Programming Languages

Dr. Michael Petter, Sarah Tilscher WS 2021/22

Exercise Sheet 3

Assignment 3.1 Lockfree vs. locked programming

The purpose of the following exercises is to get acquainted with the pthreads library, locks and lockfree algorithms.

1. Implement the bumper alloc.
2. demonstrate, that allocating memory concurrently produces inconsistent results
3. repair your implementation with
   - lockfree instructions
   - a pthread semaphore

   compare your 3 implementations considering correctness and performance!

Assignment 3.2 Lockfree Algorithms

Given the following data structures:

```
typedef struct node {
  int val;
  struct node* next;
} node;
typedef struct{
  node* top;
} stack;
```

and the following code:

```
void push(stack* s,int i){
  node* newtop  = malloc(sizeof(node));
  newtop->val=i;
  newtop->next = s->top;
  s->top = newtop;
}
```

1. Replace push with your own function push_lockfree, which is made threadsafe, without the use of locks. Instead you may use **lockfree** instructions, e.g.

   ```
   _compare_and_swap (type *ptr, type oldval, type newval)
   ```

   This method performs an atomic compare and swap. That is, if the current value of *ptr is oldval, then write newval into *ptr. The content of *ptr before the operation is returned. (type being an arbitrary type)

2. Provide an int* pop_lockfree(stack* s) function, that pops stack s threadsafe lockfree, returning null if the stack is empty and a pointer to the integer at the top of the stack otherwise.

Assignment 3.3 Parallel Programming – Monitors

Find the functions in the `pthreads` library that provide you with semaphores, monitors, and condition variables. Start your research with `pthread_mutex_init`.

Consider the following code, implementing basic functionality for the doubly linked list:

1. Upgrade the queue in `Dqueue.c` to be threadsafe, using a single mutex. Implement the `ForAll` method and use it in `main` to cause a deadlock.

2. Implement the queue using a monitor and show that the deadlock is "resolved" (there is no deadlock anymore).