Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-2692 - Laboratorio de Algoritmos y Estructuras de Datos II
Trimestre Enero-Marzo 2020

TAD Cola de Prioridad

1. Introducción

Se desea que realice la implementación concreta del TAD Cola de Prioridad. También se tiene como objetivo implementar un planificador de procesos que use el TAD Cola de Prioridad

2. Actividades a realizar

2.1. Implementación del TAD Cola de Prioridad

Debe implementar el TAD Cola de Prioridad, usando el pseudo código y las operaciones indicadas en [1] con algunas diferencias. La primera diferencia consiste en que en lugar de un max_heap, se debe hacer uso de un min_heap. Es decir, el elemento con mayor prioridad va a ser el elemento con la clave más pequeña. La segunda modificación es que la Cola de Prioridad en lugar de contener solo claves enteras, va contener objetos de cualquier tipo T, tal que se tiene como precondición que los objetos de tipo T deben tener el atributo *clave*. El TAD Cola de Prioridad debe ser implementado como una clase llamada `ColaPrioridad`, en un archivo llamado `cola_prioridad.py`. La tercera modificación consiste en que dos objetos pueden tener la misma clave. En ese caso, el objeto de mayor prioridad es aquel que tiene la menor posición en el arreglo de entrada, en el momento de crear la Cola de Prioridad.

2.2. Planificador de Procesos

El objetivo es el de hacer uso de la Cola de Prioridad de la sección anterior, para implementar un Planificador de Procesos. El Planificador de Procesos le indica al CPU del computador, cual es el siguiente proceso a ejecutarse. Existen varios algoritmos para la planificación de procesos. En este laboratorio se quiere simular el algoritmo *Priority Scheduling*. Este algoritmo recibe como entrada los procesos a ejecutarse, para luego indicar al CPU que ejecute el proceso que tenga menor prioridad. Si dos procesos tienen la misma prioridad, entonces se ejecuta el proceso que haya llegado primero para ejecutarse. Una vez que un proceso es asignado a un CPU, el proceso se ejecuta completamente, es decir, se ejecuta por todo el tiempo solicitado. Para este laboratorio, cada proceso va a tener tres atributos, el identificador, el tiempo de CPU a utilizar (*Burst time*) y la prioridad. Tanto el *Burst time*, como la prioridad van a ser expresadas como números enteros. La Tabla 1 muestra un ejemplo de una serie de procesos a ser ejecutados, que recibe como entrada el planificador de procesos. La Figura 1 muestra un diagrama de GANTT con la planificación de como se ejecutarán en el CPU los procesos de la Tabla 1, usando el algoritmo *Priority Scheduling*.

Para representar los procesos debe crear una clase `Proceso`, contenida en archivo llamado `pcpu.py`. El algoritmo *Priority Scheduling* va debe ser implementado en el archivo *priority_scheduling.py*. Esta aplicación recibe como entrada un archivo con la información

Identificador	Prioridad	*Burst time* (seg.)
P3	4	6
P1	2	21
P2	1	3
P4	3	2

Tabla 1: Tabla con procesos a ser ejecutados

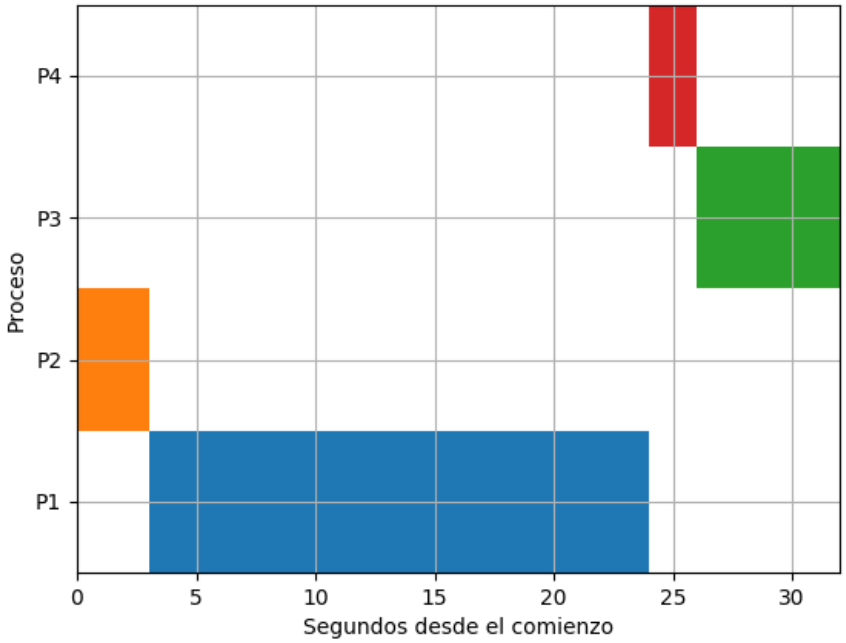

Figura 1: Diagrama de GANTT que muestra como se ejecutan los procesos de la Tabla 1 en el CPU

de los procesos a ser ejecutados en el CPU. El formato de archivo es el siguiente: cada línea contiene la información de proceso, separada por coma. El primer lugar se presenta el identificador, luego la prioridad y finalmente el *Burst time*. Los pasos a ejecutar por la implementación el algoritmo *Priority Scheduling* a realizar son como sigue.

1. El algoritmo recibe como entrada un archivo con los procesos a ser ejecutados.
2. Se crea un Cola de Prioridad con objetos de tipo Proceso.
3. Se usa la Cola de Prioridad para realizar un Diagrama de GANTT con la planificación de ejecución de los procesos.
4. Se crea un gráfico con el Diagrama de GANTT usando la librería Matplotlib, tal como en la Figura 1.
5. Se crea un gráfico del min_heap de la Cola de Prioridad, mostrando los identificadores de los procesos, tal como se muestra en la Figura 2. Para la creación de este gráfico debe hacer uso del API de Python del software Graphviz [1]

El planificador de procesos debe ser ejecutado con la siguiente línea de comando:

```
>./priority\_scheduling.py <Archivo de entrada>
```

[1] `https://graphviz.org/`

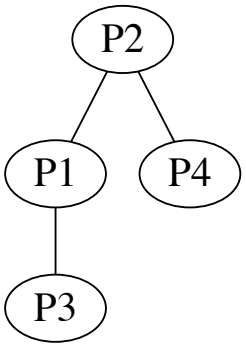

Figura 2: Min_heap de los procesos de la Cola de Prioridad de la Tabla 1

3. Condiciones de entrega

La versión final del código del laboratorio debe estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *LabSem8_X.tar.xz*, donde X es el número de carné del estudiante. La entrega del archivo *LabSem8_X.tar.xz*, debe hacerse al profesor del laboratorio por email, antes de las 9:00 pm del día domingo 01 de marzo de 2020.

Referencias

[1] Cormen, T., Leiserson, C., Rivest, R., and Stein, C. *Introduction to algorithms*, 3rd ed. MIT press, 2009.

Guillermo Palma / gvpalma@usb.ve / Febrero de 2020