Threads homework (Homework #2)
Due Wednesday, February 11, 2010 before the class

This homework focus on two of the simplest data structures used in connection with programming: lifos and dequeues. For more information about such data structures, consult the Donald Knuth, *The Art of Computer Programming*, Volume 1:Fundamental Algorithms (ISBN 0-201-89683-4), Section 2.2.1: Stacks, Queues, and Deques, pages 238-243.

A non thread-safe version of the code for each one of these data structures is provided on the class webpage. The goal of this homework is to transform these data structures into their thread-safe counterparts, and to evaluate and understand their performance and drawbacks.

**Last In First Out** LIFO is the acronym for Last In, First Out, also known as FILO (i.e. “first in, last out”). It describe the principle of a queuing system, where the most recent entry is the one to be processed first, and the oldest is the last to be processes. It is similar to a stack where the only allowed operations are to add or remove an object from the top of the stack.

(a) ![Diagram](image1)

(b) ![Diagram](image2)

(c) ![Diagram](image3)

Figure 1: A sketch for a LIFO with push and pop operations.

In the figure 1 (a) we have a design of an unbounded LIFO. On the (b) part the push operations is showed. The new element (in gray) is pushed to the bottom of the FIFO (and it will be returned only after all previously posted elements have been returned). On the (c) part the pop operation is described. The *oldest* element in the FIFO is returned while the gray element added in the (b) part is still the *newest* in the FIFO. The links between the elements should be updated by each operation in order to keep the LIFO consistent.
Deque  A double-ended queue (or deque, or head-tail linked list) is an abstract data structure that implements a queue for which elements can only be added to or removed from the front (head) or back (tail).

Figure 2: A sketch for a deque with push and pop from front and back operations.

In the figure 2 (a) we have a design of an unbounded deque, with the arrows representing the links between the records of the deque. On the (b) part the push back operation is showed, where the new elements is pushed at the end of the deque. On the (c) part the pop front operation is showed, where the element at the head (front) of the deque is remove. Both operations will update the links between the elements to keep the deque consistant.