

Wendi Tan
wtan37@ucsc.edu

Assignment 7

Compression:

- open infile (default file is stdin)
- define magic number, file size, and protection in the header file
- if the permission matches the protection bits, open the outfile successfully (default out file is stdout)
- use write_header() to determine size of byte to the outfile, and to swap endianness if needed
- create a trie. Root should be EMPTY_CODE, 1, without any child. Make a copy of it, make it as current node
- make a counter to keep track of the next assignable code. create one variable for tracking previous trie node, and create another one for keeping track of previous symbol, naming them prev_node and prev_sym respectfully.
- make a loop, read symbols in infile with read_sym() function, mark each read symbol as curr_sym.....
- (...still understanding the steps...)
- close infile and outfile

Decompression:

- open infile (default file is stdin)
- define magic number, file size, and protection in the header file
- if the permission matches the protection bits, open the outfile successfully (default out file is stdout)
- create and initialize a word table
- create two uint16_t variables: one for current code, one for the next code, which will be named as curr_code and next_code.
- use read_pair() in a loop to read all pairs from infile, and the loop breaks when we read STOP_CODE.....
- close files.

NOTE:

STOP_CODE = 0

EMPTY_CODE = 1

START_CODE = 2 (the first word we start appending)

every single time we index a new letter in the word, you add a child in the try.

trie_delete() is a recursive function, trie_reset() is non-recursive.

we need to use trie_delete inside trie_reset. trie_reset is just to delete all children under the ROOT, and trie_delete is the one who delete the children of called children and recursively, all the children are deleted.