

UNIVERSIDAD PRIVADA DE TACNA
FACULTAD DE INGENIERÍA
Escuela Profesional de Ingeniería de Sistemas



Práctica

“PROYECTO DE UNIDAD”

Que se presenta para el curso:
“Sistemas Operativos”

Docente:
MSc. Ing. Hugo Manuel Barraza Vizcarra

Estudiante:
Sierra Ruiz, Iker Alberto(2023077090)

TACNA – PERÚ
2025



índice

Introducción.....	3
1. Marco conceptual.....	4
2. Diseño de simulación.....	4
3. Metodología de Experimentos.....	5
4. Resultado.....	5
5. Discusión.....	5
6. Conclusión.....	6

Introducción

La planificación de procesos es un aspecto esencial en los sistemas operativos, ya que determina el orden en que los procesos acceden al procesador y, con ello, el rendimiento global del sistema. Comprender cómo distintas políticas de planificación afectan las métricas de desempeño resulta clave para analizar la eficiencia y la equidad en la ejecución de tareas.

El problema abordado en este proyecto consiste en el diseño e implementación de un simulador simple que permita visualizar y practicar los conceptos básicos de procesos, planificación de CPU y, de manera complementaria, la gestión de memoria. El objetivo principal es mostrar el impacto que tienen diferentes algoritmos sobre métricas clásicas como el tiempo de respuesta, el tiempo de espera y el tiempo de retorno.

El alcance se limita a tres algoritmos representativos: **First-Come, First-Served (FCFS)**, sencillo pero propenso al efecto convoy; **Shortest Process Next (SPN)**, que prioriza procesos más cortos y reduce la espera promedio; y **Round Robin (RR)**, orientado a la equidad gracias a un quantum configurable. Los procesos considerados poseen tiempos de llegada y servicio conocidos de antemano, lo que facilita la comparación de resultados.

Finalmente, el simulador se construye bajo supuestos que simplifican la realidad de un sistema operativo: un único procesador sin paralelismo, un costo de cambio de contexto despreciable y procesos independientes sin bloqueos. Estas condiciones permiten centrarse en el análisis de las políticas de planificación y en su efecto sobre el desempeño de los procesos.

1. Marco conceptual

Un proceso es un programa en ejecución que requiere recursos como CPU y memoria. Para administrarlo, el sistema operativo guarda su información en el PCB (Process Control Block).

Los procesos cambian de estado: pueden estar listos (esperando CPU), ejecutándose (usando la CPU) o terminados (cuando finalizan).

La planificación de procesos es la forma en que el sistema decide qué proceso usar en cada momento:

- FCFS: atiende en el orden de llegada, sin interrupciones.
- SPN/SJF: elige el proceso más corto, reduciendo esperas promedio.
- Round Robin: reparte la CPU en turnos de tiempo fijo (quantum), garantizando equidad.

En la gestión de memoria, los sistemas simples asignan bloques lineales con estrategias como:

- First-Fit: primer bloque libre adecuado.
- Best-Fit: bloque más ajustado posible para el proceso.

2. Diseño de simulación

El simulador fue desarrollado en C++ con entrada de datos interactiva.

Se utilizan vectores de estructuras (struct Proceso) para representar los procesos, con campos:

- pid (identificador)
- llegada (tiempo de arribo)
- servicio (CPU total requerido)
- inicio (primer despacho)
- fin (tiempo de finalización)

La **cola de listos** se maneja de forma distinta según el algoritmo:

- En **FCFS**, se ordena por llegada.
- En **SPN**, se selecciona dinámicamente el menor servicio pendiente.
- En **RR**, se implementa una **cola circular** en la que cada proceso recibe hasta quantum unidades de CPU antes de ser reinsertado si no termina.

El cálculo de métricas se realiza con fórmulas:

- Respuesta = Inicio – Llegada
- Retorno = Fin – Llegada
- Espera = Retorno – Servicio

3. Metodología de Experimentos

Se probaron casos de entrada con distintos números de procesos, tiempos de llegada y servicio.

Parámetros:

- Número de procesos variable (3–5).
- Algoritmo de planificación configurable.
- Quantum en RR ajustable por el usuario (≥ 2).

La repetibilidad está garantizada porque el simulador siempre da los mismos resultados para los mismos datos de entrada.

Ejemplo de caso usado (3 procesos):

- P1: llegada=0, servicio=12
 - P2: llegada=1, servicio=5
 - P3: llegada=2, servicio=8
- Algoritmo: RR con quantum=4.

4. Resultado

PID	Llegada	Servicio	Inicio	Fin	Respuesta	Espera	Retorno
1	0	12	0	25	0	13	25
2	1	5	4	17	3	11	16
3	2	8	8	21	6	11	19

Interpretación:

- El proceso 1 fue penalizado por el reparto de CPU, terminando más tarde.
- El proceso 2 obtuvo buena respuesta y espera baja gracias al quantum.
- El proceso 3 esperó más tiempo acumulado, mostrando el impacto del quantum en procesos medianos.

5. Discusión

Cada algoritmo tiene ventajas y desventajas:

- FCFS: simple, pero sufre de convoy effect (un proceso largo retrasa a todos).
- SPN/SJF: minimiza tiempo medio de espera, pero requiere conocimiento perfecto de la duración (poco realista).
- Round Robin: mejora la equidad, pero su eficiencia depende del tamaño del quantum. Un quantum demasiado grande degrada a FCFS; demasiado pequeño genera muchos cambios de contexto.

6. Conclusión

- El algoritmo de planificación seleccionado impacta directamente en el rendimiento del sistema, evidenciado en las métricas de respuesta, espera y retorno; por ejemplo, FCFS puede generar el “convoy effect”, mientras que RR mejora la equidad pero depende críticamente del quantum.
- No existe un algoritmo universalmente óptimo, ya que la eficiencia depende del tipo de carga de trabajo: SJF minimiza los tiempos de espera promedio cuando se conocen los servicios, mientras que RR es más adecuado para entornos interactivos e inciertos.
- El simulador demuestra el valor de abstraer la gestión de procesos en estructuras simples, permitiendo analizar y comparar políticas de planificación en escenarios controlados, lo que facilita la toma de decisiones sobre qué estrategia aplicar en sistemas reales.

repositorio:

<https://github.com/IkerASierraR/ProUniSO.git>