

LABORATORIO 05

COLAS DE PRIORIDAD

Docente: Franci Suni Lopez

21 de octubre de 2020

1 COMPETENCIA DEL CURSO

Conoce, comprende e implementa estructuras de datos generales, sus aplicaciones y usos.

2 COMPETENCIA DE LA PRÁCTICA

Implementa una cola de prioridad y sus respectivas operaciones.

3 CONCEPTOS BÁSICOS

3.1 Cola de Prioridad

Una cola de prioridad es un tipo de datos abstracto que almacena un conjunto de datos que poseen una llave perteneciente a algún conjunto ordenado, y permite insertar nuevos elementos y extraer el máximo (o el mínimo, en caso de que la estructura se organice con un criterio de orden inverso).

Es frecuente interpretar los valores de las llaves como prioridades, con lo cual la estructura permite insertar elementos de prioridad cualquiera, y extraer el de mejor prioridad. Dos formas simples de implementar colas de prioridad son:

- Una lista ordenada:
 - Inserción: $O(n)$
 - Extracción de máximo: $O(1)$
- Una lista desordenada:
 - Inserción: $O(1)$
 - Extracción de máximo: $O(n)$

3.2 Planificación de procesos

Un proceso es un programa en ejecución. Existen 3 estados en los que puede encontrarse un proceso, estos son: "Listo", "Bloqueado" y "En ejecución". Para el control de los mismos internamente son almacenados en una lista, cada uno de los nodos guarda información de un proceso. En esa información se almacena, entre otros aspectos, el estado en que se encuentra el proceso, el tiempo que el proceso ha usado el CPU, e información de E/S (entrada/salida). Los sistemas operativos cuentan con un componente llamado planificador, que se encarga de decidir cuál de los procesos hará uso del procesador. La toma de esta decisión, así como el tiempo de ejecución del proceso, estará dada por un algoritmo, denominado Algoritmo de Planificación¹. Alguno de estos algoritmos son los siguientes:

¹ Texto extraído de https://www.ecured.cu/Planificación_de_procesos_en_un_sistema_operativo

- El algoritmo de planificación de procesos Round-Robin es uno de los más sencillos y antiguos. A cada proceso se le asigna un intervalo de tiempo en la CPU, llamado quantum, durante el cual se le permite ejecutarse. Si el proceso todavía se está ejecutando al expirar su quantum, el sistema operativo se apropia de la CPU y se la da a otro proceso. Si el proceso se bloquea o termina antes de expirar el quantum, se hace la conmutación de proceso. Para ello, el planificador mantiene una lista de procesos ejecutables. Cuando un proceso gasta su quantum, se le coloca al final de la lista.
- Prioridad al más corto. Otro algoritmo de planificación de procesos es SJF (Shortest Job First). El proceso que se encuentra en ejecución cambiará de estado voluntariamente, o sea, no tendrá un tiempo de ejecución determinado para el proceso. A cada proceso se le asigna el tiempo que usará cuando vuelva a estar en ejecución, y se irá ejecutando el que tenga un menor tiempo asignado. Si se da el caso de que dos procesos tengan igual valor en ese aspecto emplea el algoritmo FCFS¹.
- Primero en llegar primero en ser servido. Conocido como FCFS (First Come First Served). Este algoritmo emplea una cola de procesos, asignando un lugar a cada proceso por el orden de llegada. Cuando el proceso llega es puesto en su lugar en la cola después del que llegó antes que él y se pone en estado de listo. Cuando un proceso comienza a ejecutarse no se interrumpe su ejecución hasta que termina de hacerlo¹.

4 EJERCICIOS

- Implementar su propia cola de prioridad (usando una lista enlazada) y usarlo en un programa que permita planificar procesos en base al tiempo de consumo de CPU.
- Utilizando una cola de prioridad (implementar su propia cola), simular la planificación de procesos Round-Robin usando la lógica del algoritmo SJF.
- El sistema manejará un Quantum como el algoritmo original Round-Robin. Sin embargo, los procesos ya no ingresarán por el orden de llegada, sino ingresará el proceso que tenga el menor tiempo de consumo de CPU.
- Cuando el proceso haya gastado su quantum este debe de ser insertado de nuevo en la cola de prioridad y deberá de ser llamado en el orden de su tiempo restante de consumo de CPU.
- Construir un archivo (procesos.txt) con la información de los procesos según su orden de llegada. El formato debe ser: identificador de proceso, tiempo de CPU requerido en ms, por ejemplo:
 - P1, 21
 - P2, 4
 - P3, 40
- Construir un programa que simule la operación de la ejecución de procesos a partir de la información del archivo procesos.txt y un valor de quantum determinado. Un ejemplo de la salida de la simulación para un quantum de 10ms, con los tres procesos descritos anteriormente sería:
 - Tiempo 0: P2 entra a ejecución.
 - Tiempo 4: P2 se conmuta. Pendiente por ejecutar 0 ms.
 - Tiempo 4: P1 entra a ejecución
 - Tiempo 14: P1 se conmuta. Pendiente por ejecutar 11 ms
 - Tiempo 14: P1 entra a ejecución
 - Tiempo 24: P1 se conmuta. Pendiente por ejecutar 1 ms
 - Tiempo 24: P1 entra a ejecución
 - Tiempo 25: P1 se conmuta. Pendiente por ejecutar 0 ms

- Tiempo 25: P3 entra a ejecución
- Tiempo 35: P3 se conmuta. Pendiente por ejecutar 30 ms.
- .
- .
- .
- Tiempo 65: P1 termina su ejecución
- ESTADÍSTICAS GENERADAS:
- Total tiempo de ejecución de todos los procesos: 65 ms

5 EQUIPOS Y MATERIALES

- Un computador.
- Bibliografía del curso [1].
- Material del curso.

6 ENTREGABLES

Al finalizar el estudiante deberá:

1. En un zip colocar su código fuente.
2. Subir su archivo al classroom (teniendo hasta el domingo 25/10 hasta las 10:00 pm) con el nombre:
Laboratorio_05_ApellidoPaterno_ApellidoMaterno_PrimerNombre_UNSA_EPCC_AED.

7 RÚBRICA DE EVALUACIÓN

Criterios	Muy Bueno	Bueno	Regular	Malo
Resolución del Laboratorio	Resuelve todos los ejercicios sin errores mostrando cada uno de los puntos solicitados. Puntaje: 16 puntos	Resuelve todos los ejercicios con pocos errores mostrando casi o todos los puntos solicitados. Puntaje: 14 puntos	Resuelve todos los ejercicios con varios errores y mostrando todos o pocos de los puntos solicitados. Puntaje: 8 puntos	No resuelve todos los ejercicios o no entrega el laboratorio. Puntaje: 0 puntos
Presentación y Resolución de Preguntas	La presentación es clara y entendible, sin errores y respondiendo todas las preguntas. Puntaje: 4 puntos	La presentación es clara y entendible, con algunos errores; y respondiendo la mayor cantidad de preguntas. Puntaje: 2 puntos	La presentación no es entendible y/o comete muchos errores. Puntaje: 1 punto	No presenta todos los ejercicios o no entrega el laboratorio. Puntaje: 0 puntos

- **IMPORTANTE** En caso de copia o plagio o similares todos los alumnos implicados tendrán sanción en toda la evaluación del curso.

BIBLIOGRAFÍA

- [1] THOMAS H. CORMEN, C. E. L. R. R. C. S. **Introduction to Algorithms**. third edition edition. USA: MIT Press, 2009. ISBN 0072958863.