

EECS 498 W26 System Design of a Search Engine

07 Multithreaded file operations

Due Sunday, February 8, 2026, 11:59 pm

Introduction

This homework is designed to teach you how to create a multithreaded program using the pthread library using mutexes (pthread_mutex_t) and condition (pthread_cond_t) variables. Threads will become important when crawling the web because downloading a page is slow, with lots of time spent waiting for data. Threads let us overlap many downloads at once.

You will implement two utilities.

1. WordCount just counts the total number of words in all the files specified on the command line by creating a separate thread to word count each file. After all the threads finish, it prints the total. Files that can't be opened are ignored.
2. Traverse is more difficult. It uses a pool of worker threads to traverse any files and directories specified on the command line, printing a sorted list of all the paths it found.

Starter files

You are given these starter materials in MultithreadedFileOperationsStarter.zip.

WordCount.cpp	A starter file for a very simple multithreaded word count utility.
Traverse.cpp	A starter file for a utility to traverse files and directories similar to way find does, but using a thread pool.
Makefile	A starter make file that builds and tests WordCount and Traverse.
Documents	A directory with some subdirectories, sample documents, and expected outputs for testing.

WordCount

Here is the expected behavior. User input is in **bold**. zork does not exist but does not cause an error. The usage message is written to stderr and exits with status = 1. Normal exit is status = 0 with a result that matches the output of the standard wc utility.

```
% ./WordCount
Usage: WordCount <filenames>
Count the total number of word in the files.
Invalid paths are ignored.
% echo $status
1
% ./WordCount zork
Total words = 0
% ./WordCount Documents/Text/Shakespeare/*
Total words = 50562
% wc -w Documents/Text/Shakespeare/*
31955 Documents/Text/Shakespeare/Hamlet.txt
18607 Documents/Text/Shakespeare/Macbeth.txt
50562 total
```

Traverse

Here is the expected behavior. User input is in **bold**. Traverse gives the same results as the find utility, but sorted.

```
% ./Traverse
Usage: Traverse <number of workers> <pathnames>
Number of workers must be greater than 0.
Invalid paths are ignored.
% ./Traverse 5 Documents/Text/Lincoln Documents/Html
Documents/Html
Documents/Html/NYTimes.html
Documents/Text/Lincoln
Documents/Text/Lincoln/GettysburgAddress.txt
% find Documents/Text/Lincoln Documents/Html | sort
Documents/Html
Documents/Html/NYTimes.html
Documents/Text/Lincoln
Documents/Text/Lincoln/GettysburgAddress.txt
```

Autograder

Submit your WordCount.cpp and Traverse.cpp files at <https://autograder.io/web/project/3490>.