ぽぴ王子とはこんな人

- 王冠の人
- 仕事は王子兼システムエンジニア
- オンラインでは威勢はいいが、オフラインでは意外とシャイです
- パソコン歴は25年ほど
- SEとしてのPC歴は20年ぐらい
- Microsoft MVPを再受賞しました

セッションのきっかけ

- .NET Framework 2.0から
  - GZipStream
  - DeflateStream
- という二種類の圧縮/伸張に関するクラスが追加されている

- ん？DeflateってZIPで使われてる圧縮形式じゃなかった？

セッションのきっかけ

- Wikipediaにはこう書いてある

> Deflate（デフレート）はPhil Katzが開発した圧縮ツールPKZIPのバージョン2で使われているデータ圧縮アルゴリズム。ZIPやgzipなどで使われている。

- ということは、これでZIP書庫を作ったり解凍したりできるんじゃね？
- ……と思っていた時期が僕にもありました。

セッションのきっかけ

- MSDNを見ると

> このクラスは本来、.zip アーカイブとの間でファイルの追加や抽出を行うための機能を提供するものではありません。

とか書いてあるの。

- 実際はDeflateStreamはZIPの**圧縮部分**のストリームなので、圧縮してもZIP書庫にはなりません！

セッションのきっかけ

- でもDeflateStreamがZIPの圧縮部分と同じモノなのだとすれば、それで足りないヘッダとか付ければZIP書庫になるんじゃないかしら？
- というのがきっかけです。

…長いよ。

Deflateとは

- PKZIPのバージョン2以降で使われているデータ圧縮アルゴリズム
- PKWAREのPhil Katz(フィル カッツ)氏が開発
- 圧縮は比較的高速、伸長は非常に高速
- ZIPやgzipなどで使われている
- パテントフリー

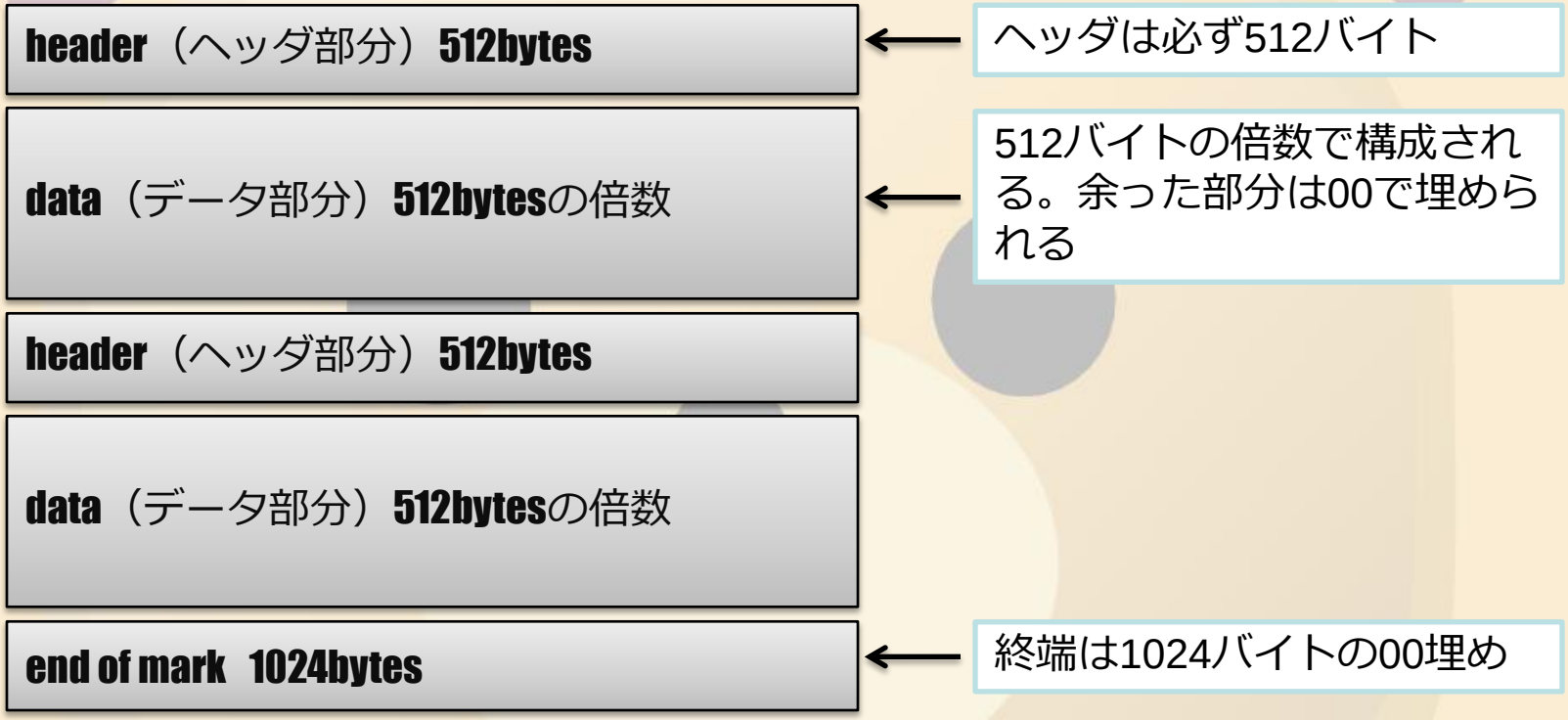
保証されているわけではないが、特許にかかわるアルゴリズムは一切使用されていないと考えられている

tar形式について

- tarはTape ARchive formatの略
  （Tape Archive and Retrieval formatとも）
- その名の通りテープに保存するために複数のファイルを連結したもの
- tarで連結したあと、GNU zipを使用して圧縮を行う
- tarで連結したものは拡張子 .tar になり、それを gzip圧縮したものは .tar.gz となる

tar形式について

- tar形式のファイル構造



tar形式について

- tar形式のヘッダは以下のようになっています

フィールド名	バイト数
ファイル名	100
属性	8
ユーザーID	8
グループID	8
ファイルサイズ	12
更新日時	12
チェックサム	8
タイプ	1
リンク先ファイル名	100
マジックコード/バージョン番号	8
ユーザ名	32
グループ名	32
メジャーデバイス番号	8
マイナーデバイス番号	8
予約領域	167

パラメータの解説

基本はASCIIの文字列（！！）数値は8進数の文字列として格納される

- ファイル名
  - ASCIIまたはSJISで100バイト（パス名含む）
- ユーザID/グループID
  - “0”はルートをあらわす
- 更新日時
  - ustat()で得られる最終更新日時の値を8進数文字列であらわしたもの（詳細は割愛）

パラメータの解説

- チェックサム
  - ヘッダ512バイト分のチェックサム
  - チェックサム自身はスペース8文字として計算する
- マジックコード/バージョン番号
  - “ustar¥0” + バージョン番号 “00”
- ユーザ名/グループ名
  - null終端のASCII文字列
- メジャーデバイス番号/マイナーデバイス番号
  - タイプが[3]または[4]の場合のみ使用

パラメータの解説

- 属性（16ビット分が8進数の文字列として登録される）

ビット	説明
0	他人の実行属性
1	他人の書き込み属性
2	他人の読み込み属性
3,4,5	グループの属性
6,7,8	オーナーの属性
9	sticky bit（詳細不明）
10	set GID
11	set UID
12	パイプ
13	キャラクタ型スペシャルファイル
14	ディレクトリ
15	通常のファイル

パラメータの解説

- タイプ

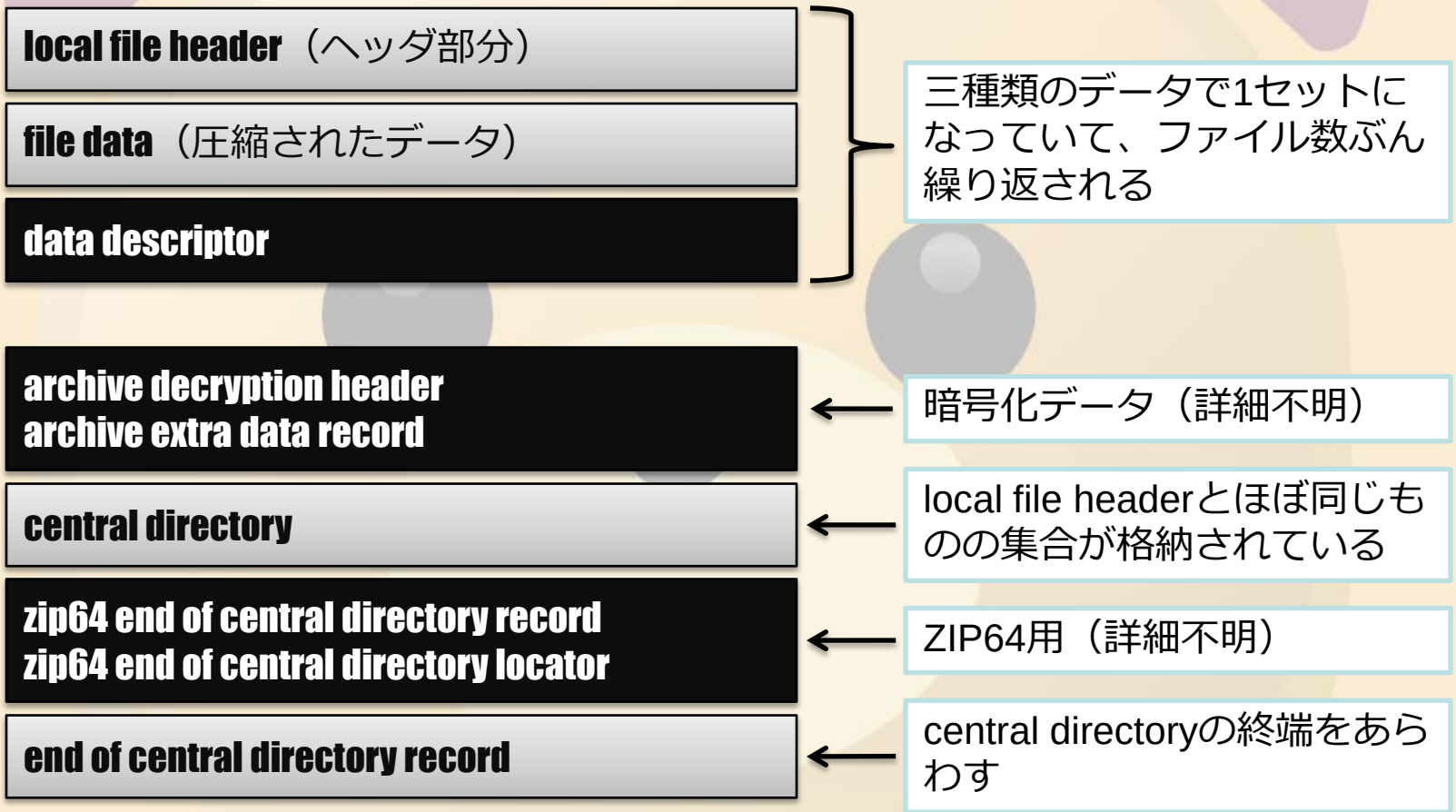
文字	説明
0	通常のファイル
1	リンク（詳細不明）
2	シンボリックリンク（詳細不明）
3	キャラクタ型デバイス（詳細不明）
4	ブロック型デバイス（詳細不明）
5	ディレクトリ
6	FIFOスペシャルファイル（詳細不明）
7	リザーブ？
A-Z	将来のために予約

ZIPファイルについて

- 日本ではあまりなじみがないが、欧米では割とメジャーな圧縮形式
- Implobe/Deflate/Deflate64/Bzip2 などの圧縮形式を使用可能（一般的に Deflate が使用される）
- 暗号化にも対応
- Javaは標準ライブラリで使用可能

ZIPファイルの構造

- 全体の作りはこんな感じ



local file header

- ヘッダ情報はこんな感じになってます

説明（日本語）	説明（英語）	バイト数
シグネチャ	local file header signature (0x04034b50)	4
解凍に必要なバージョン	version needed to extract	2
設定ビット	general purpose bit flag	2
圧縮形式	compression method	2
最終変更時刻	last mod file time	2
最終変更日付	last mod file date	2
CRC32	crc-32	4
圧縮後のサイズ	compressed size	4
圧縮前のサイズ	uncompressed size	4
ファイル名サイズ	file name length	2
拡張領域のサイズ	extra field length	2
ファイル名（可変）	file name (variable size)	
拡張領域（可変）	extra field (variable size)	

central directory

- local file headerとほぼ同様です

説明（日本語）	説明（英語）	バイト数
シグネチャ	central file header signature (0x02014b50)	4
作成されたバージョン	version made by	2
解凍に必要なバージョン～拡張領域のサイズはlocal file headerと同じ		
コメントのサイズ	file comment length	2
開始ディスク番号	disk number start	2
内部ファイル属性	internal file attributes	2
外部ファイル属性	external file attributes	4
ローカルヘッダの位置	relative offset of local header	4
ファイル名（可変）	file name (variable size)	
拡張領域（可変）	extra field (variable size)	
ファイルコメント（可変）	file comment (variable size)	

end of central directory record

- central directoryの終端をあらわします

説明（日本語）	説明（英語）	バイト数
シグネチャ	end of central dir signature (0x06054b50)	4
ディスク番号	number of this disk	2
詳細不明	number of the disk with the start of the central directory	2
ディスク内のセントラルディレクトリのエントリ数	total number of entries in the central directory on this disk	2
セントラルディレクトリ内のエントリ数	total number of entries in the central directory	2
セントラルディレクトリのサイズ	size of the central directory	4
詳細不明	offset of start of central directory with respect to the starting disk number	4
ZIPファイルのコメントサイズ	.ZIP file comment length	2
ZIPファイルのコメント（可変）	.ZIP file comment (variable size)	

パラメータの解説

- 作成されたバージョン
  - 上位バイトは以下の環境をあらわす
  - 下位バイトは作成されたZIPフォーマットのバージョンをあらわす（2.0ならば10進数で20［0x14］）

値	説明	値	説明
0	MS-DOS and OS/2	11	MVS (OS/390 - Z/OS)
1	Amiga	12	VSE
2	OpenVMS	13	Acorn Risc
3	UNIX	14	VFAT
4	VM/CMS	15	alternate MVS
5	Atari ST	16	BeOS
6	OS/2 H.P.F.S.	17	Tandem
7	Macintosh	18	OS/400
8	Z-System	19	OS/X (Darwin)
9	CP/M	20-255	未使用
10	Windows NTFS		

パラメータの解説

- 解凍に必要なバージョン

値	説明
1.0	デフォルト値
1.1	ボリュームラベル
2.0	フォルダ（ディレクトリ） Deflateアルゴリズム PKWARE製の伝統的な暗号化方式
2.1	Deflate64アルゴリズム
2.7	パッチデータ（詳細不明）
4.5	ZIP64フォーマット
4.6	Bzip2アルゴリズム
5.0	DES/3DES/オリジナルRC2暗号/RC4暗号
5.1	AES/正式な(corrected)RC2暗号
5.2	正式な(corrected)RC2-64暗号
6.2	central directoryが暗号化されている？
6.3	LZMA/PPMd+/Blowfish/Twofish

基本的には2.0を指定しておけば大丈夫です。

パラメータの解説

- 設定ビット

<table>
<tr><th>ビット</th><th>説明</th></tr>
<tr><td>Bit 0</td><td>暗号化されていることを示す</td></tr>
<tr><td>Bit 1-2</td><td>以下の組み合わせによる
<table>
<tr><th>Bit 0</th><th>Bit 1</th><th>意味</th></tr>
<tr><td>0</td><td>0</td><td>通常圧縮</td></tr>
<tr><td>0</td><td>1</td><td>最大圧縮</td></tr>
<tr><td>1</td><td>0</td><td>速度優先</td></tr>
<tr><td>1</td><td>1</td><td>最大速度優先</td></tr>
</table></td></tr>
<tr><td>Bit 3</td><td>1だった場合はCRC32フィールドとサイズフィールドが0になり、正しい値はdata descriptorで設定される</td></tr>
<tr><td>Bit 4</td><td>予約</td></tr>
<tr><td>Bit 5</td><td>パッチデータ（詳細不明）であることをあらわす</td></tr>
</table>

パラメータの解説

- 設定ビットの続き

ビット	説明
Bit 6	AES暗号化されていることを示す このビットをセットする場合にはBit 0もセットして下さい
Bit 7-10	未使用
Bit 11	Language encoding flag (EFS) Early Feature Specification このビットがセットされている場合はファイル名がUTF-8でエンコーディングされている
Bit 12	PKWAREによって予約
Bit 13	central directoryが暗号化されていることを示す(?) 詳細はStrong Encryption Specificationを参照のこと （よくわかっていません）
Bit 14-15	PKWAREによって予約

パラメータの解説

- compression method
  代表的なもののみ示す

値	説明
0	圧縮なし
8	Deflate圧縮
9	拡張Deflate圧縮 Deflate64
12	BZip2圧縮

パラメータの解説

- 日付

Bit	説明
0-4	日付
5-8	月
9-15	1980年からの経過年

- 時刻

Bit	説明
0-4	秒の1/2の値
5-10	分
11-15	時

パラメータの解説

- そのほかのパラメータについて
  - CRC32
    - 未圧縮のデータのCRC32を示します
  - 圧縮後のサイズ
  - 圧縮前のサイズ
  - ファイル名
    - 文字コードとして基本的にASCII または日本語の場合はシフトJISを使用
    - ただしMacintoshで作成された書庫などでUTF8が使われる場合もあり

実際にファイルを作ってみる

- tar.gzファイルを作る
  - tar形式を作ってからGZipStreamで圧縮する
- ZIPファイルを作る
  - ヘッダ部分を設定したものと、DeflateStreamを使用した圧縮データを合成する

⇒自作プログラムを使用したデモ

まとめ

- GZipStreamまたはDeflateStreamを使用してtar.gzファイルやZIPファイルを作成することは可能！
- ただし圧縮率を指定することはできない
- どちらかと言えば圧縮よりも解凍に特化した方がいいかもしれない

参考資料

- tarの構造
  - http://www.redout.net/data/tar.html
- TAR32.DLL フォーマット説明ファイル
  - http://openlab.ring.gr.jp/tsuneo/soft/tar32_2/tar32_2/sdk/TAR_FMT.TXT
- APPNOTE.TXT - .ZIP File Format Specification
  - http://www.pkware.com/documents/casestudies/APPNOTE.TXT