Noel Ball
CSE13S

Design Document Lempel-Ziv Compression

Description:

Encode compresses any file, text, or binary input, with command line arguments that allow for printing statistics, and selecting the input and output files. Decode decompresses any file, text, or binary input, with command line arguments that also allow for printing statistics, and selecting the input and output files. Both executables depend on function libraries for tries, word tables, and I/O modules.

Goal:

Running encode compresses a valid input, and decode decompresses the encoded data, 4kb at a time, using a trie. Encode and decode must be interoperable with the provided binaries, and operate on little and big endian systems, with varying bit length inputs.

Pseudo:

Encode.c is the main C file for compressing inputs to an output

- Initializes default values and sets up get opt arguments for command line inputs
- Opens in and out files
- Creates stat struct to store file size and protection
- Writes header to outfile, and sets its permissions
- Creates a trie, a copy of its root node, a code counter, and prev node and sym

- Sets up a uint8_t buffer and reads symbols from infile to buffer
    - Next node is a step down from current node to current symbol
    - If prefix is seen move forward
    - If prefix is not seen write it and add it to trie, increment code
    - If next code is max code reset trie and mod next code by max code
- Write the last prefix, and then the stop code pair
- Flush the buffer and close files

Decode.c is the main C file for decompressing inputs to an output

- Initializes default values and sets up get opt arguments for command line inputs
- Opens in and out files
- Makes fileheader struct to store header from infile
- Sets permissions to outfile
- Creates a new word table, with code counters and symbol buffer
- While reading pairs from infile:
    - Stop when curr_code == stop_code
    - Append read symbol with word denoted by the read code, and write it to file
    - Increment the code
    - If next_code reaches max_code
        - Reset word table
        - Reset next code to start code
- Flush the buffer and close files

Noel Ball
CSE13S

Trie.c is a library of the following functions:

- node_create()

Allocates space for a node, setting it's code based on the parameter

Set children pointers to null

- node_delete()

Frees space for a node

- trie_create()

Allocates space for the root trie node, with EMPTY_CODE

Returns node pointer

- trie_reset()

Recursively delete children nodes

- trie_delete()

Delete tree from the root

- trie_step()

Returns pointer to child representing the symbol

Noel Ball
CSE13S

Word.c is a library of the following functions:

- word_create()

Allocates space for word based on length parameter

- word_append_sym()

Returns pointer to array with the appended symbol in word

- word_delete()

Frees memory for word with a single pointer

- wt_create()

Allocates space for MAX_CODE array

Calls word_create() for the EMPTY_CODE word

- wt_reset()

Sets all words to NULL besides the first

- wt_delete()

Frees memory for every word in word table

Noel Ball
CSE13S

Io.c is a library of the following functions:

- read_bytes()

Loops read until to_read parameter or until all bytes have been read

Returns the number of bytes read

- write_bytes()

Loops write until to_write parameter or no bytes were written

Returns the number of bytes written

- read_header()

Swaps endianness if byte order isn't little endian

Reads size of file header bytes from input to the header

- write_header()

Swaps endianness if byte order isn't little endian

Writes size of file header bytes from input to the output file

- read_sym()

Index tracks the currently read symbol in buffer, block at a time

Returns true if there are symbols to be read, false if not

- write_pair()

Write to outfile a pair of the code and symbol

- flush_pairs()

Writes to buffer the remaining pairs

- read_pair()

Reads a pair to set a pointer for the code and symbol

- write_word()

Adds symbol from word to buffer

- flush_words()

Writes to outfile the remaining buffer