

Laboratorio 6: Algoritmos de Planificación

rypaccotacya@ucsp.edu.pe

Objetivos

- Implementar diferentes algoritmos de planificación de CPU.
- Analizar su rendimiento en términos de tiempo de respuesta promedio (average response time) y tiempo de retorno promedio (average turnaround time).
- Comprender el impacto del time quantum en el algoritmo Round Robin.

Instrucciones

Parte I

Deberás escribir un programa en C que implemente cuatro algoritmos de planificación de CPU:

- First-Come, First-Served (FCFS/FIFO)
- Shortest Job First (SJF)
- Shortest-Time-to-Completion-First STCF
- Round Robin (RR) – Con quantum de tiempo definido por el usuario

Su programa debe poder aceptar dos parámetros de línea de comandos. El primer argumento especifica el nombre del archivo que contiene la información de los procesos. Este archivo está separado por comas y tiene tres columnas, donde cada fila representa un proceso. Se espera que este archivo contenga un máximo de 12 procesos. El segundo argumento es el quantum de tiempo (time slice) para la programación Round Robin.

Entrada de datos

La información de cada proceso debe ingresar a partir de un archivo .csv que contiene:

```
ProcessID, Arrival Time, Burst Time
3,0,6
2,0,5
...
```

El archivo de entrada si tiene la cabecera y además los procesos van a estar ordenados por el tiempo de llegada. En caso de empate, usar el orden lexicográfico del ID del proceso.

Salida Esperada

Para cada algoritmo, el programa debe mostrar:

1. Diagrama de Gantt con el orden de ejecución.
2. Una tabla con, al menos:
 - Tiempo de respuesta de cada proceso.
 - Tiempo de retorno de cada proceso.
3. Promedios de:
 - Tiempo de respuesta.
 - Tiempo de retorno.
4. Número de cambios de contexto

Nota: Los procesos en el archivo (primer argumento de la línea de comandos) pueden especificarse de tal manera que generen tiempo de inactividad cuando no haya procesos listos para ejecutarse. Durante este tiempo, la CPU no tiene procesos que ejecutar y espera a que el siguiente aparezca en la cola de procesos listos.

El diagrama de Gantt debe incluir estos tiempos de inactividad como IDLE. Además, no es necesario que el primer proceso comience en el tiempo 0. En el tiempo 0, puede haber tiempo de inactividad en el diagrama de Gantt hasta que el primer proceso llegue posteriormente.

Supongamos que, cuando dos procesos nuevos llegan simultáneamente, el proceso con el ID más bajo entra primero en la cola de procesos listos.

Ejemplo parcial de formato del Output

Nota: Waiting time es diferente que Response time, el waiting time es igual a Turnaround time - Burst time, pueden incluir de manera opcional esta otra métrica.

Round Robin		
Process ID	Waiting Time	Turnaround Time
1	6	14
2	3	8
3	5	11
4	5	11

Round Robin Gantt Chart

```
[0]--2--[3]
[3]--3--[6]
[6]--2--[8]
[8]--3--[11]
[11]--IDLE--[12]
[12]--1--[15]
[15]--4--[18]
[18]--1--[21]
[21]--4--[24]
[24]--1--[26]
```

Average Waiting Time: 4.75
Average Turnaround Time: 11.0

Parte II

Realiza experimentos para comprender cómo el valor del time quantum (q) afecta el rendimiento del algoritmo Round Robin (RR).

Mide y compara métricas relevantes (tiempo de respuesta promedio, tiempo de retorno promedio y número de cambios de contexto).

Presenta tus hallazgos en un informe escrito. Puedes apoyar tus resultados con tablas y gráficos.

Responde:

1. ¿Cómo afecta q al tiempo de respuesta y retorno promedio? ¿Por qué?
2. ¿Qué efecto tiene q en el número de cambios de contexto?
3. ¿Existe un q óptimo en tu experimento? ¿Por qué ese valor parece “bueno”?
4. ¿Qué sucede cuando $q \rightarrow 1$ (muy pequeño) y cuando $q \rightarrow \infty$ (muy grande)?
5. ¿Qué compromisos prácticos habría que considerar al elegir q en un sistema real?

Entregables

- Código y .csv comprimido en un archivo .zip
- Informe conteniendo screenshots de la ejecución de Parte I y sus hallazgos de la Parte II
- En laboratorio, (día 30-09-25 Grupo 6-2 / día 01-10-25 Grupo 6-1) se les compartirá un archivo .csv
- Evaluación oral
- Grupos de 2 personas, solo un integrante realiza la entrega