Modulo parallel loop

1 Modulo parallel loop

This exercise is about parallelizing a simple loop that updates all the coefficients of an array `x` of size `n`:

```
for(i=0; i<n; i++) x[i] = update(i, nthreads);
```

All the iterations of this loop are independent. The only constraint is that the loop has to be parallelized using `nthreads` threads in such a way that thread `t` does all the iterations `i` such that `i%nthreads==t`.

2 Package content

In the `modulo_parallel_loop` directory you will find the following files:

- `main.c`: this file contains the main program that executes above loop and prints on the screen its duration. **Only this file has to be modified for this exercise**.
- `aux.c, aux.h`: these two files contain auxiliary routines and **must not be modified**.

The code can be compiled with the `make` command: just type `make` inside the `modulo_parallel_loop` directory; this will generate a `main` program that can be run like this:

```
$ ./main n nthreads
```

where `n` is the number of coefficients in the `x` array and `nthreads` the number of threads to be used.

3 Assignment

- ⌨ parallelize the loop as requested above, i.e., in such a way that thread `t` does all the iterations `i` such that `i%nthreads==t`.