

En ella cada proceso tiene un total de tickets aleatorio, también omití la parte en la que un proceso puede esperar a que el otro le traspase tickets, me pareció de por si confuso implementarlo jeje

Lo único que no pude evitar es que diera finales superiores al tiempo estimado por proceso, por lo que la implementación tiene tiempo muerto en determinados espacios

A	120
B	89
C	115
D	116
E	107
F	106
G	92
H	98

A	inicio: 90	final: 714	ETA: 624	tickets: 51
B	inicio: 170	final: 563	ETA: 393	tickets: 92
C	inicio: 0	final: 533	ETA: 533	tickets: 94
D	inicio: 269	final: 831	ETA: 562	tickets: 21
E	inicio: 40	final: 327	ETA: 287	tickets: 88
F	inicio: 10	final: 848	ETA: 838	tickets: 17
G	inicio: 517	final: 851	ETA: 334	tickets: 21
H	inicio: 30	final: 249	ETA: 219	tickets: 76

Luego la implementación de la retroalimentación multinivel la pensé más sencilla, aunque siento que no tiene inanición, usé para ella un quantum fijo en cada pila haciendo que cada proceso no se pueda ejecutar más de cierto tiempo, luego de eso su prioridad baja para darle paso a los demás procesos.

La prioridad inicial y el tiempo de llegada se declaran desde un inicio.

```
C:\Users\ghero\OneDrive\Documentos\GitHub\sistop-2023-2\proyectos\2\ArcigaFernando
>python multinivel.py

Lista de procesos:
proceso llegada duraci3n|prioridad
A      0      96      |  0
B      3     119      |  3
C      8     119      |  1
D     15      97      |  1
E      5     102      |  3
F      8     106      |  3
G     27      90      |  3

Planificador:
ACDAABEFGCDD

Tabla de ejecuci3n
Proceso inicio  ETA    final
A      0      224    224
B     224     128    352
C      32     832    864
D      96     832    928
E     352     128    480
F     480     128    608
G     608     128    736
```

Con respecto al cuestionario

Loteria:

- ¿C3mo se compararía este método con los otros abordados?

R: Ca3tico para procesos que no llegan de manera s3ncrona, pero bastante justo ya que todos terminan en rangos casi esperados

- ¿Para qu3 tipo de carga es m3s apto y menos apto?

R: no se que responder aqu3

- ¿Qu3 tan susceptible resulta a producir inanici3n?

R: bastante por la aleatoria elecci3n

- ¿Qu3 tan \_justa\_ ser3a su ejecuci3n?

R: un 70% ya que todos reciben tiempo generoso para su ejecuci3n, pero todav3a llega a haber tiempo muerto cuando el rifador no asigna procesos con tiempos grandes para ejecutar

- ¿Qu3 modificaciones requerir3a para planificar procesos con necesidades de tiempo real? (aunque sea \_tiempo real suave\_)

R: Para tiempo real duro se puede utilizar bien ya que suelen ser pocos procesos o procesos cuasi perpetuos, para tiempo real suave tampoco tendría muchos problemas con dichos espacios entre procesos para seleccionar al siguiente

- ¿Alguna otra reflexión o inquietud que les provoque?

R: al principio de mi implementación todos se ejecutaban a la par, no se que hice jajajaja

Multilevel Feedback:

- ¿Cómo se compararía este método con los otros abordados?

R: Fácil de programar, pero replantearía la proyección de los desarrolladores al pedir prioridades a un sistema operativo

- ¿Para qué tipo de carga es más apto y menos apto?

R: Creo que se refiere a carga alta/baja, yo creo que este es bueno para carga alta ya que (al menos en mi versión) va reduciendo la prioridad dando ejecución a todo sin llegar a detenerse del todo en ningún momento.

- ¿Qué tan susceptible resulta a producir inanición?

R: poco ya que los procesos dan paso a los siguientes para que se puedan ejecutar siempre y cuando se avise de su tiempo total de ejecución, que en otro caso sería como la implementación de la lotería donde hay poco tiempo muerto entre cambio de procesos debido al Quantum asignado

- ¿Qué tan \_justa\_ sería su ejecución?

R: Mucho ya que va reduciendo todo hasta ejecutar al menos una vez entre procesos de diferentes prioridades

- ¿Qué modificaciones requeriría para planificar procesos con necesidades de tiempo real? (aunque sea \_tiempo real suave\_)

R: Al menos para mi implementación se puede solo aumentar el Quantum para facilitar la ejecución de lo importante primero

- ¿Alguna otra reflexión o inquietud que les provoque?

R: Me fue más fácil entender su funcionamiento luego de comenzar a programarlo