

Assignment #4 - Word Blast

Goal: Read 'War's Peace' to count and tally each word longer than 6 characters or longer.

Only using Linux file functions (open, close, read, lseek, pread) so additional libraries

But: must do this using threads (This assignment will be using pthreads)

- Each thread will take a chunk of the file and process it, returning its results to the main which tallies.
- The main will print the top, 6 or more character, words with the highest tallies, from highest to lowest.
- Program should take two parameters - COMPLETE
 - FileName is the name of the file to read - WarAndPeace.txt ✓ Read-Only mode
 - TreadCount is the number of threads you want spawn to evenly divide the work ✓
- These threads should be launched together in a loop

- * HINTS:
- Do not forget to protect critical sections
 - Make sure to use thread safe library calls
 - strtok - r is a tokenizer that is thread safe
 - You need to know how long the input file is (use lseek)
 - Do not use pipes. You must use mutex locks

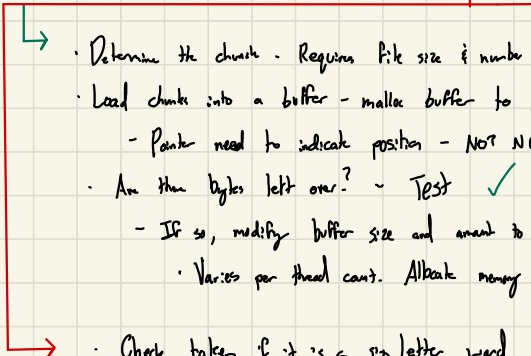
1000:

- Accepting parameters - COMPLETE
 - FileName - Read into opening files w/ r ✓
 - TreadCount as int (accepting argv[2] as TreadCount ✓
 - Argv is a pointer to a string - used atoi function to convert to int
 - You need to know how long the input file is (use lseek) Determine file size ✓
 - Partition must be determined (Divided by number of threads) ✓
- Creating a structure to the top 10 5's-character word tally.
 - Option: Create a hashmap to count each word occurrence
 - < key: word, Value: Occurrence Increment >
 - Define the structure - Needs string/char* for word, int for count
- Threading
 - Thread Pool - Create a number of threads at start-up and use as needed? ✓
 - Method or code block must be written that each thread will execute (Instructions or algorithm)
 - pthread - create, pthread_join, pthread_exit?
 - What needs to be used to destroy the threads after use? Mutex-lock? buffers ✓
 - Function must be created to run pthread_create (Code block to execute)
- Close file - Release allocated memory

• Threading (Cont.)

- Function must be created to run pthread_create (Code block to execute)
 - Then should be a buffer to hold chunks allocated to thread (malloc use) ✓
 - Tokeniser used (strtok_r)
 - Pointer used to point to token

~~Each thread will take a chunk of the file and process it, returning its results to the main which gathers.~~

- 
- Determine the chunks - Requires file size & number of threads ✓
 - Load chunks into a buffer - malloc buffer to load for each thread - using read() ✓
 - Pointer need to indicate position - NOT NEEDED ✓ CHECK AFTER COMPLETION
 - Are there bytes left over? ~ Test ✓
 - If so, modify buffer size and amount to pass it. How to allocate for this? ✓
 - Varies per thread count. Allocate memory in buffer to handle leftover for final thread? ✓
 - Check token if it is a six letter word
 - If so, check if it is in data structure - (compare string function - case sensitive)
 - If so, increment
 - If not, add and increment
 - Continue until token returns null