Chris Fretz
Professor Parashar
Systems Programming
11/5/14

Search

- System Overview
  - Indexer uses the following data structures
    - Index Node
      - Keeps track of the count for a specific token/filename pair.
    - Hash
      - Basic hashtable that hashes linked lists of index nodes against tokens.
    - Hash Node
      - Node used for storing data inside of hash. Allows for resolution of hash-collisions through chaining.
    - Splitter
      - Tokenizer that allows a given file to be tokenized through the use of the split_next helper function.
  - Search uses the following data structures
    - Index Node
      - Keeps track of the count for a specific token/filename pair.
    - Hash
      - Basic hashtable that hashes linked lists of index nodes against tokens.
    - Hash Node
      - Node used for storing data inside of hash. Allows for resolution of hash-collisions through chaining.
- Algorithm Overview
  - Search allocates a hash to store inverted index, and reads through the specified input file, updating the hash as it goes.
  - Search enters query loop and continuously prompts the user for search queries. After receiving one, search attempts to get all terms entered from the hash, ANDs/ORs the results depending on the type of query, sorts the results by frequency and name, then outputs the results.
- Memory Usage
  - Memory usage of each structure
    - index_node - 20 bytes.
      - Filename string, takes up 8 bytes locally, and usage in the heap is determined by the size of the filename.
      - Count integer, 4 bytes
      - Next pointer, 8 bytes.
    - hash_node - 24 bytes
      - Key string, takes up 8 bytes locally, and usage in the head is determined by the size of the key.
      - Data index node, 8 bytes locally, sizeof(index_node) bytes in the heap.
      - Next pointer, 8 bytes.

    - hash - 16 bytes
      - hash_node array, takes up 8 bytes locally, and usage in the heap is determined by the current number of elements contained in the hash.
      - Count integer, 4 bytes.
      - Size integer, 4 bytes.

Chris Fretz
Professor Parashar
Systems Programming
11/5/14

- Experimental Data
  - Program appears to use approximately o(2n) or O(n) (where n is the size of the input file) memory at runtime.
  - Data was generated by running variations on the big test documented in the testplan.txt file
    - Generated inverted index files for different subsets of the data contained in the big_test directory.
    - Sizes of different inverted index files were 3mb, 4mb, 5mb, 6mb, and the runtime memory usage was respectively 6.2mb, 8.4mb, 10.7mb, and 13mb (this was measured on my local computer, not the iLabs).
- Exception Handling
  - Search validates arguments passed in, state of specified file, and all memory allocations. In the case that any check fails, search prints an error message detailing the nature of the error to stderr and immediately halts execution.