```
#ifndef __FIFO_H
#include "sem.h"

#define MYFIFO_BUFSIZ 4096

struct fifo
{
    unsigned long buffer[MYFIFO_BUFSIZ];
    int next_read, next_write;
    struct sem empty; // empty spots
    struct sem full;  // filled spots
    struct sem mutex; // mutex for operations
};

//    Initialize the shared memory FIFO *f including any
//    required underlying initializations (such as calling sem_init)
//    The FIFO will have a fifo length of MYFIFO_BUFSIZ elements,
//    which should be a static #define in fifo.h (a value of 4K is
//    reasonable).
void fifo_init(struct fifo *f);

//    Enqueue the data word d into the FIFO, blocking
//    unless and until the FIFO has room to accept it.
//    Use the semaphore primitives to accomplish blocking and waking.
//    Writing to the FIFO shall cause any and all processes that
//    had been blocked because it was empty to wake up.
void fifo_wr(struct fifo *f, unsigned long d);

//    Dequeue the next data word from the FIFO and return it.
//    Block unless and until there are available words
//    queued in the FIFO.  Reading from the FIFO shall cause
//    any and all processes that had been blocked because it was
//    full to wake up.
unsigned long fifo_rd(struct fifo *f);

#define __FIFO_H
#endif
```