

Realización de la practica:

1º-Con la función `__cpuid`, averiguábamos las extensiones que soportaba la CPU de nuestro ordenador.

Con la función, la cual tenía por parámetros, un array de tamaño 4 y un 1, nos devolvía en un array los registros `eax`, `ebx`, `ecx` y `edx` en un array de enteros, y al haber un uno en el parámetro, obteníamos el array del registro `eax`, este tiene como característica, que devuelve unos flags sobre las extensiones soportadas por la CPU, haciendo la operación `and` sobre esos flags, obteníamos la respuesta(1 en el caso de que la soportase).

Operaciones primera fase:

En esta fase realizábamos las operaciones marcadas en la hoja, de forma “normal”, sin operaciones propias de la CPU, en C.

Operaciones segunda fase:

- ❖ Para la operación 1: `Blurr2`, $R_i = (U_i + U_{i+1})/2$
 - Primero creamos un vector de flotantes aleatorios de 64 bits (U_i) y otro para guardar las respuestas(R_i).
 - Definimos las variables `rp1`, `rp2`, `rr`(valor 1, 2 y resultado respectivamente).
 - Con un bucle `for` vamos sumando (gracias a la función, `_mm_add_pd`) cada valor que va tomando `rp1` y `rp2` y lo metemos en `rr`.
 - El array resultado lo dividimos entre dos con la función `_mm_div_pd` y lo volvemos a meter en `rr`.
- ❖ Para la operación 2: `ContarNegativos`:
 - Fuimos recorriendo el vector `U`
 - Gracias a la función `_mm_cmplt_pd`, haciendo un `and`, ya obteníamos si el número era negativo o no.
 - Devolvíamos un contador con el número de negativos en dicho array.
- ❖ Para la operación 3: `or`
 - Entre los elementos de dos arrays (U_i y V_i), realizábamos la operación `or`, y lo guardábamos en un tercer array (S_i).
 - Utilizamos la función `_mm_or_pd` la cual realiza la operación `or` directamente.

Sendos proyectos fueron ejecutados y medimos el tiempo como en la práctica de la sesión1-2 del laboratorio, con la salvedad de que utilizamos la función `queryPerformanceFrequency()` en vez de `clock_gettime()`.

Distribución de tareas:

Todo el trabajo se realizó en el aula de estudio de la facultad.

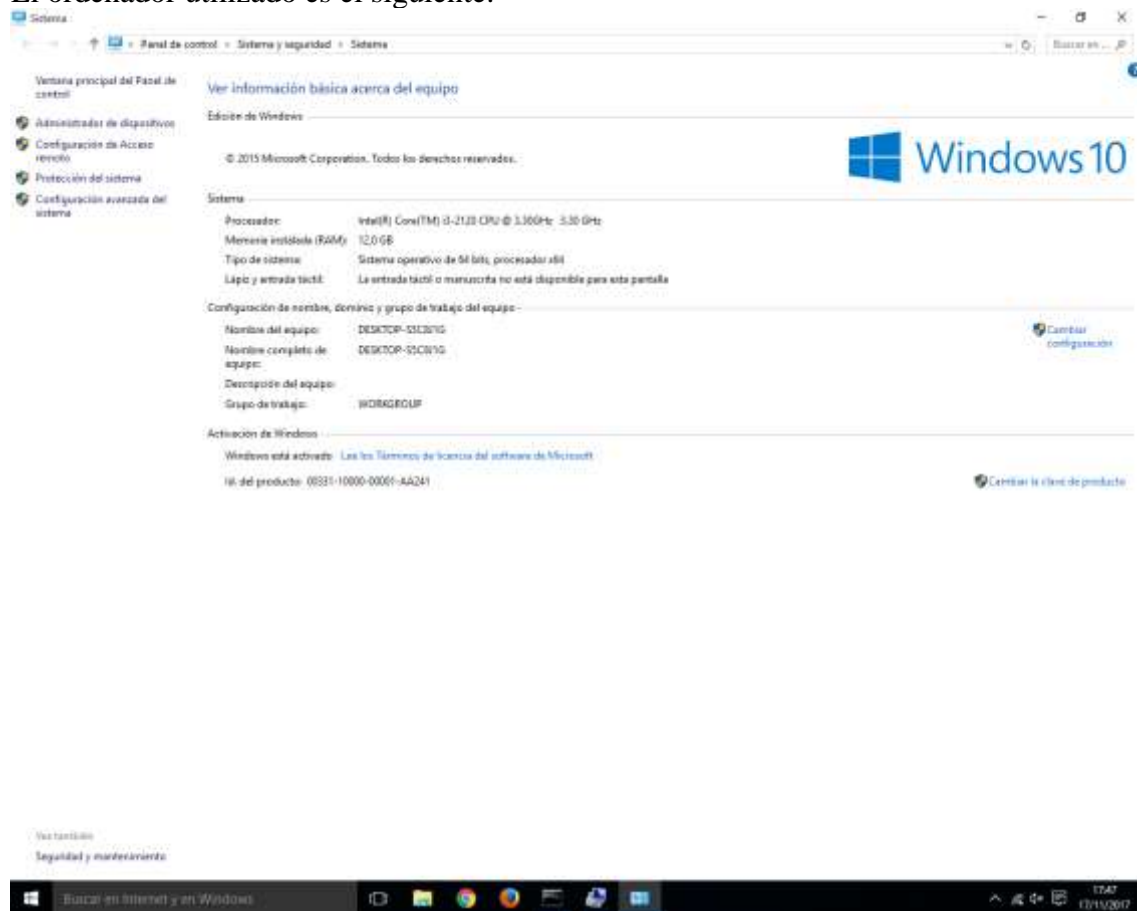
La primera fase, se realizó entre todos los componentes, salvo ciertas formulas y algún problema con la sintaxis de C (no estamos acostumbrados a dicho lenguaje), no fue excesivamente complicado. Cada uno en su ordenador iba realizando un método y ayudándose entre sí, cuando teníamos un método correcto, lo copiábamos en una hoja de Google docs en línea una vez que tuvimos todos, se copiaron en un ordenador todos juntos.

La segunda fase tuvo más complicación, requirió de todo el grupo cohesionado, pensando en el mismo método a la vez, se planteaban posibilidades, se probaban distintas funciones de la página de Intel hasta conseguir dar con el resultado deseado, todas se escribieron en el mismo ordenador en el que estaba la fase anterior, así con cada función.

La tala de Excel fue realizada a la vez que se medían la funciones, uno ejecutaba y media mientras que otro copiaba lo que le dictaban en la tabla.

La cohesión y compenetración del grupo fue tal que no se puede destacar la participación sobresaliente de alguien en específico.

El ordenador utilizado es el siguiente:



Trabajo realizado por:

Autores: Luis Martínez Moreno (UO258273), Alejandro Rodríguez Álvarez (UO232163), Jonathan Vallés Robla (UO231850)