

Sistemas Operativos

Planificadores

Tarea #2

Barrero Olguín Patricio
Espino Rojas Héctor Daniel.



Objetivo:

Para esta tarea, les pido que escriban un programa que genere varias cargas aleatorias, y compare el resultado sobre varias ejecuciones. Les pido que presenten unas cinco ejecuciones, para poder comparar las tendencias — ¡Y revisen manualmente por lo menos algunos de los resultados, para confirmar que son correctos!

Requerimientos:

¿Cómo se calificará esta tarea?

- Toda entrega que muestre trabajo y se acerque al planteamiento, aunque no funcione, tiene asegurada una calificación mínima de 6
- Si replica exitosamente el primer ejemplo (sólo tabla de resultados), 7.5
- Si presenta el esquema visual, 9
- Si desarrollan un algoritmo de colas múltiples (retroalimentación multinivel/FB, ronda egoísta/SRR), pero sólo lo presentan como tabla, 9
- Si presenta el esquema visual y además desarrollan un algoritmo de colas múltiples, 10

¿Les parece buena motivación? 😊

Al implementar los algoritmos, verán que hay puntos donde hay más de un curso válido de acción. Por ejemplo, en el ejemplo que les dí, con RR1: ¿Es correcto ABABCABC... o ABACBACB...? Ustedes, como implementadores, pueden decidir — Ambos cumplen con el planteamiento formal.

Descripción

La tarea #9 consistió en crear un módulo de pruebas -de simulación- aleatorias de los distintos planificadores además de las implementaciones de cada uno de estos.

Para la implementación de las posibles conceptualizaciones -tanto naturales y las artificiales creadas para un mejor manejo del sistema- se trabajó bajo el paradigma POO en el lenguaje de programación Python3. Aunado a lo anterior, se siguió el estilo de guía de Python PEP8.

El proyecto consta -en una visión de alto nivel- de procesos, cuyo manejo será responsabilidad de la clase Manipulador. Además de esto, el Manipulador realiza la liberación de procesos -con base en sí el proceso ya puede ser ejecutado-, permite realizar cargas predefinidas o crear cargas aleatorias de procesos. También Manipulador hará muestra de los efectos de los planificadores tic a tic.

Para la estructura de los planificadores se utilizó una clase abstracta de nombre Planificador. Dicha clase definirá el comportamiento de los diferentes planificadores mediante la creación del método siguiente_proceso, el cual retorna el próximo proceso a ejecutar junto con el tiempo que este se encontrará en estado ejecución.

Se definieron los siguientes planificadores y sus consideraciones:

Generales:

- Los asteriscos representan espacios donde ningún proceso se encontraba listo para ejecutarse ni en estado de ejecución.
- Las métricas T, E y P, es posible que el resultado varíe contra los obtenidos de una calculadora debido al manejo de decimales en Python. Importante notar, serán bastante similares.
- Las rondas, número de procesos y tiempo total son totalmente adaptables a conveniencia del usuario.
- El tiempo inicial -en generación aleatoria- de procesos se calcula mediante una distribución binomial. Ya que, de este modo, aumentamos la probabilidad de que los procesos se encuentren en disponibles para su ejecución en la primera mitad del tiempo.
- Al final de las pruebas aleatorias se incluyó una prueba con las condiciones iniciales del ejercicio muestra del profesor. Demostrando de este modo el funcionamiento de los planificadores.

FCFS:

Ninguna.

SPN:

Ninguna.

FB:

Al ser instanciado recibe una función que determina el quantum del proceso a ejecutar.
Trabajando en función de la cola en que se encuentre el proceso por el momento.

RR:

Su constructor recibe el número de tics que dura un pato -quantum-. Es imperativo notar que para nuestra implementación, primero se añade a la cola los nuevos procesos y hasta el final el anterior proceso en haberse ejecutado.

SRR:

Su constructor recibe el número de tics que dura un pato -quantum- y los parámetros a y b. Los anteriores son respectivamente: Constante de crecimiento de la cola de procesos nuevos y la constante de crecimiento de la cola de aceptados.

Esta clase hereda de RR el comportamiento y atributos necesarios para realizar RR en la cola de aceptados. E implementa funcionamiento propio para la cola de nuevos.

Un detalle propio de nuestra implementación es que, en caso de añadirse múltiples procesos al mismo tiempo, el primero de estos será añadido a la cola de aceptados y el resto a la cola de nuevos.

En caso de que un proceso de la cola de nuevos llegase a superar en prioridad el nivel de los aceptados, esta clase es resiliente a este caso y utiliza el valor máximo de la cola de aceptados como la prioridad de la cola.

Requerimientos:

*Python =3.6.9

*Módulo scipy y numpy de Python.

Instalación de los módulos (para python3 en linux/mac os):

pip3 install scipy

pip3 install numpy

Utilización de modulos externos:

Utilizados para facilitar la creación de procesos aleatorios

Ejecución

linux/mac os:

python3 planificador.py

windows:

python planificador.py

Pruebas de escritorio:

```
Applications
vie, mar 27 12:50 AM
21% 37%

Planificadores: python3

principal@principal-Inspiron-5447:~/Documents/Planificadores$ python3 planificador.py
--Ronda #1--
A: 3, t=8; B: 7, t=2; C: 7, t=2; D: 7, t=2; E: 7, t=4; F: 10, t=2; (tot:20)
**FCFS**
*****AAAAABCCDDEEEFF
Datos: T=9.833333333333334, E=6.5, P=3.8333333333333335
**RR_1**
*****ABCDEFCEAFEEA
Datos: T=11.0, E=7.666666666666667, P=3.625
**RR_4**
*****ABCCDDEEEAAFF
Datos: T=8.833333333333334, E=5.5, P=2.875
**SPN**
*****ABCCDDFFEEEEE
Datos: T=9.5, E=6.166666666666667, P=3.5833333333333335
**SRR_Q_5_A_2_B_1**
*****ABCCDDEEEFF
Datos: T=9.833333333333334, E=6.5, P=3.8333333333333335
**FB_EXPONENCIAL**
*****ABCDEFBCDEFEA
Datos: T=12.666666666666667, E=9.333333333333334, P=4.458333333333333
**FB_CONSTANTE_1**
*****ABCDEFBCDEFEAAA
Datos: T=10.0, E=6.666666666666667, P=3.25
**FB_LINEAL**
*****ABCDEFBCDEFEAA
Datos: T=11.833333333333334, E=8.5, P=4.083333333333333

--Ronda #2--
A: 8, t=3; B: 10, t=5; C: 12, t=3; D: 13, t=6; E: 13, t=3; F: 17, t=5; (tot:25)
**FCFS**
*****AABBBBCCDDDDDEEEFFFFF
Datos: T=9.833333333333334, E=5.666666666666667, P=2.4555555555555557
**RR_1**
*****AABBCDEBCDEFBCDEFBDFDFD
Datos: T=13.166666666666667, E=9.0, P=3.1277777777777778
**RR_4**
*****AABBBBCCDDDEEEBFFFDDF
Datos: T=12.0, E=7.833333333333333, P=2.761111111111111
**SPN**
*****AABBBBCCCEEEFFFFDDDD
Datos: T=9.166666666666667, E=5.0, P=2.1444444444444444
**SRR_Q_5_A_2_B_1**
*****AABBBBCCDDDEEEDFFFFF
Datos: T=10.0, E=5.833333333333333, P=2.4277777777777776
**FB_EXPONENCIAL**
*****AABCDEBBFCDDDEEFFBDDDDF
Datos: T=12.333333333333334, E=8.166666666666667, P=2.8555555555555556
**FB_CONSTANTE_1**
*****AABCDECDDEFABCDEFBDFBDFD
Datos: T=15.166666666666667, E=11.0, P=3.722222222222222
**FB_LINEAL**
*****AABCDEBBFCDDDEEFFBDDDDF
Datos: T=12.333333333333334, E=8.166666666666667, P=2.8555555555555556 (total)
```

```
Applications
vie, mar 27 12:50 AM 21% 37%

Planificadores: python3

*****AABBCDECFEFABCFBDFBDFD
Datos: T=15.166666666666666, E=11.0, P=3.722222222222222
**FB_LINEAL**
*****AAABCDEBBFCCDDEEFFBDDDDFF
Datos: T=12.333333333333334, E=8.166666666666666, P=2.8555555555555556

---Ronda #3---
A: 5, t=6; B: 6, t=4; C: 8, t=3; D: 8, t=2; E: 9, t=3; F: 11, t=2; (tot:20)
**FCFS**
*****AAAAAABBBBCCDDEEEFF
Datos: T=10.833333333333334, E=7.5, P=4.041666666666667
**RR_1**
*****ABABCDABCFDAEBCFAEA
Datos: T=13.666666666666666, E=10.333333333333334, P=4.361111111111111
**RR_4**
*****AAAABBBBCCDDEEEAAFF
Datos: T=11.5, E=8.166666666666666, P=3.9027777777777772
**SPN**
*****AAAAAADFFCCCEEEBBB
Datos: T=9.333333333333334, E=6.0, P=2.9305555555555556
**SRR_Q_5_A_2_B_1**
*****AAAAAABBBBCCDDEEEFF
Datos: T=10.833333333333334, E=7.5, P=4.041666666666667
**FB_EXPONENCIAL**
*****ABAACDEFBBCCDEEFAAAB
Datos: T=13.0, E=9.666666666666666, P=4.097222222222222
**FB_CONSTANTE_1**

Planificadores: python3
vie, mar 27 12:50 AM 21% 37%

Planificadores: python3

---Ronda #4---
A: 7, t=5; B: 8, t=3; C: 8, t=3; D: 10, t=2; E: 11, t=3; F: 11, t=4; (tot:20)
**FCFS**
*****AAAAABBBBCCDDEEEFFFF
Datos: T=10.0, E=6.666666666666667, P=3.2777777777777778
**RR_1**
*****ABCABDCEFABDCEFAEFAF
Datos: T=13.166666666666666, E=9.833333333333334, P=3.9944444444444445
**RR_4**
*****AAAABBBBCCDDEEEFFFFA
Datos: T=11.666666666666666, E=8.333333333333334, P=3.4861111111111107
**SPN**
*****AAAAADBBBCCCEEEFFFF
Datos: T=9.666666666666666, E=6.333333333333333, P=3.0
**SRR_Q_5_A_2_B_1**
*****AAAAAABBBBCCDDEEEFFFF
Datos: T=10.0, E=6.666666666666667, P=3.2777777777777778
**FB_EXPONENCIAL**
*****ABCDEFABBBCCDEEFFAFAF
Datos: T=12.666666666666666, E=9.333333333333334, P=3.8555555555555556
**FB_CONSTANTE_1**
*****ABCDEFABCFABCEFAFA
Datos: T=13.5, E=10.166666666666666, P=4.041666666666667
**FB_LINEAL**
*****ABCDEFABBBCCDEEFFAFAF
Datos: T=12.666666666666666, E=9.333333333333334, P=3.8555555555555556
```



```
Applications
vie, mar 27 12:51 AM
21% 37%
Planificadores: python3

Planificadores: python3

---Ronda #5---
A: 9, t=8; B: 9, t=2; C: 13, t=2; D: 15, t=4; E: 16, t=3; F: 19, t=6; (tot:25)
**FCFS**
*****AAAAAABBBCCDDDEEEFFFFFFF
Datos: T=10.5, E=6.33333333333333, P=3.166666666666665
**RR_1**
*****ABABACADCEADFEADFEADFAFFF
Datos: T=11.83333333333334, E=7.66666666666667, P=2.819444444444444
**RR_4**
*****AAAABBBCCAAAADDDEEEFFFFFFF
Datos: T=9.83333333333334, E=5.66666666666667, P=2.583333333333335
**SPN**
*****BBAAAAAAACCEEEDDDDFFFFFFF
Datos: T=9.33333333333334, E=5.16666666666667, P=2.444444444444444
**SRR_Q_5_A_2_B_1**
*****AAAABBBAAAACDDDEEEFFFFFFF
Datos: T=10.16666666666667, E=6.0, P=2.875
**FB_EXPONENCIAL**
*****ABAACBDECDDEEFFFAAADFFFA
Datos: T=12.0, E=7.83333333333333, P=2.840277777777778
**FB_CONSTANTE_1**
*****ABABCCDEFFADEFADEFAFAFAA
Datos: T=10.66666666666667, E=6.5, P=2.326388888888889
**FB_LINEAL**
*****ABAACBDECDDEEFFFAAADFFFA
Datos: T=11.66666666666667, E=7.5, P=2.770833333333335
```

Prueba con las condiciones iniciales del ejercicio muestra del profesor:

```
Applications
vie, mar 27 12:51 AM
21% 37%
Planificadores: python3

Planificadores: python3

----Caso Gunnar----
A: 0, t=3; B: 1, t=5; C: 3, t=2; D: 9, t=5; E: 12, t=5; (tot:20)
**FCFS**
AAABBBBCCDDDDDEEEEEE
Datos: T=6.2, E=2.2, P=1.740000000000002
**RR_1**
ABABCABCBDBDEDEDEDEE
Datos: T=7.6, E=3.6, P=1.98
**RR_4**
AAABBBBCCBDDDDDEEEEEE
Datos: T=7.2, E=3.2, P=1.880000000000001
**SPN**
AAACCBBBBDDDDDEEEEEE
Datos: T=5.6, E=1.6, P=1.319999999999998
**SRR_Q_5_A_2_B_1**
AAABBBBCCBDDDDDEEEEEE
Datos: T=6.4, E=2.4, P=1.72
**FB_EXPONENCIAL**
ABAACBBBBDDEEEEDDEE
Datos: T=7.0, E=3.0, P=1.806666666666667
**FB_CONSTANTE_1**
ABABCABBDDEEEDEBDE
Datos: T=9.0, E=5.0, P=2.166666666666667
**FB_LINEAL**
ABAACBBBBDDEEEEDDEE
Datos: T=7.0, E=3.0, P=1.806666666666667
principal@principal-Inspiron-5447:~/Documents/Planificadores$
```