

# Tarea 4: Simulación de eventos discretos

Mario Medina

mariomedina@udec.cl

Programación orientada al objeto — 24 de septiembre de 2018

## Introducción

En esta cuarta tarea, Uds. deben experimentar con las clases `list`, `queue`, `priority queue` y/o `pair` de la biblioteca estándar de C++. Para ello, se le pide escribir un simulador de eventos discretos.



**Info:** De más está decirle que esta tarea es individual: puede comentar posibles métodos de solución con sus compañeros, pero se espera que los códigos entregados por todos los alumnos sean diferentes.

Envíeme su código fuente junto con un informe de a lo más 3 planas, detallando su método de solución y los resultados de las cuatro simulaciones el día viernes 5 de octubre, antes de medianoche a [mariomedina@udec.cl](mailto:mariomedina@udec.cl).

## Simulación de eventos discretos

En InfoAlumno Ud. encontrará un archivo llamado `eventos.txt`, que contiene una lista de 1000 eventos. El formato del archivo `eventos.txt` especifica en su primera columna el tiempo de llegada de un evento, en segundos, en su segunda columna la duración del evento, también en segundos y en la tercera columna la prioridad del evento, en un rango de 0 a 9.

Cada evento en el archivo representa un proceso que llega a una cola de atención que alimenta a una CPU. La CPU extrae un proceso de la cola y lo ejecuta por el tiempo especificado hasta que el proceso termina. Luego, la CPU extrae otro proceso de la cola y lo ejecuta. Si no hay procesos en la cola, la CPU espera ociosa la llegada de un nuevo proceso. Este ciclo de ejecución continúa hasta haber ejecutado todos los eventos del archivo.

Este esquema de ejecución se muestra en la figura 1.

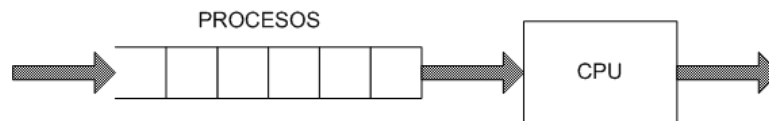


Figura 1: Cola de atención de procesos

## Definiciones

En este contexto, definiremos los siguientes parámetros:

**Tiempo de servicio ( $T_s$ )** Corresponde a la duración del evento.

**Tiempo de espera ( $T_w$ )** Corresponde al período entre el tiempo de llegada del evento y el tiempo de selección del evento por la CPU.

**Tiempo de respuesta ( $T_r$ )** Corresponde al período entre la llegada del evento y que la CPU termina su ejecución. Es igual a  $T_w + T_s$ .

**Tasa de respuesta ( $R$ )** Corresponde a la razón entre el tiempo de respuesta y el tiempo de servicio

$$R = \frac{T_w + T_s}{T_s}.$$

**Utilización ( $\rho$ )** Porcentaje del tiempo de simulación que la CPU está ocupada.

## Políticas de selección de eventos

Uno de los tópicos de estudio en la implementación de una cola de atención de procesos es cómo elegir el siguiente evento a ejecutar. En este proyecto, se le pide implementar cuatro políticas de selección de eventos no apropiativas:

**FCFS** La política *First-Come, First-Served* escoge el evento a ejecutar en base al orden de llegada: siempre atiende primero al evento que lleva más tiempo esperando.

**SJF** La política *Shortest Job First* escoge el evento de menor duración. En el caso que dos eventos tengan la menor duración, se aplica FCFS entre ellos.

**HRRN** La política *Highest Response Ratio Next* escoge aquel evento cuya tasa de respuesta  $R$  es la mayor. En caso que dos eventos tengan la misma tasa, se aplica FCFS entre ellos.

**Prioridades** La política por *Prioridades* escoge aquel evento de la cola de espera que tiene la mayor prioridad, donde 0 es la prioridad más alta y 9 es la prioridad más baja. En caso que dos eventos tengan la misma prioridad, se aplica FCFS entre ellos.

## Ejemplo de políticas de selección de eventos

Suponga que su simulador recibe los siguientes 4 eventos:

Evento	Tiempo de llegada	Duración	Prioridad
1	0	5	7
2	3	5	2
3	4	2	4
4	5	1	8

Su simulación debe proceder como sigue:

- Al tiempo  $t = 0$ , se recibe el evento 1 y comienza su ejecución por un período de 5 s.
- Al tiempo  $t = 3$ , se recibe el evento 2. Sin embargo, como la CPU está ocupada atendiendo al evento 1, el evento 2 se ve postergado.

- Al tiempo  $t = 4$ , se recibe el evento 3. Como la CPU está todavía ocupada atendiendo al evento 1, se posterga el evento 3.
- Al tiempo  $t = 5$ , el evento 1 termina su ejecución, liberando a la CPU. Al mismo tiempo, se recibe el evento 4.

Entonces, para  $t = 5$  hay tres eventos esperando por la CPU. El evento a ejecutar ahora dependerá de la política de selección de eventos. En caso de usarse una política FCFS (*First-Come, First-Served*), se elige al evento 2 por ser el primero en llegar al sistema. Si se aplica una política SJF (*Shortest Job First*), se elige al evento 4 por ser el de menor tiempo de servicio. Una política HRRN (*Highest Response Ratio Next*) escoge al evento 3 por tener éste una razón  $R = 1.5$ . Finalmente, una política por prioridades escoge el evento 2 por ser el de prioridad más alta entre los tres eventos.

## Requerimientos del problema

En este problema, se le pide escribir un programa en C++ llamado `simulador.exe` que reciba como argumentos de línea de comando el archivo a procesar y la política de selección, que puede ser FCFS, SJF, HRRN ó PRIO. Su programa debe simular la ejecución de los eventos y al término de la simulación, desplegar en pantalla los siguientes parámetros:

- Duración total de la simulación
- $T_w$  mínimo, máximo y promedio
- $T_r$  mínimo, máximo y promedio
- Largo mínimo, máximo y promedio de la cola de espera
- Utilización de la CPU