

GEBZE TECHNICAL UNIVERSITY

CSE 344

SYSTEM PROGRAMMING

Homework #5 Report

Just Updates on HW4

BARIŞ YURDAKUL

1801042620

## Explanations

### Barriers:

- A barrier is added to ensure that all worker threads wait until they all complete their tasks. This is initialized at the start and destroyed at the end of the main function.
- The `pthread_barrier_wait` is called by each worker thread to synchronize their execution phases.

### Condition Variables:

- `pthread_cond_t buffer_not_empty` and `pthread_cond_t buffer_not_full` are used to manage the buffer state.
- Workers wait on `buffer_not_empty` if the buffer is empty, and the manager waits on `buffer_not_full` if the buffer is full.
- These condition variables are signaled when the state changes (an item is added or removed from the buffer).

*1. 'pthread\_barrier\_t' type was defined and 'worker\_barrier' was created and initialized in the main function.*

```
pthread_barrier_t worker_barrier;
```

*2. 'pthread\_barrier\_wait' functions were added to ensure correct use of the barrier in 'worker\_function' and 'main' functions.*

```

pthread_barrier_wait( barrier: &worker_barrier);
//printf("Main thread waits for all workers\n");// Main thread waits for all workers
pthread_join( th: manager_thread, thread_return: NULL);

for (int i = 0; i < num_workers; i++) {
    pthread_join( th: worker_threads[i], thread_return: NULL);
}

gettimeofday( tv: &end, tz: NULL);

print_statistics(num_workers, buffer_size, start, end);

pthread_barrier_destroy( barrier: &worker_barrier);
free( ptr: buffer);
return 0;

```

```

void *worker_function(void *args) {
    while (!done_flag_set() || buffer_count > 0) {
        char src_path[MAX_PATH];
        char dest_path[MAX_PATH];
        if (get_from_buffer(src_path, dest_path)) {
            if (src_path[0] != '\0' && dest_path[0] != '\0') {
                copy_file(src_path, dest_path);
            } else {
                fprintf( stream: stderr, format: "Invalid paths received by worker\n");
            }
        }
    }
}

pthread_barrier_wait( barrier: &worker_barrier);
//printf("Worker thread reached the barrier\n");// Each worker waits at the barrier
return NULL;
}

```

3.The '*pthread\_cond\_broadcast(&buffer\_not\_full)*' and '*pthread\_cond\_broadcast(&buffer\_not\_empty)*' signals added to the '*set\_done\_flag*' function ensured that all threads were informed of the situation when the program needed to stop.

```

void set_done_flag() {
    pthread_mutex_lock( mutex: &buffer_mutex);
    done_flag = 1;
    pthread_cond_broadcast( cond: &buffer_not_empty);
    pthread_cond_broadcast( cond: &buffer_not_full);
    pthread_mutex_unlock( mutex: &buffer_mutex);
}

```

## Results and Test Cases

- Test 1
  - > valgrind ./1801042620\_main 10 10 ../testdir/src/libvterm  
../tocopy

```
-----STATISTICS-----
Consumers: 10 - Buffer Size: 10
Number of Regular File: 194
Number of FIFO File: 0
Number of Directory: 7
TOTAL BYTES COPIED: 25009680
TOTAL TIME: 00:00.538 (min:sec.mili)
==26405==
==26405== HEAP SUMMARY:
==26405==      in use at exit: 0 bytes in 0 blocks
==26405==    total heap usage: 21 allocs, 21 frees, 348,464 bytes allocated
==26405==
==26405== All heap blocks were freed -- no leaks are possible
==26405==
==26405== For lists of detected and suppressed errors, rerun with: -s
==26405== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

- Test 2
  - > ./1801042620\_main 10 4 ../testdir/src/libvterm/src  
../tocopy

```
-----STATISTICS-----
Consumers: 4 - Buffer Size: 10
Number of Regular File: 140
Number of FIFO File: 0
Number of Directory: 2
TOTAL BYTES COPIED: 24300387
TOTAL TIME: 00:00.089 (min:sec.mili)
```

- Test 3
  - > ./1801042620\_main 10 100 ../testdir ../tocopy

```
-----STATISTICS-----
Consumers: 100 - Buffer Size: 10
Number of Regular File: 3116
Number of FIFO File: 0
Number of Directory: 151
TOTAL BYTES COPIED: 71566680
TOTAL TIME: 00:00.282 (min:sec.mili)
```

- In some examples, ‘Main thread waits for all workers’ print could be different line.

[illegible]