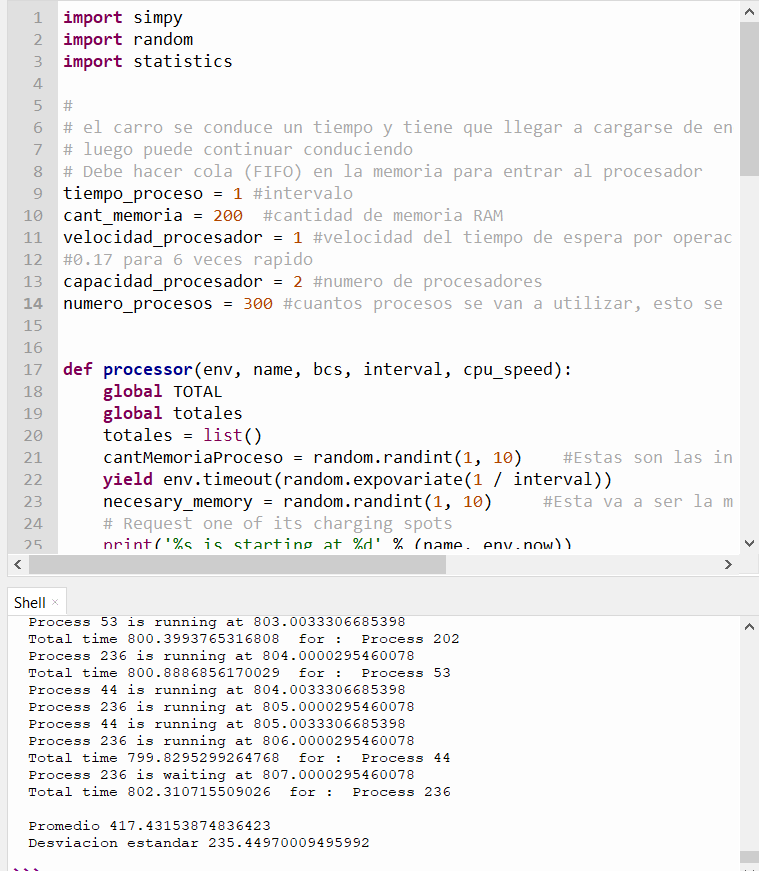
**Estrategia reducción tiempo y análisis posterior**

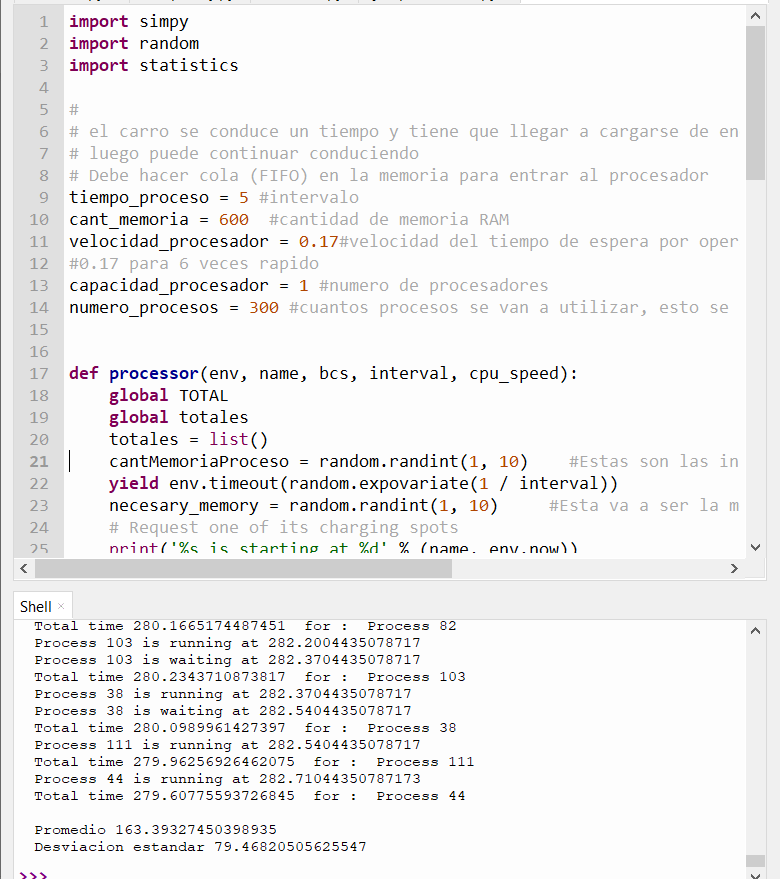
Lo que se observó al analizar los datos de estrategias para disminución de tiempo separadas por intervalo de tiempo de llegada fue, que en general la desviación estándar tiende a disminuir al momento de reducir el intervalo de llegada de los procesos. Se observó también que el tiempo promedio de la estrategia de aumentar la capacidad de memoria tiende a ser ligeramente mayor a la estrategia por defecto, pero luego la estrategia de aumento de memoria cuando se utilizan mas procesos posee un tiempo menor a la corrida de forma estándar, el incremento de tiempo se debe a que al inicio con pocos datos para lograr guardar y sacar un dato se debe de recorrer toda la memoria, la cual es de capacidad grande, para terminar la acción. Luego se vio que con mayor cantidad de procesos el tiempo disminuye debido a que esto evita como un tipo “Bottle neck”, porque los datos no se acumulan en cola para poder entrar a la memoria, causando así una reducción de tiempo. Finalmente se observa que las estrategias de doble procesador son las que poseen menor tiempo y desviación, aunque la estrategia de procesador rápido tendía a ser más rápida que la de doble procesador. Por lo tanto, al haber observado los comportamientos anteriores se llegó a dos estrategias para la reducción de tiempo basada en recursos.

La primera estrategia esta diseñada para ser optima al inicio y al final sin tener que gastar demasiados recursos (dinero), esta involucra tener un doble procesador y una memoria grande ya que con esto se puede emplear hasta cierto punto un tipo de paralelismo el cual nos permitiría realizar múltiples operaciones al mismo tiempo, con esto se aumenta el rendimiento del tiempo globalmente, pero para balancear y disminuir aun mas el tiempo cuando se tienen muchos procesos se debe utilizar más memoria para poder trabajar de una manera eficiente grandes cantidades de procesos sin tener que estos deban de realizar cola para poder entrar a la memoria.



*(Ejemplo de corrida de primera estrategia)*

La segunda estrategia esta diseñada suponiendo un limite inexistente de recursos. Aquí se toma la utilización de un procesador más rápido con una memoria grande proporcional a la velocidad del procesador para evitar el bottle neck, se dice que es suponiendo que no hay limite de recursos ya que para obtener un procesador 6 veces más rápido a lo convencional significa un gasto fuerte de dinero. El procesador rápido causa que los datos se procesen de forma secuencial 6 veces más rápido, y para disminuir la variación de esta combinación y el tiempo con grandes cantidades de procesos se usa una memoria proporcional grande. Debido a lo anterior explicado se concluye que esta sería la estrategia más rápida.

  
*(Ejemplo de corrida de segunda estrategia)*

**Debido a que es una simulación de algo existente, el análisis y estrategias se tomaron desde el punto de vista como que si yo fuese contratado para dar estrategias de reducción de tiempo**