



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA



SISTEMAS OPERATIVOS

TAREA 2

ALUMNOS:

PALÍ FIGUEROA SANTIAGO

HERNÁNDEZ SALDÍVAR HÉCTOR SAÚL

PROFESOR:

GUNNAR EYAL WOLF ISZAEVICH

GRUPO 6

SEMESTRE 2025 - 1

FECHA DE ENTREGA 19/11/2024

¿Qué necesito para poder ejecutar el programa?

Para ejecutar el simulador de planificadores de procesos, necesitarás cumplir con los siguientes requisitos:

- Lenguaje de programación: El programa está escrito en Python, por lo que es necesario tenerlo instalado en tu sistema.
- Dependencias: Debes instalar las dependencias requeridas para el programa. En este caso, solo necesitas instalar el paquete colorama. Esto se puede hacer fácilmente ejecutando el siguiente comando en la terminal:

pip install colorama

- Sistema operativo: El programa debería funcionar en cualquier sistema operativo que soporte Python, incluyendo Windows, macOS y Linux.

Explicación de cada función del código

Generación de procesos:

- La función encargada de generar los procesos aleatorios toma como parámetros la cantidad de procesos, la probabilidad de generación y las duraciones mínimas y máximas. Esta función crea una lista de procesos, asignándoles una duración aleatoria dentro del rango especificado.

Algoritmos de planificación:

- FCFS (First Come First Serve): Los procesos se ejecutan en el orden en que llegan, sin interrupciones. Este algoritmo es el más sencillo y se utiliza para evaluar el rendimiento de los demás algoritmos.
- Round Robin: Los procesos se ejecutan de manera circular, utilizando un quantum (Q) que define cuánto tiempo puede ejecutarse un proceso antes de que se cambie al siguiente. Hay dos versiones: una con Q=1 y otra con Q=4.
- Shortest Process Next (SPN): Este algoritmo selecciona el proceso con el tiempo de ejecución más corto disponible para ejecutarlo a continuación.
- Selfish Round Robin: Es una variante del Round Robin, con parámetros A y B que definen cómo se asignan los tiempos de CPU entre los procesos en función de su duración.

Cálculos de métricas:

- Para cada algoritmo, el programa calcula el tiempo de ejecución (T), el tiempo de espera (E) y el rendimiento (P) de los procesos. Estas métricas se utilizan para comparar la eficiencia de cada algoritmo en términos de su rendimiento.

Generación de esquemas:

- El programa genera un esquema visual de la ejecución de los procesos, que muestra la alternancia de los procesos en función del algoritmo utilizado. Los esquemas ayudan a entender cómo se distribuye el tiempo de CPU entre los procesos.

Ejecuciones

Ejecución 1:

```
Simulador de Planificadores de Procesos
Número de procesos a generar: 5
Probabilidad de generación (1 en P): 3
Duración mínima de un proceso: 2
Duración máxima de un proceso: 10

--- Algoritmo FCFS ---

FCFS

Proceso Duración      Inicio  Fin    T     E     P
D       9             0      9     9     0     1.00
E       8             9     17     8     9     1.00
Promedio:              T=8.50 E=4.50 P=1.00

--- Round Robin (Q=1) ---

Round Robin (Q=1)

Proceso Duración      Inicio  Fin    T     E     P
E       8             1     16    15     1     1.88
D       2             2     17    15     2     1.67
Promedio:              T=15.00 E=1.50 P=1.77
ESQUEMA:
DEDEDEDEDEDEDEDEDE

--- Round Robin (Q=4) ---

Round Robin (Q=4)

Proceso Duración      Inicio  Fin    T     E     P
E       8             4     16    12     4     1.50
D       9             8     17     9     8     1.00
Promedio:              T=10.50 E=6.00 P=1.25
ESQUEMA:
DDDEEEEEDDDEEEED

--- Shortest Process Next ---

Shortest Process Next

Proceso Duración      Inicio  Fin    T     E     P
E       8             0      8     8     0     1.00
D       9             8     17     9     8     1.00
Promedio:              T=8.50 E=4.00 P=1.00

--- Selfish Round Robin (A=2, B=1) ---

Selfish Round Robin (A=2, B=1)

Proceso Duración      Inicio  Fin    T     E     P
E       8             1     16    15     1     1.88
D       2             2     17    15     2     1.67
Promedio:              T=15.00 E=1.50 P=1.77
ESQUEMA:
DEDEDEDEDEDEDEDEDE
```

FCFS: El algoritmo ejecuta los procesos en el orden de llegada. Los tiempos de espera son bajos, pero el tiempo total de ejecución es relativamente alto.

Round Robin (Q=1): El tiempo total de ejecución es alto debido al pequeño quantum, lo que provoca más conmutaciones entre procesos. El rendimiento y el tiempo de espera son relativamente elevados.

Round Robin (Q=4): Con un quantum mayor, el tiempo total de ejecución es más bajo, pero el tiempo de espera aumenta.

Shortest Process Next (SPN): Similar a FCFS, ya que las duraciones de los procesos son similares, lo que genera tiempos de ejecución y espera equilibrados.

Selfish Round Robin (A=2, B=1): El comportamiento es similar al Round Robin con Q=1, pero con una alternancia de prioridades que genera un mayor tiempo total de ejecución.

Ejecución 2:

```
Simulador de Planificadores de Procesos
Número de procesos a generar: 6
Probabilidad de generación (1 en P): 4
Duración mínima de un proceso: 3
Duración máxima de un proceso: 12

--- Algoritmo FCFS ---

FCFS

Proceso Duración Inicio Fin T E P
A 12 0 12 12 0 1.00
Promedio: T=12.00 E=0.00 P=1.00

--- Round Robin (Q=1) ---

Round Robin (Q=1)

Proceso Duración Inicio Fin T E P
A 12 1 12 11 1 0.92
Promedio: T=11.00 E=1.00 P=0.92
ESQUEMA:
AAAAAAAAAAAA

--- Round Robin (Q=4) ---

Round Robin (Q=4)

Proceso Duración Inicio Fin T E P
A 12 4 12 8 4 0.67
Promedio: T=8.00 E=4.00 P=0.67
ESQUEMA:
AAAAAAAAAAAA

--- Shortest Process Next ---

Shortest Process Next

Proceso Duración Inicio Fin T E P
A 12 0 12 12 0 1.00
Promedio: T=12.00 E=0.00 P=1.00

--- Selfish Round Robin (A=2, B=1) ---

Selfish Round Robin (A=2, B=1)

Proceso Duración Inicio Fin T E P
A 12 2 12 10 2 0.83
Promedio: T=10.00 E=2.00 P=0.83
ESQUEMA:
AAAAAAAAAAAA
```

FCFS: Un solo proceso, sin tiempo de espera, con el rendimiento óptimo.

Round Robin (Q=1): La ejecución es más lenta debido al pequeño quantum, pero el rendimiento es decente.

Round Robin (Q=4): El tiempo total de ejecución es más eficiente, pero el tiempo de espera es más alto.

Shortest Process Next (SPN): Similar a FCFS, con tiempos de ejecución y espera eficientes.

Selfish Round Robin (A=2, B=1): Los tiempos de ejecución y espera son similares a Round Robin con un rendimiento ligeramente inferior.

Ejecución 3:

```
Simulador de Planificadores de Procesos
Número de procesos a generar: 4
Probabilidad de generación (1 en P): 2
Duración mínima de un proceso: 1
Duración máxima de un proceso: 8

--- Algoritmo FCFS ---

FCFS

Proceso Duración Inicio Fin T E P
A 5 0 5 5 0 1.00
D 2 5 7 2 5 1.00
Promedio: T=3.50 E=2.50 P=1.00

--- Round Robin (Q=1) ---

Round Robin (Q=1)

Proceso Duración Inicio Fin T E P
D 2 1 4 3 1 1.50
A 5 2 7 5 2 1.00
Promedio: T=4.00 E=1.50 P=1.25
ESQUEMA:
ADADAAA

--- Round Robin (Q=4) ---

Round Robin (Q=4)

Proceso Duración Inicio Fin T E P
D 2 4 6 2 4 1.00
A 5 6 7 1 6 0.20
Promedio: T=1.50 E=5.00 P=0.60
ESQUEMA:
AAAADDA

--- Shortest Process Next ---

Shortest Process Next

Proceso Duración Inicio Fin T E P
D 2 0 2 2 0 1.00
A 5 2 7 5 2 1.00
Promedio: T=3.50 E=1.00 P=1.00

--- Selfish Round Robin (A=2, B=1) ---

Selfish Round Robin (A=2, B=1)

Proceso Duración Inicio Fin T E P
D 2 2 6 4 2 2.00
A 5 3 7 4 3 0.80
Promedio: T=4.00 E=2.50 P=1.40
ESQUEMA:
AADAADA
```

FCFS: Los tiempos de ejecución y espera son equilibrados.

Round Robin (Q=1): El algoritmo es más eficiente en términos de tiempo de ejecución total, pero aumenta el tiempo de espera.

Round Robin (Q=4): El mayor quantum mejora el tiempo total de ejecución, pero aumenta el tiempo de espera en algunos procesos.

Shortest Process Next (SPN): Este algoritmo muestra un buen rendimiento al seleccionar el proceso más corto primero.

Selfish Round Robin (A=2, B=1): Similar al Round Robin, pero con una ligera ventaja en los tiempos de ejecución debido a la política de prioridades.