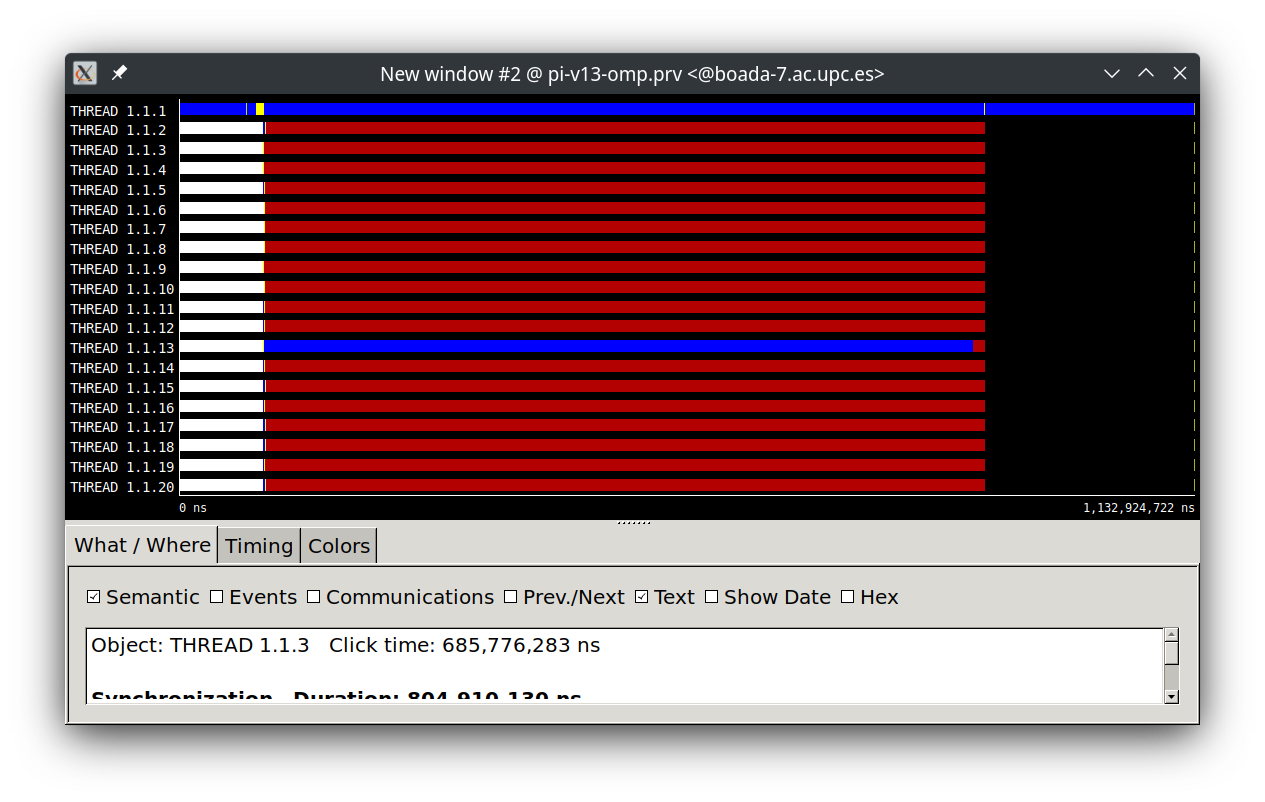
**LAB2 SESSION 2: A very practical introduction to OpenMP (Part II)**

paraver v13

****

**1.1 OpenMP questionnaire (Day 2: explicit tasks)**

* 1.single.c

1. What is the nowait clause doing when associated to single?
2. Then, can you explain why all threads contribute to the execution of the multiple instances of single? Why those instances appear to be executed in bursts?

* 2.fibtasks.c

1. Why all tasks are created and executed by the same thread? In other words, why the program is not executing in parallel?
2. Modify the code so that tasks are executed in parallel and each iteration of the while loop is executed only once.
3. What is the firstprivate(p) clause doing? Comment it and execute again. What is happening with the execution? Why?

* 3.taskloop.c

1. Which iterations of the loops are executed by each thread for each task grainsize or num tasks specified?
2. Change the value for grainsize and num tasks to 5. How many iterations is now each thread executing? How is the number of iterations decided in each case?
3. Can grainsize and num tasks be used at the same time in the same loop?
4. What is happening with the execution of tasks if the nogroup clause is uncommented in the first loop? Why?

* 4.reduction.c

1. Complete the parallelisation of the program so that the correct value for variable sum is returned in each printf statement. Note: in each part of the 3 parts of the program, all tasks generated should potentially execute in parallel.

* 5.synchtasks.c

1. Draw the task dependence graph that is specified in this program
2. Rewrite the program using only taskwait as task synchronisation mechanism (no depend clauses allowed), trying to achieve the same potential parallelism that was obtained when using depend.
3. Rewrite the program using only taskgroup as task synchronisation mechanism (no depend clauses allowed), again trying to achieve the same potential parallelism that was obtained when using depend.

**1.2 Overheads**

Please explain in this section of your deliverable the main results obtained and your conclusions in terms of overheads for parallel, task and the different synchronisation mechanisms. Include any tables/plots that support your conclusions.