

This file discusses Project 2 which involves pthread multithread programming of the Pipeline design pattern.

Please read all of these instructions as failure to comply may result in a loss of points in your grade.

Your project deliverables are to be posted to CUNY Blackboard using the Project 2 link.

1) Only one source code file as follows with filename as in the example where you will put your own first initial and lastname and your

own section number. I show the example for my own name with an example section 23:
JSvadlenka_prj2_sect23_src.c (or .cpp)

a) At top of the file: your name in comments, for example: `// First_Name Last_Name`

b) Next, the gcc or g++ code compilation and linkage command, for example in my case (not yours) it would be:

`// gcc -o JSvadlenka_prj2_sect23_src.c -o JSvadlenka_prj2_sect23_src.exe`

Note that you will additionally need the `-pthread` option for compiling and linking.

c) In-line code comments no fewer than every 5 lines of code which describe your algorithmic steps. Syntax transcriptions of the code are

not considered in-line code comments! (For example, I do not want to see the comment for `i++` which says "Increment i.")

2) An MS Word or PDF document with a write-up of your software design and an explanation of the issues encountered and resolved. (1 page)

Filename should be according to this example of my own name:
JSvadlenka_prj2_sect23_wrt.doc (or pdf)

No ZIP file submissions.

3) The source code accepts one command line argument which is the name of the input text file to be read (input files to be provided later by instructor). For your own testing, please create and use your own test input text files.

Project 2 Description

1) Your program will make use of the POSIX system calls and pthreads library functions discussed in lectures. That is, you must not use any C or C++ standard library functions. Your program will have 3 pthreads in a pipeline pattern. The mainline thread opens the input text file to get a file descriptor, it also creates 3 pthreads, and simply waits for each pthread to do its work and terminate. So the mainline thread calls `pthread_join()` three times before closing the input file and exiting the program. Each pthread will do a different task so each of the 3 pthreads executes a different function. So let's name the three pthreads as pthread 1, pthread 2, and pthread 3.

Remember, pthread 1 is a producer of data for pthread 2, and pthread 2, in turn, is a producer for pthread 3 (the consumer for pthread 2). So pthread 2 is both a consumer and producer! You will be implementing the producer and consumer algorithms of Chapter 3. But wait, you say, thread 1 is easy as it is only a producer and thread 3 is easy because it is only a consumer. Yes, but remember you need different shared buffer arrays, and in and out variables between Threads 1 and 2 and between threads 2 and 3. `BUFFER_SIZE` size will equal 10 for the shared buffer array 1 between pthread 1 and pthread 2 while `BUFFER_SIZE` will be equal to 20 for the shared buffer array 2 between pthread 2 and pthread 3. Also, you will remember that I mentioned in lecture that you have to use the `sched_yield()` function for any thread producer or consumer as in:

```
#include <sched.h>
```

```

while((in + 1) % BUFFER_SIZE) == out) {

    // Insert sched_yield() function call here.
}

and for any thread consumer as in:

while (in == out) {
    // Insert sched_yield() function call here
}

```

So what will pthread 1 do? For each line of text it reads from the input text file, it will produce that line of text into an element of the shared buffer array 1. So what about about pthread 2 as it is both producer and consumer? That is where the interesting work lies! In a sense the consumption activity of thread 2 is really the production of the data content to eventually be consumed by thread 3. So just what exactly is that content being produced by pthread 2 and that will be consumed by pthread 3? It is all the words in the line of text consumed from an element of shared buffer array 1. So in pthread 2 you will have to parse each line into a set of all words in that line (what white space characters will you check?). Each word will be produced to an element of the shared buffer array 2. Additionally, pthread 2 will maintain a counter of the number of lines consumed from shared buffer 1.

What does pthread 3 do with the content it consumes from shared buffer array 2? It will write each word out to standard output (STDOUT_FILENO) with each word on its own line. After the last word of the input text file has been displayed, the total number of words will be printed out. Therefore, you have to maintain a word counter in pthread 3. Finally, pthread 3 will also print the total number of lines in the input file. Therefore, you have to devise a strategy to get the line counter from pthread 2 to pthread 3.

You are NOT permitted to use any C++ nor C standard I/O library functions nor cin nor cout. Furthermore, you cannot use any C++ nor C standard library functions for string operations.

OUTPUT: Remember, all thread output goes to the screen (standard out) and is done only by pthread 3.

Given a sample input file with only one line of text (of course, you can expect to have actual input files with more than one line) as follows:

Let us all sing out loud together and enjoy the good times while we can.

The output to the screen from pthread 3 will be the following:

```

*Let*
*us*
*all*
*sing*
*out*
*loud*
*together*
*and*
*enjoy*
*the*
*good*
*times*
*while*
*we*
*can*
*WORD COUNT = 15*
*LINE COUNT = 1*

```