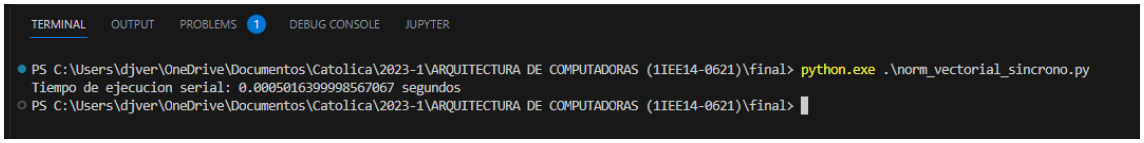
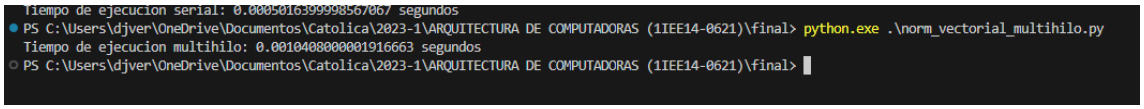
a) Se implemento el programa con el nombre correspondiente de manera sincrona y se adjunta el tiempo de ejecución

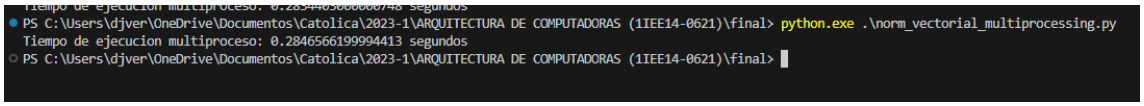


Se realizó 5 medidas consecutivas del calculo para luego sacar el promedio de estas y tener una medida mas significativa para poder comparar con los otros códigos.

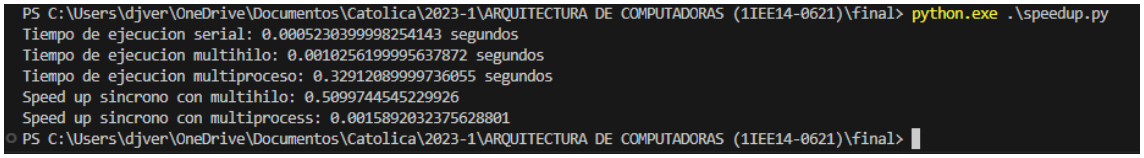
b) Se implementó el programa nuevamente pero ahora en multihilo y se adjunta tiempo de ejecución



c)Se implemento el programa ahora en multiprocessing y se adjunta el tiempo



d) Ahora se haya el speedup correspondiente de las implementaciones, el síncrono con respecto a al multihilo y al multiprocessing



e) Comentarios y Conclusiones

Después de hacer respectivas mediciones al pc utilizado para el código se puede observar mediante el uso de -m mprof run que la cantidad óptima para la CPU de 24 núcleos y 32 hilos es de 8 ya que mayores a estos se ve que el tiempo incrementa considerablemente. Esto se puede lograr al aplicar iteraciones cambiando el pool process principalmente para el multiprocessing e iterando un par de veces para sacar la media.

Lastimosamente en mis pruebas no se ha podido apreciar una mejora al pasar del código síncrono al multihilo y multiprocesing . Esto se debe principalmente por 2 posibles causas:

1. Por mala implementación de esto, en la parte del threathing se utilizo varios for en vez de utilizar un pool el cual directamente podría dar los valores de manera vectorial para no estar iterando ya que estas iteraciones generan más tiempos de espera.

2. Por ser un calculo relativamente sencillo para el cpu que poseo, anteriormente con otros ejemplos del profesor también pasó algo similar ya que al llamar mas núcleos o hilos termina aumentando el tiempo por cargar variables y abrir otros interpretes

Se adjuntan imágenes de los Consumos de memoria footprint para cada una de las implementaciones. Podemos observar que el que mas consumió memoria es el multiprocessing incluso limitándolo a solo 8 procesos ya que tiene que cargar su propia memoria y el intérprete de Python.