Problem set 1, Introduction to C and MPI

TDT4200, Fall 2013

Deadline: Tuesday 27.08.2013 at 23.59 Contact course staff if you cannot meet the deadline.

Evaluation: Pass/Fail. You must complete this assignmet in order to submit future assignments.

Delivery: Use It's Learning. Deliver exactly two files:

- *yourusername_ps1.pdf*, with answers to the theory questions
- *yourusername_code_ps1.*{*zip* |*tar.gz* |*tar*} containing your modified versions of the files:
  - *linkedlist.c*
  - *ring.c*

General notes: All problem sets are to be done **INDIVIDUALLY**. Code must compile and run on `clustis3.idi.ntnu.no`. You should only make changes to the files indicated. Do not add additional files or thrid party code/libraries.

Part 1, Theory

Problem 1, General Theory

a) Explain why multi-core processors have become so popular the past few years, despite being harder to program.

b) Briefly describe the four kinds of parallel systems in Flynn's taxonomy (SISD, SIMD, MISD, MIMD), as well as SPMD.

c) Briefly describe the main difference between a shared-memory system and a distributed-memory system.

Part 2, Code

C programming

In this task, you should complete *linkedlist.c*, by implementing several functions. The code provided in the main function shows how these functions should work.

a) Implement the function `print_array_stats()`. This function takes an array and its size as arguments, and should print the sum of the elements, the maximum value, and the average value of the elements to the screen.

b) Implement the function `new_linked_list()`. The function should initialize a linked list with the number of elements given by the first argument, where all the elements should be initialized to the value of the second argument. The function should return a pointer to the first element.

c) Implement the function `print_linked_list()`. The function should print the vaule of all the elements of the linked list provided as the first argument. The second argument indicates whether all elements should be printed on a single line, or if each element should be printed on a separate line. The third argument indicates whether the elements should be printed starting with the first, and ending with the last, or in the opposite order. See the exampel useages in the main function for further details.

d) Implement the function `sum_linked_list()`. The function should return the sum of the elements of the linked list provided as the first argument.

e) Implement the function `insert_linked_list()`. The function should insert a new element with the value of the third argument in the linked list provided as the first argument. The position of the new element after insertion is indicated by the second argument. You may assume that the second argument is less than the length of the linked list.

f) Implement the function `merge_linked_list()`. The function should merge the two liked lists provided as arguments to form a single liked list. The elements of the first liked list should become the odd elements of the new linked list, while the elements of the second linked list should become the even elements. You may assume that the linked lists have the same length. The first argument should point to the first element of the new linked list.

g) Implement the function `destroy_linked_list()`. The function should free the memory of the linked list provided as argument.

MPI programming

In this task, you should complete the file *ring.c*. The program should implement the ring comunication pattern in MPI, that is, each process should receive a value from another process, increment it by the value of its own rank, and send it on to the next, with the last process sending the value back to the first. When a process receives the value, it should print its rank, the rank of the process it received the message from, and sent it to, the value it received and the value it sent. With three processes, the output should therefore be:

```
Rank 1 received 0 form rank 0 and sent 1 to rank 2
Rank 2 received 1 from rank 1 and sent 3 to rank 0
Rank 0 received 3 from rank 2
```

Additional details can be found in the recitation slides for this problemset