Lempel–Ziv–Stac

From Wikipedia, the free encyclopedia

Lempel–Ziv–Stac (**LZS**, or **Stac compression**) is a lossless data compression algorithm that uses a combination of the LZ77 sliding-window compression algorithm and fixed Huffman coding. It was originally developed by Stac Electronics for tape compression,[1] and subsequently adapted for hard disk compression and sold as the Stacker disk compression software. It was later specified as a compression algorithm for various network protocols. LZS is specified in the Cisco IOS stack.

Contents [hide]

Standards [edit]

LZS compression is standardised as an INCITS (previously ANSI) standard.[2]

LZS compression is specified for various Internet protocols:

- RFC 1967 – *PPP LZS-DCP Compression Protocol (LZS-DCP)*
- RFC 1974 – *PPP Stac LZS Compression Protocol*
- RFC 2395 – *IP Payload Compression Using LZS*
- RFC 3943 – *Transport Layer Security (TLS) Protocol Compression Using Lempel-Ziv-Stac (LZS)*

Algorithm [edit]

LZS compression and decompression uses an LZ77 type algorithm. It uses the last 2 KB of uncompressed data as a sliding-window dictionary.

An LZS compressor looks for matches between the data to be compressed and the last 2 KB of data. If it finds a match, it encodes an offset/length reference to the dictionary. If no match is found, the next data byte is encoded as a "literal" byte. The compressed data stream ends with an end-marker.

Compressed Data Format [edit]

Data is encoded into a stream of variable-bit-width tokens.

Literal byte [edit]

A literal byte is encoded as a '0' bit followed by the 8 bits of the byte.

Offset/length reference [edit]

An offset/length reference is encoded as a '1' bit followed by the encoded offset, followed by the encoded length. One exceptional encoding is an end marker, described below.

An offset can have a minimum value of 1, maximum value of 2047. A value of 1 refers to the most recent byte in the history buffer, immediately preceding the next data byte to be processed. An offset is encoded as:

- If the offset is less than 128: a '1' bit followed by a 7-bit offset value.
- If the offset is greater than or equal to 128: a '0' bit followed by an 11-bit offset value.

A length is encoded as:

Length	Bit Encoding

2	00
3	01
4	10
5	1100
6	1101
7	1110
8 to 22	1111 xxxx, where xxxx is length - 8
23 to 37	1111 1111 xxxx, where xxxx is length - 23
length > 7	(1111 repeated N times) xxxx, where N is integer result of (length + 7) / 15, and xxxx is length - (N*15 - 7)

End marker [edit]

An end marker is encoded as the 9-bit token 110000000. Following the end marker, 0 to 7 extra 0 bits are appended as needed, to pad the stream to the next byte boundary.

Patents [edit]

Stac Electronics' spin-off Hifn has held several patents for LZS compression.[3][4] These patents had lapsed due to non-payment of fees and attempts to reinstate them in 2007 had failed.

In 1993–94, Stac Electronics successfully sued Microsoft for infringement of LZS patents in the DoubleSpace disk compression program included with MS-DOS 6.0.[5]

See also [edit]

- LZ77
- MPPC

References [edit]

1. ^ Stac Electronics
2. ^ INCITS/ANSI X3.241-1994 - *Data Compression Method – Adaptive Coding with Sliding Window for Information Interchange*
3. ^ Friend, Robert C. "Hifn's Statement about IPR claimed in draft-friend-tls-lzs-compression, RFC1967, RFC1974, RFC2118, RFC2395, and RFC3078". Retrieved 21 July 2010.
4. ^ Friend, Robert. "Hifn's Statement on IPR Claimed in LZS and MPPC compression algorithms". Retrieved 21 July 2010.
5. ^ Complaint for patent infringement and Demand for jury trial by Stac Electronics v Microsoft Corporation

Compression formats · Compression software (codecs)

Categories: Lossless compression algorithms