Appendix

Please find below a list of the synchronization primitives available:

Mutexes

- `mutex`: variable of type `pthread_mutex_t`
- `pthread_mutex_init(&mutex, ...)`: initialize the mutex
  - The macro `PTHREAD_MUTEX_INITIALIZER` can be used to initialize a mutex allocated statically with the default options
- `pthread_mutex_lock(&mutex)`
- `pthread_mutex_unlock(&mutex)`

Condition Variables

- `cond`: variable of type `pthread_cond_t`
- `pthread_cond_init(&cond, ...)`: initialize the condition
  - The macro `PTHREAD_COND_INITIALIZER` can be used to initialize a condition variable allocated statically with the default options
- `pthread_cond_wait(&cond, &mutex)`
- `pthread_cond_signal(&cond)`
- `pthread_cond_broadcast(&cond)`

Semaphores

- `sem`: variable of type `sem_t`.
- `sem_init(&sem, int pshared, unsigned int value)`: initialize the semaphore to `value`. Set `pshared` to 0 for a semaphore to be shared between threads.
- `sem_wait(&sem)`
- `sem_post(&sem)`

Atomic operations

- `test_and_set(type *ptr)`
- `fetch_and_add(type *ptr, type val)`
- `compare_and_swap(type *ptr, type oldval, type newval)`: returns `1` on success, `0` otherwise.