



Universidad Nacional Autónoma de México

Facultad de Ingeniería
Sistemas Operativos



Grupo: 8

Tarea: 2 Comparación de Planificadores

Integrante:

321101464 Michelle Ariana Castañeda González

Comparación de planificadores

En la tarea se compararon distintos mecanismos de planificación de procesos del sistema operativo, bajo diferentes cargas aleatorias de trabajo, para analizar su comportamiento y rendimiento.

Se desarrolló un programa en Python denominado `tarea_planificadores.py`, que implementa los siguientes algoritmos:

- FCFS (First Come First Served / FIFO)
- RR (Round Robin) con quantum variable ($q = 1$ y $q = 4$)
- SPN (Shortest Process Next), versión no expropiativa de SJF
- FB (Feedback multinivel) con tres colas de prioridad
- SRR (Selfish Round Robin) o ronda egoísta, que prioriza procesos nuevos

El programa genera aleatoriamente varias rondas para después calcular las métricas promedio de cada algoritmo y muestra la ejecución mediante un diagrama de Gantt textual. Además, se comprobó manualmente la coherencia de los resultados obtenidos.

Descripción del programa

El simulador fue desarrollado en Python 3.11. El código es completamente portable y se ha probado en macOS. No requiere instalación de paquetes adicionales ni configuraciones externas.

Para ejecutar el programa desde terminal:

```
python3 tarea_planificadores.py
```

El programa utiliza estructuras básicas como colas (`deque`), tuplas (`namedtuple`) y diccionarios para controlar tiempos de llegada, ráfagas y estados de ejecución.

Estructura del código

1. Generación de procesos

La función `generar_ronda()` crea listas de procesos con:

- Nombre (A, B, C...)
- Tiempo de llegada aleatorio
- Ráfaga de CPU aleatoria

Cada ronda simula una carga de trabajo diferente, incluyendo posibles huecos de CPU.

2. Cálculo de métricas

La función `calcular_metricas()` obtiene para cada proceso:

- T (Turnaround): tiempo desde la llegada hasta finalizar
- E (Espera): tiempo en cola antes de ejecutarse
- P (Penalización): relación entre T y ráfaga

También calcula el promedio de cada métrica para comparar algoritmos.

3. Planificadores

- **FCFS (First Come First Served)**: ejecuta procesos en orden de llegada, sin interrupciones.
- **RR (Round Robin)**: ejecuta procesos en cola circular por un quantum fijo; si no terminan, regresan al final. Quantum configurable ($q = 1$ o $q = 4$).
- **SPN (Shortest Process Next)**: selecciona el proceso con menor ráfaga entre los listos, no expropiativo.

- **FB (Feedback multinivel):** tres colas de prioridad (RR1, RR2, FCFS). Los procesos nuevos empiezan en la más alta; si no terminan, se degradan.
- **SRR (Selfish Round Robin):** procesos recién llegados se insertan al frente de la cola, obteniendo preferencia; quantum base = 2.

Ejecución

Cada algoritmo genera un diagrama de Gantt y calcula las métricas de los procesos. Se pueden ejecutar múltiples rondas para comparar resultados.

El programa ejecutó cinco rondas aleatorias con distintas cargas de trabajo. En cada una se muestran las métricas promedio (T, E, P) y el diagrama de Gantt.

```
- Ronda 1:
A: 3, t=6; B: 9, t=7; C: 11, t=5; D: 12, t=1; E: 12, t=7 (tot:26)
FCFS: T=10, E=4.8, P=3.29
---AAAAAABBBBBBCCCCDEEEEEEE
RR1: T=11.2, E=6, P=2.31
---AAAAAABBBBCBDECBECECECEEE
RR4: T=12, E=6.8, P=2.94
---AAAAAABBBBCCCCDEEEEBBBCEEE
SPN: T=9.2, E=4, P=2.33
---AAAAAABBBBBBBDCCCCCEEEEEEE
FB: T=10.6, E=5.4, P=2.05
---AAAAAABBBBCDECCEEBBBCEEEEE
SRR: T=12, E=6.8, P=2.59
---AAAAAABBCCEEDBBCEEBBCEEBE

- Ronda 2:
A: 1, t=4; B: 3, t=3; C: 5, t=1; D: 10, t=4 (tot:12)
FCFS: T=4.25, E=1.25, P=1.92
---AAAABBBBC-DDDD
RR1: T=4.5, E=1.5, P=1.81
---AAABABCB-DDDD
RR4: T=4.25, E=1.25, P=1.92
---AAAABBBBC-DDDD
SPN: T=3.75, E=0.75, P=1.25
---AAAACBBB-DDDD
FB: T=4.5, E=1.5, P=1.42
---AAABCBBA-DDDD
SRR: T=4.5, E=1.5, P=1.44
---AABBCAAB-DDDD
```

- Ronda 4:
A: 0, t=7; B: 1, t=6; C: 6, t=4; D: 7, t=4; E: 9, t=2; F: 11, t=2 (tot:25)
FCFS: T=12, E=7.83, P=3.88
AAAAAABBBBBBCCCCDDDEEFF
RR1: T=16.17, E=12, P=4.4
AABABABACBDACEBDFACEBDFCD
RR4: T=13.33, E=9.17, P=4.03
AAAABBBBAAACCCDDDDBBEEFF
SPN: T=9.33, E=5.17, P=2.21
AAAAAAACCCCEEFFDDDDBBBBBB
FB: T=14.83, E=10.67, P=3.5
ABAABBCDCCEFDFEFAAABBCD
SRR: T=13.83, E=9.67, P=2.94
AABBAACCDDEEFFBBAACDDBBA
- Ronda 5:
A: 3, t=5; B: 6, t=5; C: 12, t=2; D: 12, t=5 (tot:17)
FCFS: T=5.75, E=1.5, P=1.38
---AAAAABBBBBBCCDDDD
RR1: T=6.25, E=2, P=1.55
---AAAABABBBBCDCDDDD
RR4: T=6.75, E=2.5, P=1.57
---AAAABBBBABCCDDDD
SPN: T=5.75, E=1.5, P=1.38
---AAAAABBBBBBCCDDDD
FB: T=6.75, E=2.5, P=1.65
---AAABBBAAABBCDCDDDD
SRR: T=7.5, E=3.25, P=1.8
---AAAABBAABDDCCBDDDD

Verificación manual

Se revisaron manualmente las rondas 1 y 5, y los cálculos de T, E y P coincidieron con los promedios del programa.

Los periodos de inactividad (–) y los diagramas de Gantt son correctos según cada planificador.

Procesos: A: 3, t=6; B: 9, t=7; C: 11, t=5; D: 12, t=1; E: 12, t=7. Carga total: 26 unidades de tiempo.

FCFS (First Come First Served)

Gantt: ---AAAAAABBBBBBBBCCCCDEEEEEEE

Proceso	Llegada	Ráfaga	Finalización	T	E	P
A	3	6	9	6	0	1.00
B	9	7	16	7	0	1.00
C	11	5	21	10	5	2.00
D	12	1	22	10	9	10.00
E	12	7	29	17	10	2.43
Promedio				10.00	4.80	3.29

Coincide también con la salida del programa: FCFS: T=10, E=4.8, P=3.29

RR(q=1)

Gantt: ---AAAAAABBBBCBDECBECBECBCEEEE

Proceso	Llegada	Ráfaga	Finalización	T	E	P
A	3	6	9	6	0	1.00
B	9	7	24	15	8	2.14
C	11	5	26	15	10	3.00
D	12	1	15	3	2	3.00
E	12	7	29	17	10	2.43
Promedio				11.20	6.00	2.31

Coincide también con la salida del programa: RR1: T=11.2, E=6, P=2.31

Ronda 5

Procesos: A: 1, t=4; B: 3, t=3; C: 5, t=1; D: 10, t=4. Carga total: 12 unidades de tiempo.

SPN (Shortest Process Next)

Gantt: -AAAACBBB-DDDD

Proceso	Llegada	Ráfaga	Finalización	T	E	P
A	1	4	5	4	0	1.00
B	3	3	9	6	3	2.00
C	5	1	6	1	0	1.00
D	10	4	14	4	0	1.00
Promedio				3.75	0.75	1.25

Coincide también con la salida del programa: SPN: T=3.75, E=0.75, P=1.25

Conclusión

Los algoritmos no expropiativos (FCFS, SPN) generan menos cambios de contexto, mientras que RR, FB y SRR ofrecen más equidad y mejor respuesta.

SPN logra los mejores tiempos promedio con cargas ligeras, y RR y FB equilibran bien el rendimiento en cargas mixtas.

Tras comparar las métricas promedio con las salidas del simulador, se confirma que los resultados obtenidos por el programa son consistentes, reproducibles y lógicos según la teoría.