Universidad Nacional Autónoma de México

Facultad de Ingeniería

Sistemas Operativos | Grupo 8

Tarea 2

Jiménez Olivo Evelin – 321184191

Lara Hernández Emmanuel – 321189536

Fecha: 28 de octubre de 2025

Comparación de planificadores: diseño, implementación y uso

Objetivo

Implementar un comparador de planificadores de CPU que:

- Genere **múltiples cargas aleatorias** (con **huecos** en los tiempos de llegada).
- Ejecute y compare: **FCFS/FIFO**, **RR(q)**, **SPN** (SJF no expropiativo), **FB** (multinivel con retroalimentación) y **SRR** (ronda egoísta).
- Calcule métricas promedio por algoritmo: $\overline{T}$ (turnaround), $\overline{E}$ (espera), $\overline{P}$ (penalización o slowdown).
- Presente un **Gantt textual** por algoritmo.

Para el 10: además de FCFS/RR/SPN, se implementaron **dos** algoritmos de colas múltiples: **FB** (quanta 1–2–4 con democión por agotamiento) y **SRR** (NEW con $q=1$ y OLD con $q=4$, prioridad a NEW).

Modelo de proceso y métricas

Cada proceso p_i está dado por nombre, a_i (llegada), s_i (ráfaga total). Durante la simulación se registran f_i (finalización) y r_i (remanente).

$$T_i = f_i - a_i, \qquad E_i = T_i - s_i, \qquad P_i = \frac{T_i}{s_i}.$$

Para cada algoritmo se informa el promedio $\overline{T}, \overline{E}, \overline{P}$.

Motor de simulación por ticks

La simulación avanza en pasos discretos ($\Delta t = 1$):

1. En el tick t arriban todos los procesos con $a_i = t$.
2. El planificador `select` decide qué proceso corre (o CPU ociosa).
3. El ejecutado consume 1 unidad: $r_i \leftarrow r_i - 1$. Si $r_i = 0$, se marca $f_i = t + 1$.
4. Si no hay listos, se imprime "`-`" en el Gantt (hueco).

Cada algoritmo se implementa como máquina de estados con cinco *callbacks*: `init`, `arrivals`, `select`, `tick`, `finish`. Así se reutiliza el mismo motor para todos.

Algoritmos

FCFS/FIFO Cola simple, no expropiativo. Se mantiene el proceso actual hasta terminar.

RR(q) Cola circular. Al agotar q sin terminar, reencola al final (probamos q=1 y q=4).

SPN (SJF no expropiativo) Al elegir, toma el listo con menor ráfaga total s_i. Luego no se expropia.

FB (MLFQ) Tres niveles con quanta $(1, 2, 4)$. Arriban al nivel superior; si agotan el quantum sin terminar, **descienden** un nivel. Se atiende siempre la cola de mayor prioridad no vacía.

SRR (egoísta) Dos colas: **NEW** (quantum=1) y **OLD** (quantum=4). **NEW** tiene prioridad absoluta; tras la primera rebanada, si no terminó, el proceso pasa a **OLD**.

Generación de cargas con huecos

Para cada ronda se generan n procesos con:

$$a_i \sim \mathcal{U}\{0, \ldots, t_{\max}\}, \qquad s_i \sim \mathcal{U}\{s_{\min}, \ldots, s_{\max}\}.$$

Al no haber listos en algunos instantes, aparecen huecos explícitos en el Gantt.

Uso del programa

El código está en `compara_planif.py`. Se ejecuta así:

Terminal

```
python compara_planif.py --rondas 3 --proc-min 4 --proc-max 7 --tmax 8 --seed
    123
# Para ocultar el Gantt:
python compara_planif.py --rondas 2 --no-gantt
```

Listing 1: Ejecución por CLI

Notebook (Colab/Local)

```
from compara_planif import run_demo, run_experiment
run_demo(seed=42)  # 2 rondas con Gantt
run_experiment(rondas=5, proc_min=4, proc_max=8, tmax=8, seed=2025, show_gantt=
    True)
```

Listing 2: API de alto nivel

Salida esperada y validación

Cada ronda imprime la descripción de la carga y, por algoritmo, las métricas y el Gantt textual. Ejemplo real:

```
- Ronda 1:
 D: a=0, t=6 | A: a=1, t=2 | B: a=3, t=9 | C: a=5, t=2   (tot:19)
 FCFS: T=10.2, E=5.5, P=3.26
  DDDDDDAABBBBBBBBBCC
 RR1:  T=9.5,  E=4.8, P=2.03
  DADABDCBDCBDBDBBBBB
 RR4:  T=10.5, E=5.8, P=2.69
  DDDDAABBBBDDCCBBBBB
 SPN:  T=8.5,  E=3.8, P=2.19
  DDDDDDAACCBBBBBBBBB
 FB:   T=10.0, E=5.2, P=2.36
  DADDBCABBCDDDBBBBBB
 SRR:  T=11.0, E=6.2, P=3.11
  DADDDDBCADBBBBCBBBB
```

Criterios de la rúbrica y cómo se cubrieron

1. **Múltiples ejecuciones y tendencias**: bandera `--rondas` y `--seed` para reproducibilidad.

2. **Huecos**: el motor nunca ejecuta procesos antes de su llegada; si la cola está vacía, imprime "–".

3. **Esquema visual**: Gantt textual por algoritmo en cada ronda.

4. **Cálculo de T, E, P promedios**: reportados para cada planificador.

5. **Colas múltiples**: se incluyeron **FB** y **SRR**.

Tabla guía de interpretación

Algoritmo	$\overline{T}$ bajo carga mixta	Interactividad	Justicia/Fairness
FCFS	Puede ser alta si llegan ráfagas largas	Baja	Baja
RR(q)	Depende del q; buena respuesta	Alta	Alta
SPN	Minimiza espera en promedio	Media	Media (riesgo de inanición)
FB	Favorece trabajos cortos recientes	Alta	Media-Alta
SRR	Favorece recién llegados	Muy alta	Media

Conclusiones

El comparador permite observar, con cargas aleatorias y huecos, las diferencias de política entre FCFS, RR, SPN, FB y SRR. Las métricas $\overline{T}, \overline{E}, \overline{P}$ y el Gantt textual facilitan verificar ejecuciones y discutir trade-offs: respuesta corta de RR/FB/SRR frente a riesgo de inanición en SPN. La arquitectura de callbacks hace sencillo extender y experimentar con nuevas variantes (p.ej., prioridades por envejecimiento).