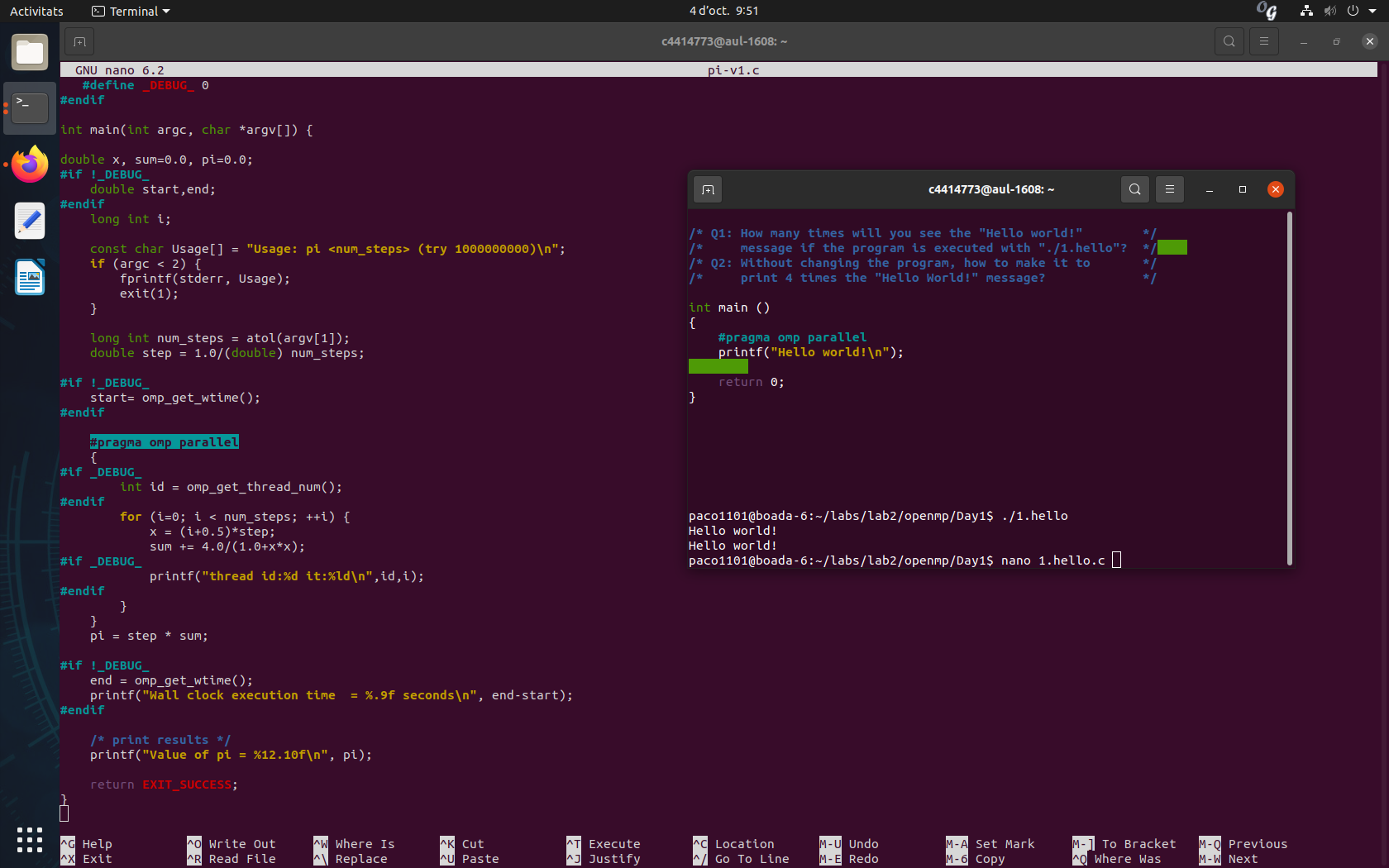
1. Do the 4 programs benefit from the use of several processors in the same way? Can you guess the reason for this behaviour?

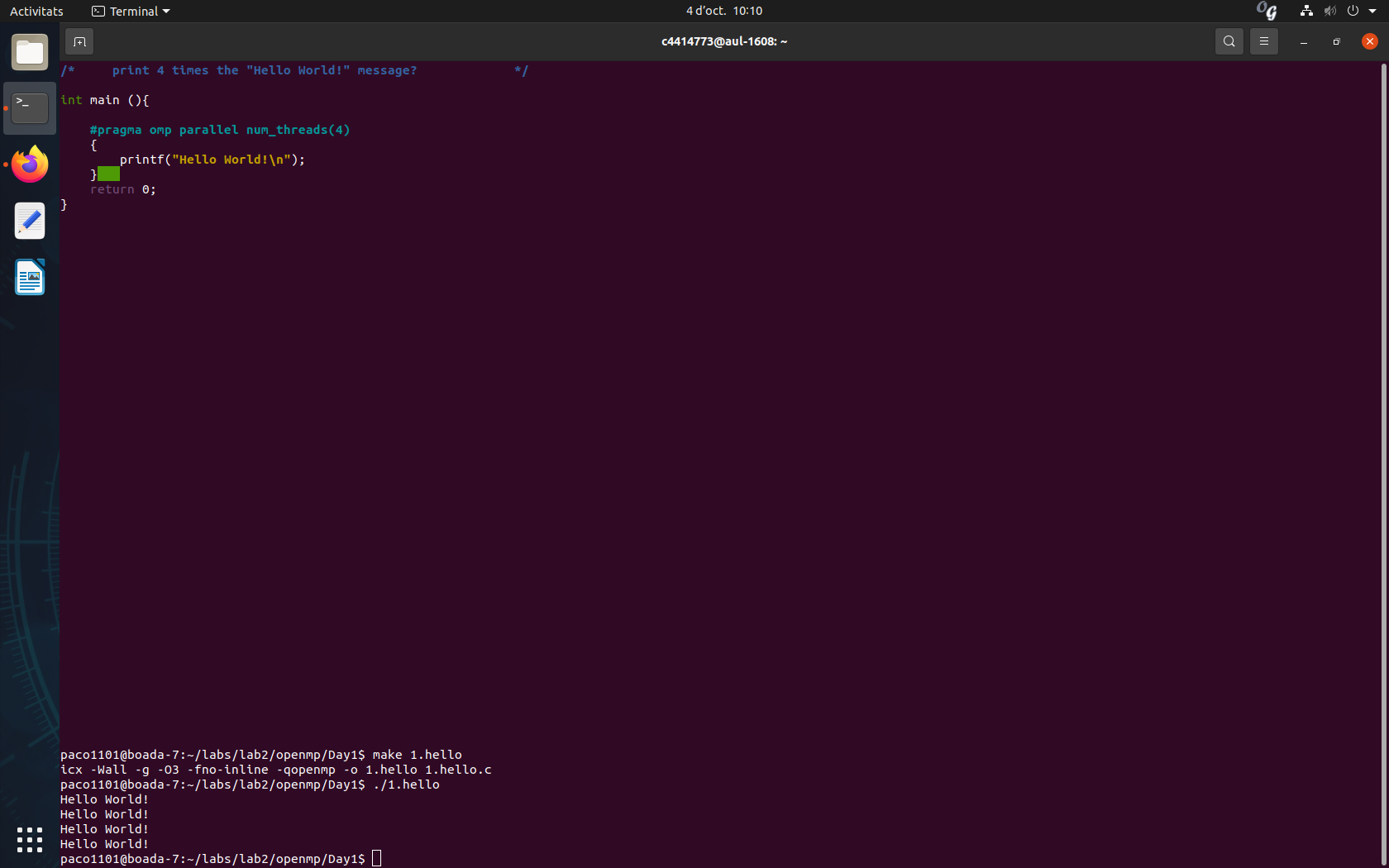
2. For versions pi omp critical.c and pi omp atomic.c and only 1 thread and 10.000.000 itera-

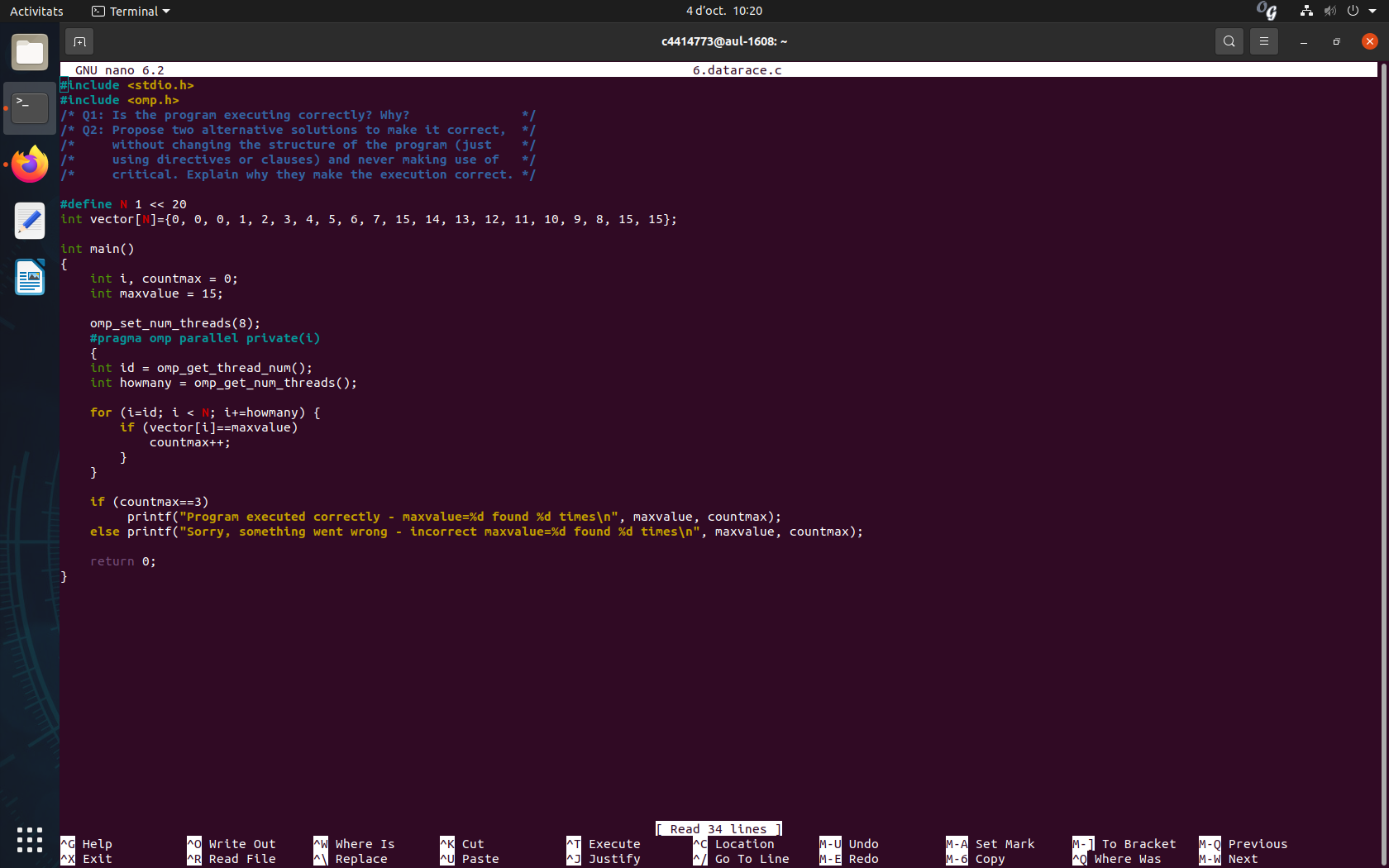
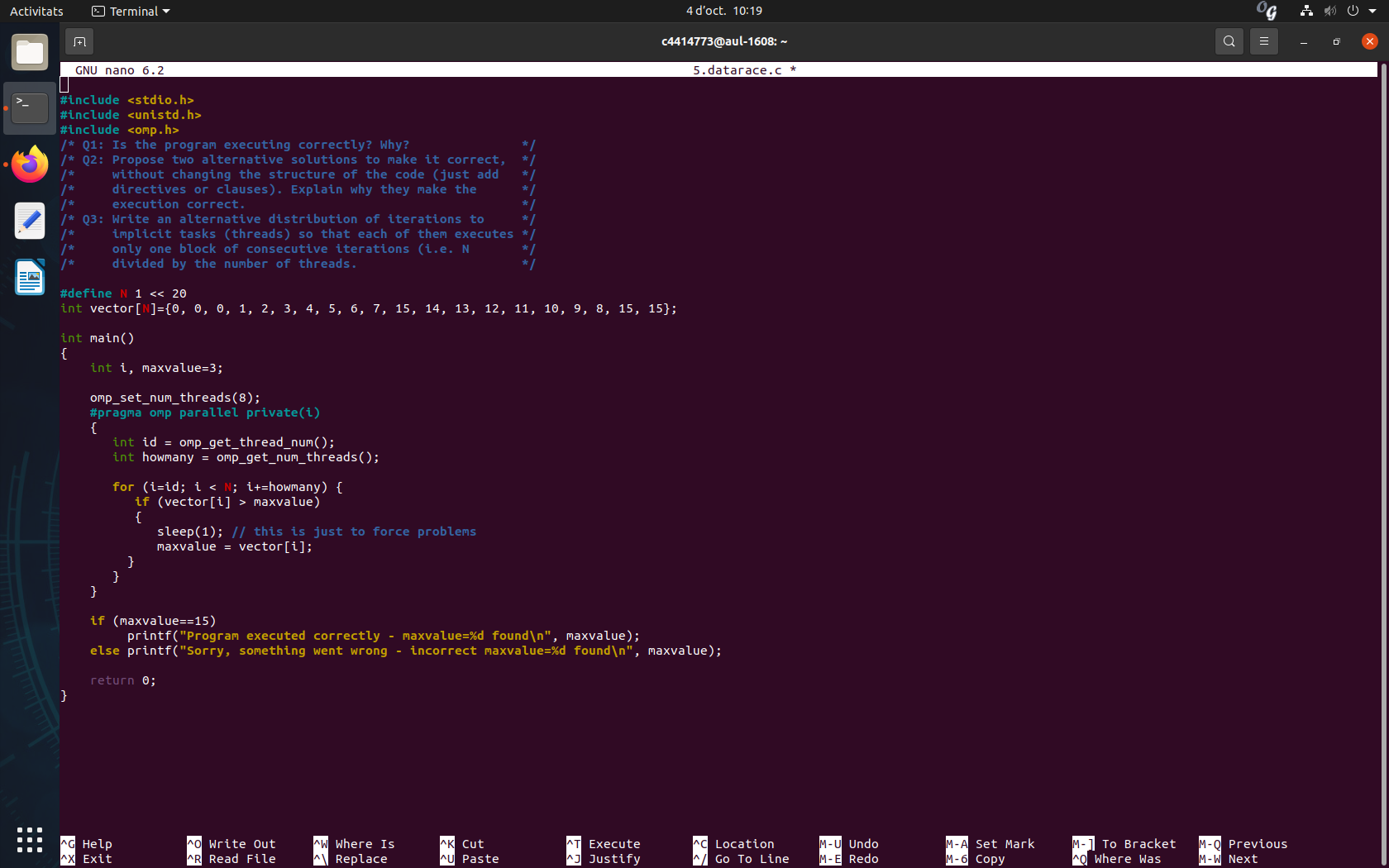
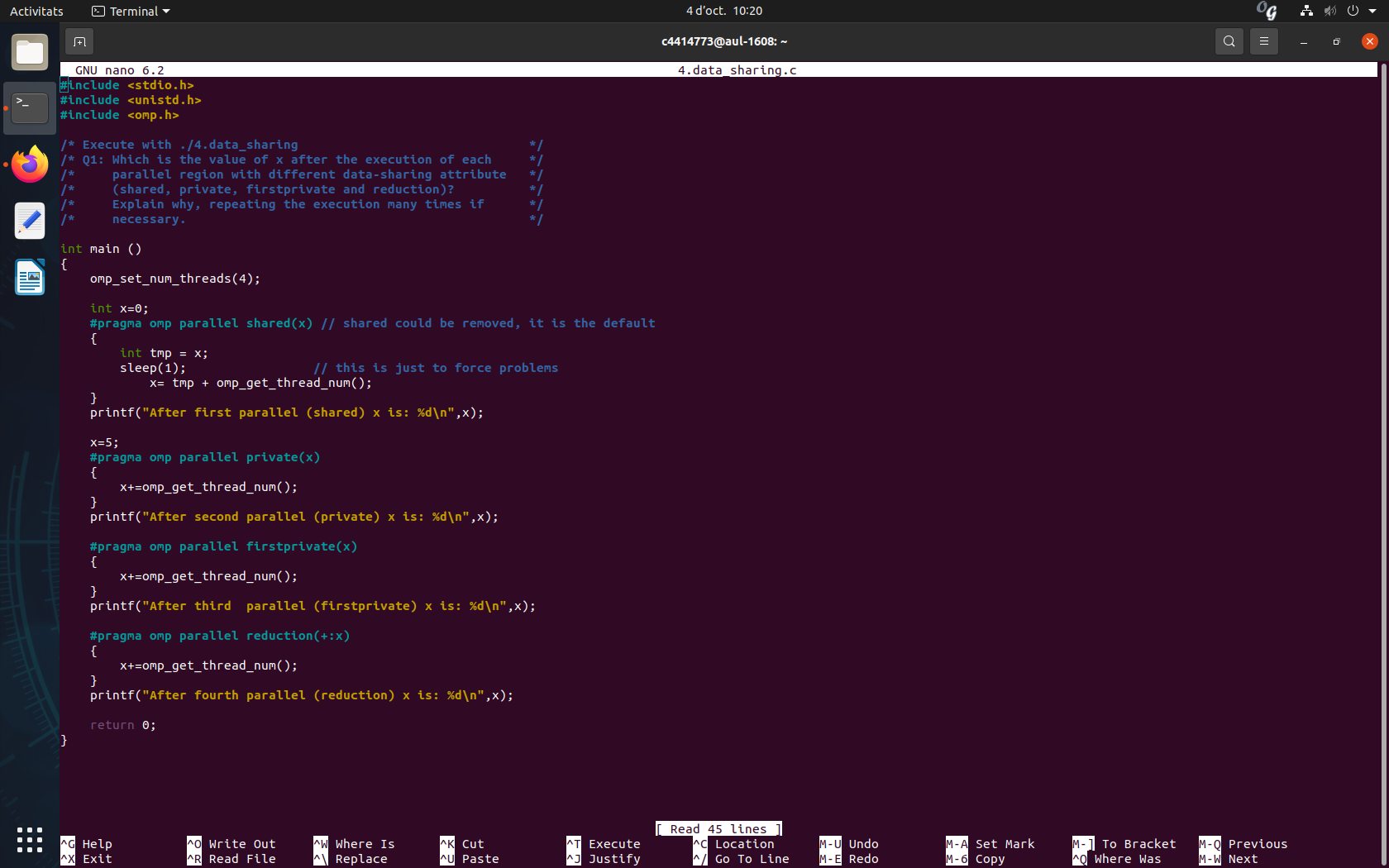
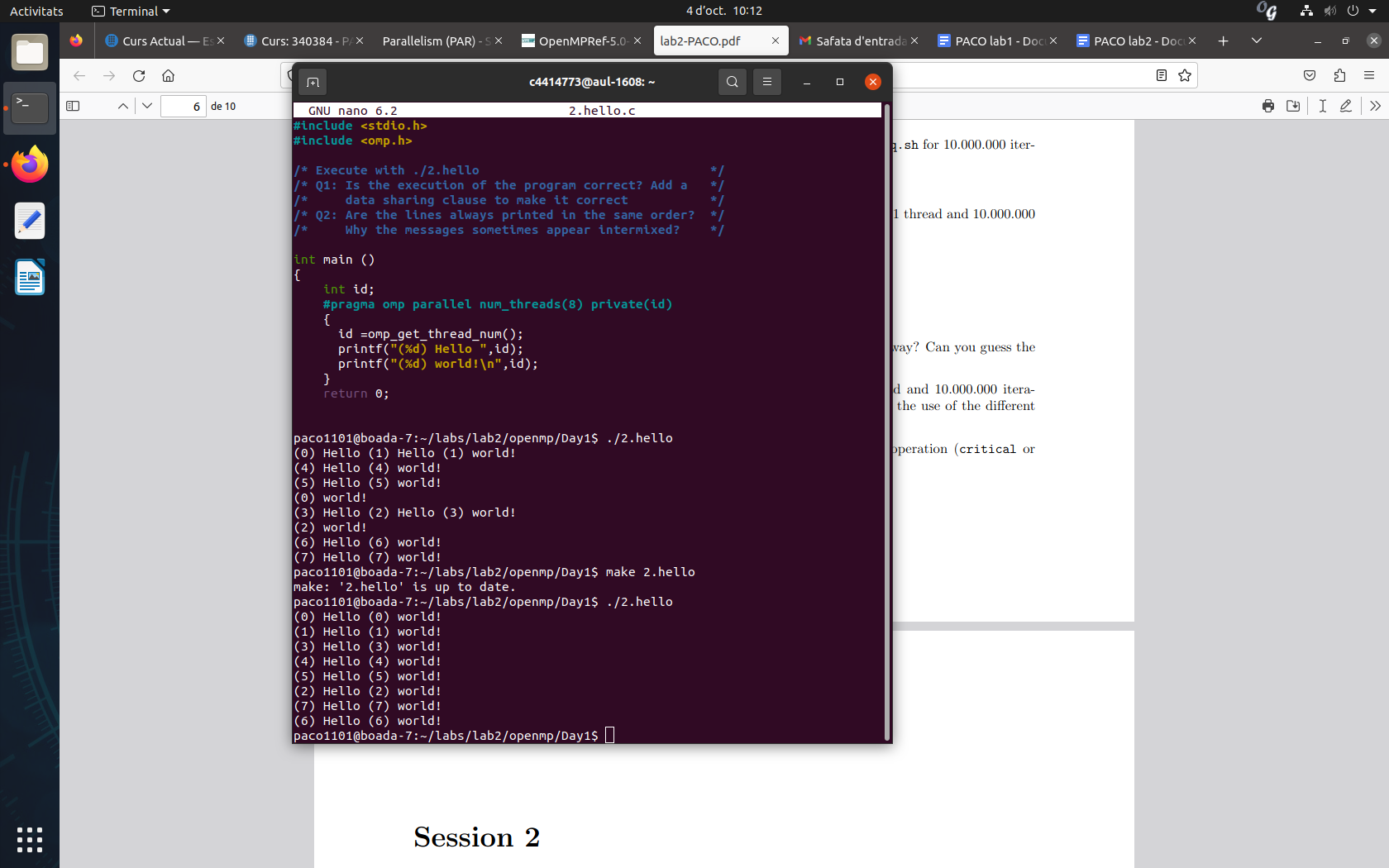
tions, do you notice any major overhead in the execution time caused by the use of the different synchronisation mechanisms?

3. Quantify (in microseconds) the cost of each individual synchronisation operation (critical or

atomic) that is used when executing with 1 and 4 threads.







pragma omp critical ->

pragma omp atomic -> un thread a la vegada (accés exclusiu) però ho fan tots

pragma omp single -> només ho fa un thread

pragma omp taskwait -> espera a la resta de threads abans d’acabar la zona paral·lela

pragma omp taskgroup -> fa el mateix efecte que un taskgroup però amb tot el que hi ha dins {}

pragma omp taskreduction ->

pragma omp taskloop ->

depend -> dependència de sortida de la variable (out: )