Algoritmos de Planificación de Procesos

En este documento se desarrollará una breve descripción de los algoritmos de planificación de procesos implementados en lenguaje Python, así también como una rápida guía sobre como correr los scripts de dicho lenguaje, para cada tipo de algoritmo.

En la entrega de este Sprint(4), se incluyen cuatro archivos de extensión .py, los cuales pueden ser ejecutados desde la línea de comandos con el lenguaje Python.

Instalación de Python y librerías

En primer lugar, el usuario debe contar con Python instalado en su computadora. Existen diversas formas de hacerlo, así también como diferentes alternativas según el sistema operativo que se utilice:

Windows: Instalador de Python 3

1- Abrir en un navegador el siguiente enlace: https://www.python.org/downloads/windows/
2- Descargar el instalador según el sistema, Windows x86-64 o Windows x86

Linux:

1- Ejecutar el siguiente comando en la terminal:

```
Sudo apt-get install python3.7
```

MacOS:

1- Ejecutar el siguiente comando en la terminal:

```
brew install python3
```

Además de instalar Python, también es necesario instalar una librería llamada tabulate, la cual es utilizada para mostrar la información en forma de tablas por la terminal. Esta librería puede ser instalada de la siguiente manera en la terminal:

```
pip install tabulate
```

Ejecución de scripts

Para ejecutar los scripts incluidos en este sprint, se debe abrir una nueva terminal, en la carpeta o ubicación donde se encuentren dichos archivos. Seguidamente, se utiliza el comando ***python,*** seguido del nombre del archivo de extensión .py que se desea ejecutar:

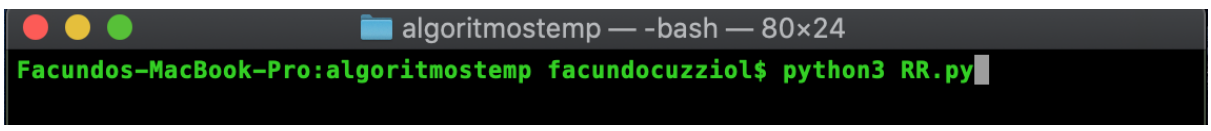


Figura 1: Llamada para ejecutar el algoritmo de Round Robin

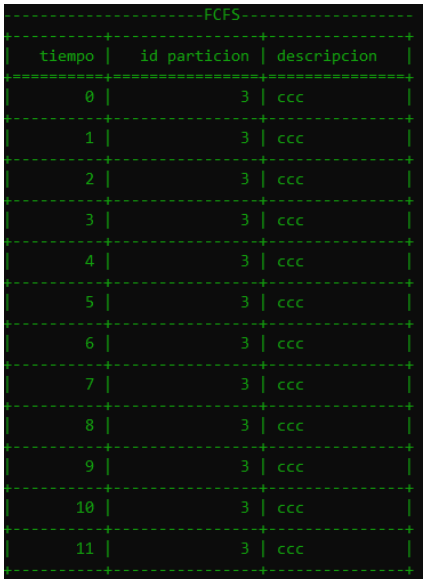
Para la ejecución de los algoritmos, se utilizó una estructura de lista, con algunos procesos cargados, donde cada elemento de la lista es a su vez una lista, con información de dicho proceso:

```
procesos= [[1,'aaa',0,25,20,2],
[2,'bbb',0,130,20,1],
[3,'ccc',3,65,60,0],
[4,'ddd',1,47,7,7],
[5,'eee',0,2,5,2],
[6,'fff',2,59,3,0],
[7,'ggg',2,30,2,8],
[8,'hhh',6,200,7,8]]
```

Lo cual puede ser interpretado de la siguiente manera:

idProceso	Descripción	Prioridad	Tamaño	TI	TA
1	'aaa'	0	25	20	2
2	'bbb'	0	130	20	1
3	'ccc'	3	65	60	0
4	'ddd'	1	47	7	7
5	'eee'	0	2	5	2
6	'fff'	2	59	3	0
7	'ggg'	2	30	2	8
8	'hhh'	6	200	7	8

Algoritmo FCFS: Al ejecutar este algoritmo, podemos obtener la siguiente salida, donde se indica para cada tiempo, el proceso que se ha ido ejecutando:

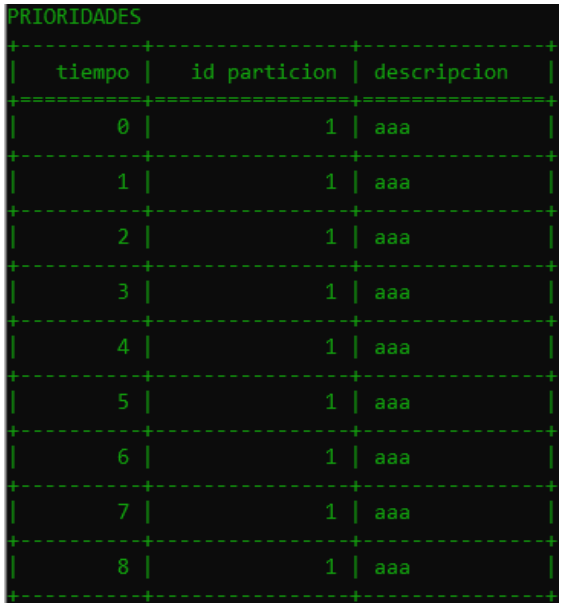


FCFS

tiempo	id particion	descripcion
0	3	ccc
1	3	ccc
2	3	ccc
3	3	ccc
4	3	ccc
5	3	ccc
6	3	ccc
7	3	ccc
8	3	ccc
9	3	ccc
10	3	ccc
11	3	ccc

Figura 2: Muestra de Resultado de FCFS

Algoritmo Prioridades: Al ejecutar este algoritmo, obtenemos la siguiente salida, teniendo en cuenta que una mayor prioridad esta dada por un menor valor del numero de dicho campo. Por ejemplo: el numero 0 representa mayor prioridad que 1, y este ultimo representa mayor prioridad que 2, etc.

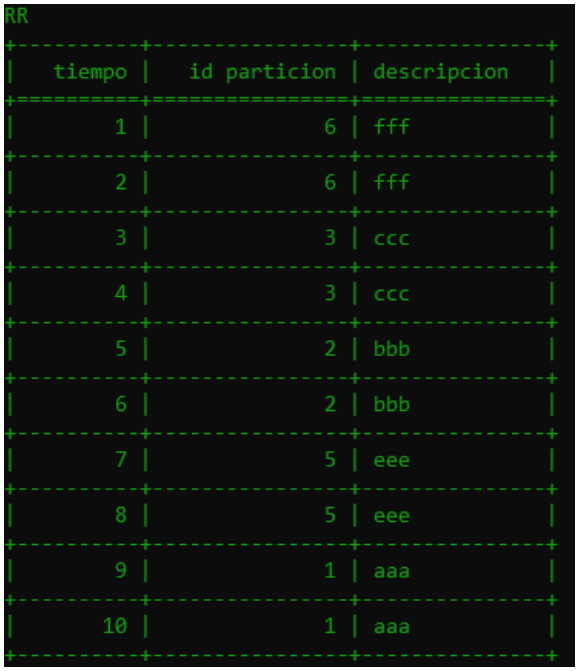


PRIORIDADES

tiempo	id particion	descripcion
0	1	aaa
1	1	aaa
2	1	aaa
3	1	aaa
4	1	aaa
5	1	aaa
6	1	aaa
7	1	aaa
8	1	aaa

Figura 3: Muestra de Resultado de Prioridades

Algoritmo Round Robin (RR): Para este algoritmo, se tuvo en cuenta un valor de quantum igual a 2, tal como se indica en la figura 4, donde cada proceso se ejecuta por 2 unidades de tiempo:

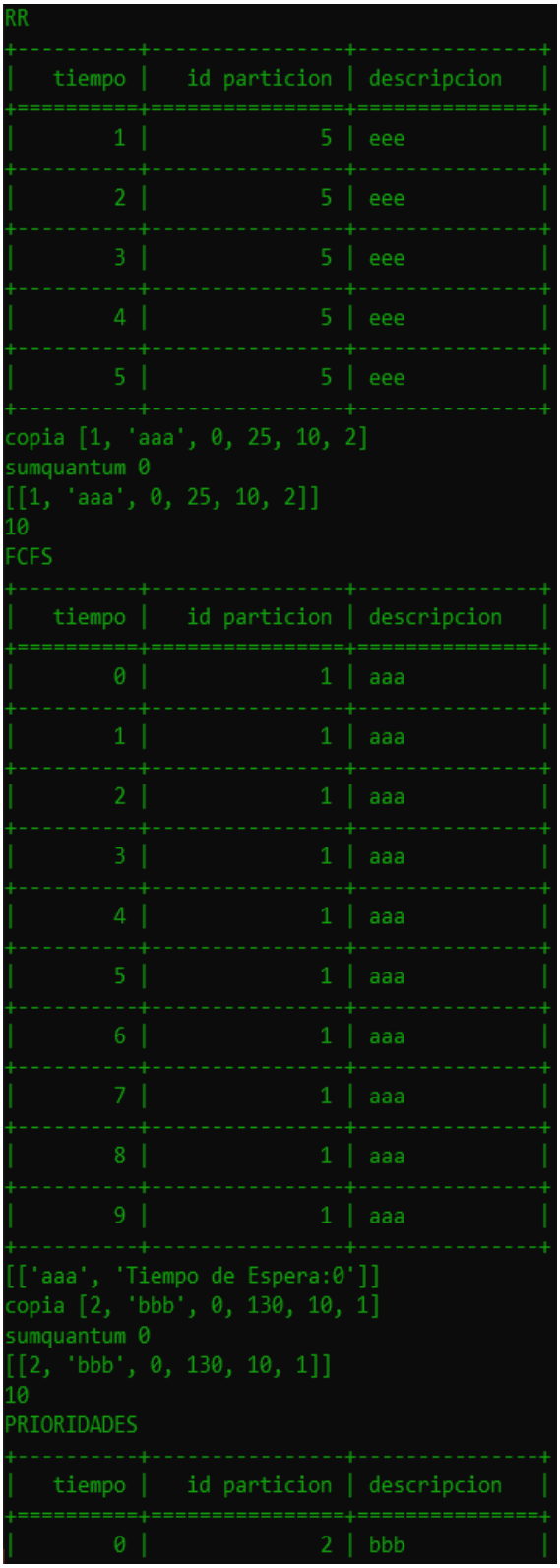


RR

tiempo	id particion	descripcion
1	6	fff
2	6	fff
3	3	ccc
4	3	ccc
5	2	bbb
6	2	bbb
7	5	eee
8	5	eee
9	1	aaa
10	1	aaa

Figura 4: Muestra de Resultado de RR

Algoritmo de Colas Multinivel (MQ): Finalmente, para el algoritmo de colas multinivel, establecimos un máximo posible de 5 colas. Para la figura 5 que se mostrará a continuación, se tuvieron en cuenta 3 colas. La primera, utilizando un algoritmo RR, la segunda con fcfs, y la tercera utilizando un algoritmo por prioridades.



```
RR
+----------+---------------+---------------+
|   tiempo |   id particion | descripcion   |
+==========+===============+===============+
|        1 |              5 | eee           |
+----------+---------------+---------------+
|        2 |              5 | eee           |
+----------+---------------+---------------+
|        3 |              5 | eee           |
+----------+---------------+---------------+
|        4 |              5 | eee           |
+----------+---------------+---------------+
|        5 |              5 | eee           |
+----------+---------------+---------------+
copia [1, 'aaa', 0, 25, 10, 2]
sumquantum 0
[[1, 'aaa', 0, 25, 10, 2]]
10
FCFS
+----------+---------------+---------------+
|   tiempo |   id particion | descripcion   |
+==========+===============+===============+
|        0 |              1 | aaa           |
+----------+---------------+---------------+
|        1 |              1 | aaa           |
+----------+---------------+---------------+
|        2 |              1 | aaa           |
+----------+---------------+---------------+
|        3 |              1 | aaa           |
+----------+---------------+---------------+
|        4 |              1 | aaa           |
+----------+---------------+---------------+
|        5 |              1 | aaa           |
+----------+---------------+---------------+
|        6 |              1 | aaa           |
+----------+---------------+---------------+
|        7 |              1 | aaa           |
+----------+---------------+---------------+
|        8 |              1 | aaa           |
+----------+---------------+---------------+
|        9 |              1 | aaa           |
+----------+---------------+---------------+
[['aaa', 'Tiempo de Espera:0']]
copia [2, 'bbb', 0, 130, 10, 1]
sumquantum 0
[[2, 'bbb', 0, 130, 10, 1]]
10
PRIORIDADES
+----------+---------------+---------------+
|   tiempo |   id particion | descripcion   |
+==========+===============+===============+
|        0 |              2 | bbb           |
```

Figura 5: Muestra de Resultado de Colas Múltiples