CST 334: Operating Systems

Dr. Glenn Bruns

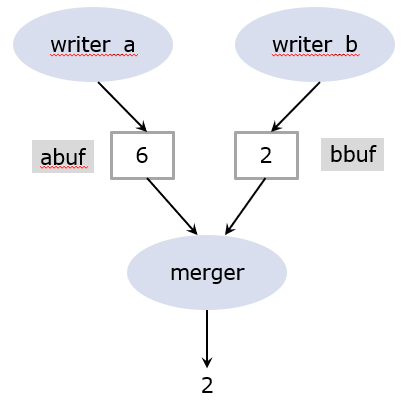
# Pthreads: simple merging

**Purpose**. The key skill we will need in pthreads programming is understanding how to use locks and especially how to use condition variables. The purpose of this programming assignment is to give you practice in using condition variables.

**Instructions**. In this assignment you will edit a file merge.c. Copy the tar file shown below to a directory of your own, and use 'tar' to extract the file.

/home/CLASSES/brunsglenn/cst334/hw/hw9/simple-merge.tar

One of the files in the tar file is merge.c. Read the code in this file carefully. The main program creates three threads. One thread runs the function write\_a(), another runs write\_b(), and the third runs merger(). The three processes share two integer variables: abuf and bbuf. Function write\_a() takes data from an array of integers and writes one array value after another into abuf. Similarly, write\_b() writes data from an int array into bbuf. The merger() function repeatedly takes a value from abuf or bbuf and prints it. However, merger() will print whichever of abuf or bbuf is smaller (if the values are equal it doesn't matter which is printed. This picture shows the idea:



In the example of the picture, but abuf and bbuf contain a value. In this situation the merger thread would remove the 2 from bbuf, and print it. This will leave bbuf empty, and abuf will not change. The writer\_b thread will put a new value into bbuf and the process will continue.

There are a couple of important details:

* Valid data values are non-negative. When writer\_a runs out of data, it will write the special value -2 to abuf (and similarly for writer\_b and bbuf).
* If writer\_a writes a value to abuf, it won't write to abuf again until merger has removed the value from abuf. When merger removes a value from abuf, it will put -1 in abuf to indicate that abuf is empty.
* Three condition variables are used to let the threads notify each other. For example, when merger has removed a value from abuf, it will signal on the ready\_awrite condition variable. When writer\_a has written a value into abuf, it will signal on the ready\_read condition variable.

Further details can be found in merge.c. Your job is to replace the comment YOUR CODE HERE in function merger() with your own code. Do not modify merge.c in any other way!

.

Hints:

* A comment in merger() explains the cases that merger must handle. For example, if the value of abuf is 6, and the value of bbuf is 2, then merger should print 2 and set bbuf to -1 (indicating that it is now empty). However, if abuf is -1 and bbuf is 2, then merger must wait for a value to be placed in abuf before doing anything else (because -1 means abuf is empty).

Testing your code: The tar file also includes test scripts and a Makefile. I may modify some of the scripts for different script and slightly different input file for grading. Make sure your code runs correctly on mlc104.

**Submission**: Submit your edited merge.c on iLearn.

**Grading**. You will get 5 points for each of 6 tests that your code passes. Please submit tidy code.

if(abuf == -2 && bbuf == -2){

return;

}

else if(abuf == -1 || bbuf == -1){

wait();

}

else if(abuf >= 0 && bbuf >= 0){

if(abuf < bbuf){

printf(abuf);

}

else{

printf(bbuf);

}

}

else if(abuf >= 0 && bbuf == -2){

printf(abuf);

}

else{

printf(bbuf);

}