

Tarea Programada III: Algoritmos de Planificación de Procesos

Objetivos

- Poner en práctica el funcionamiento de los algoritmos de planificación de procesos vistos en clase en el capítulo 5 del libro de texto, además del cálculo de métricas de relacionadas.

Instrucciones

Para esta tarea, debe trabajar en un programa que compare algunos de los algoritmos de planificación de procesos en términos de los criterios vistos en clase. Para esto, su programa debe tener la capacidad de recibir por parámetro lo siguiente:

- Al menos dos procesos que estén listos para hacer uso del procesador
- El “burst time” de cada proceso
- La prioridad del proceso (si aplica)
- El tiempo de llegada (si aplica)
- El algoritmo a analizar

Una vez su programa reciba estos datos, debe tomarlos, aplicar el algoritmo correspondiente y analizar el rendimiento basado en los criterios vistos en clase:

- Throughput
- Turnaround time
- Waiting time

Además, tome en cuenta que para esta tarea programada, los algoritmos que debe tomar en cuenta son:

- First Come First Served
- Shortest Job First
- Round Robin ($Q = 2$)
- Priority

Entonces, dada toda esta información, su programa debe recibir una entrada similar a (en el orden indicado en la lista indicada en las instrucciones):

Entrada de ejemplo: 5 / 10, 1, 2, 1, 3 / 3, 1, 4 5, 2 / 0 / Priority

En la entrada anterior, el primer parámetro indica la cantidad de procesos, el segundo parámetro indica el “burst time” por proceso, el tercer parámetro indica la prioridad por proceso, el cuarto proceso indica el tiempo de llegada, y por último se tiene el nombre del algoritmo a utilizar. Tome en cuenta que si para la prioridad o el tiempo de llegada tienen un 0, se toma como que todos tienen el mismo tiempo de llegada o la misma prioridad.

Como respuesta, luego del procesamiento de la cadena, su programa debe mostrar en consola una salida similar a:

Throughput por proceso y promedio
Turnaround time por proceso y promedio
Waiting time por proceso y promedio

Puntos extra (25 puntos)

- De manera adicional, puede integrar en su programa una funcionalidad que permita, cambiando la última entrada por la palabra “all” de manera que se realicen los cálculos para todos los algoritmos y compare el rendimiento, a manera individual y promedio (10 puntos).
- Puede implementar el funcionamiento para el algoritmo “Multilevel Feedback Queue”. En este caso debe trabajar con tres niveles, donde cada uno implemente un algoritmo distinto. Usted puede elegir los algoritmos a implementar en la cola multinivel. Debe definir la forma en que se sube o baja de nivel en las colas y explicarlo bien en la documentación. Sus casos de prueba deben ser lo suficientemente completos como para que se pueda ver el cambio de prioridad. Además, debe incluir impresiones de pantalla indicando como se están procesando y cómo cambian de prioridad (15 puntos).

Notas adicionales

- Debe correr en el sistema operativo: Ubuntu o Fedora.
- Trabajo de tipo individual o en parejas.
- Debe trabajar en C o C++
- Asegúrese de que su entrega se compile y se ejecute múltiples veces. **Tenga en cuenta que no se revisarán proyectos que no compilen o funcionen adecuadamente.**

Consideraciones del entregable:

- **Buenas prácticas de código:** Cada función, clase o estructura debe tener comentarios claros que expliquen su propósito, modularidad de funcionalidades y nombres descriptivos.
- **Testeo y resultados:** incluir pruebas de su sistema, mostrando cómo se utiliza el programa y los resultados obtenidos.
- **Imágenes de resultados:** se deben de agregar imágenes o screenshots de las salidas ideales o esperables al probar su proyecto
- **README:** se debe de incluir un README que explique cómo compilar, ejecutar el programa y corroborar los test previamente realizados, además de dar una visión general del funcionamiento del programa. Además, debe incluir el carnet de cada uno de los integrantes de la pareja. Éste documento será su documentación externa, asegúrese de entregarlo completo y claro.
- **Formato de entrega:** Toda entrega se debe de hacer en mediación, con un formato de archivo **carnet_TPI.zip** o **carnet&carnet_TPI.zip** que sea la versión final y que incluya lo suficiente para que su código pueda ser correctamente ejecutado.

Entregables:

- Implementación (código y README): 14 de Junio, 11:59 PM