



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Ingeniería en Computación

Sistemas Operativos

Tarea 2.

Grupo: 4

Rivas Gil María Lucía

Fecha: 28/10/2025

En esta tarea se realizó un código en python para hacer comparaciones en los distintos tipos de planificadores.

En el código realizado se tienen las funciones que describen el funcionamiento de cada planificador, una función para generar una carga aleatoria de procesos (del proceso A al E), además de una función que copia estos procesos y los agrega como argumento a las funciones de los algoritmos para que tengan los mismos y claro la función para imprimir los resultados en tiempos:

Tiempo de espera: Tiempo que estuvo en espera el proceso hasta que se ejecutó

Tiempo de retorno: Tiempo total del proceso desde su llegada hasta que finaliza

Penalización: La relación entre el tiempo total del sistema y la duración.

Se realizaron 5 ejecuciones y a partir de ahí se hicieron las siguientes relaciones:

Ejecución 1:

```
[mariarivas@MacBook-Malu Documents % python3 comparaPlani.py

- Carga aleatoria:
  A: 2, t=6; C: 9, t=6; B: 11, t=5; D: 11, t=4; E: 14, t=6 (tot:27)

FCFS: T=10.0, E=4.6, P=1.94
      --AAAAAA-CCCCCBBBBBDDDDDEEEEEEE

RR1: T=13.6, E=8.2, P=2.59
      --AAAAAA-CCBDCBDECBDCECBDECBEEE

RR4: T=12.6, E=7.2, P=2.38
      --AAAAAA-CCCCBBBBDDDDCCCEEEEBEE

SPN: T=9.8, E=4.4, P=1.85
      --AAAAAA-CCCCCDDDDBBBBBBEEEEEEE
```

Ejecución 2:

```
[mariarivas@MacBook-Malu Documents % python3 comparaPlani.py

- Carga aleatoria:
  A: 2, t=6; C: 5, t=3; E: 5, t=3; B: 9, t=3; D: 9, t=5 (tot:20)

FCFS: T=8.4, E=4.4, P=2.25
      --AAAAAACCCCEEEBBBDDDDDD

RR1: T=11.2, E=7.2, P=2.92
      --AAACEACEABDCEABDBDDDD

RR4: T=8.8, E=4.8, P=2.19
      --AAAACCCCEEEAABBBDDDDDD

SPN: T=8.4, E=4.4, P=2.25
      --AAAAAACCCCEEEBBBDDDDDD
```

Ejecución 3:

```
[mariarivas@MacBook-Malu Documents % python3 comparaPlani.py

- Carga aleatoria:
  A: 0, t=6; B: 5, t=5; D: 5, t=7; E: 12, t=6; C: 15, t=2 (tot:26)

FCFS: T=9.6, E=4.4, P=2.31
      AAAAAABBBBBDDDDDDDEEEEEEC
RR1:  T=11.6, E=6.4, P=2.32
      AAAAAABDABDBDBEDBECDECEDEE
RR4:  T=11.2, E=6.0, P=2.45
      AAAAAABBBBBDDDDDBEEEEDDDCCEE
SPN:  T=8.8, E=3.6, P=1.78
      AAAAAABBBBBDDDDDDDCCEEEEE
```

Ejecución 4:

```
[mariarivas@MacBook-Malu Documents % python3 comparaPlani.py

- Carga aleatoria:
  A: 1, t=3; C: 1, t=2; E: 1, t=2; B: 5, t=7; D: 11, t=2 (tot:16)

FCFS: T=6.2, E=3.0, P=2.29
      -AAACCEBBBBBBBDD
RR1:  T=6.8, E=3.6, P=2.28
      -ACEACEBABDBDBBB
RR4:  T=6.0, E=2.8, P=2.04
      -AAACCEBBBBBDBBB
SPN:  T=5.8, E=2.6, P=1.95
      -CCEEAABBBBBBBDD
```

Ejecución 5:

```
[mariarivas@MacBook-Malu Documents % python3 comparaPlani.py

- Carga aleatoria:
  A: 6, t=5; C: 8, t=6; E: 9, t=5; B: 10, t=4; D: 12, t=3 (tot:23)

FCFS: T=12.0, E=7.4, P=2.95
      -----AAAAACCCCEEEEEBBBDD
RR1:  T=16.6, E=12.0, P=3.7
      -----AACAECAEDCBAEDCBEDCBEC
RR4:  T=16.6, E=12.0, P=3.68
      -----AAAACCCCEEEEEBBBADDCCCE
SPN:  T=10.2, E=5.6, P=2.11
      -----AAAAABBBBDDDEEEEECCCCC
```

Calculando del promedio de las 5 ejecuciones:

Planificador	T	E	P
FCFS	9.24	4.76	2.348
RR1	11.96	7.48	2.762
RR4	11.04	6.56	2.548
SPN	8.6	4.12	1.988

Tomando en cuenta las tendencias observadas, se podría decir que el SPN es el más eficaz (al menos en este escenario) ya que prioriza los procesos cortos, por otro lado, los que tienen mayor fragmentación para ejecutar los procesos son más lentos y claro que sus tiempos dependen en el patrón de llegada de los procesos y la duración de estos.