

Tarea 2

Programación orientada al objeto

1er. Semestre 2017

Introducción

Esta tarea está diseñada para que Ud. experimente 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.

Simulación de eventos discretos

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

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.

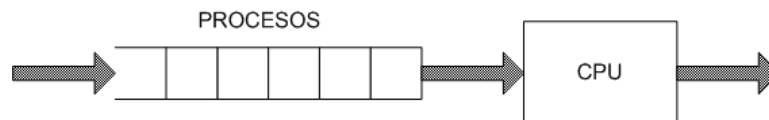


Figura 1: Cola de atención de procesos

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 (ρ) 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 tres políticas de selección de eventos:

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.

Ejemplo de políticas de selección de eventos

Suponga que su simulador recibe los siguientes 4 eventos:

Evento	Tiempo de llegada	Duración
1	0	5
2	3	5
3	4	2
4	5	1

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. Finalmente, una política HRRN (*Highest Response Ratio Next*) escoge al evento 3 por tener éste una razón $R = 1.5$.

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 ó HRRN. 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

Su proyecto será corregido en base a la simulación correcta de los eventos en el archivo y además en base a la calidad y eficiencia de su código. Utilice donde sea posible las bibliotecas y algoritmos de la STL, usando iteradores. *Sugerencia:* implemente su cola de atención de procesos como un contenedor que se mantiene ordenado según diferentes criterios correspondientes a las políticas de selección de procesos.

Esta tarea puede ser realizada en grupos de máximo 3 personas. Su código fuente deberá ser subido a InfoAlumno antes de las 6 pm del día viernes 12 de mayo. Ud. deberá entregar un informe escrito en secretaría de Electrónica antes de esa fecha, que debe incluir un listado de su programa, además de una descripción de las dificultades encontradas y cómo fueron solucionadas. La calificación de la tarea se basará enteramente en la información incluida en el informe.