

## Práctica 2 - Simulación de un Planificador de Procesos en un Sistema Operativo

### Parte 3 - Árboles de Búsqueda

En esta práctica se pretende completar la práctica 1 (parte 2), simulación del funcionamiento de un planificador de procesos de un sistema operativo, obteniendo los siguientes datos sobre los procesos que han sido ejecutados:

- **Procesos asignados a un nivel de prioridad específico:** Listado de los procesos que han sido ejecutados con cada nivel de prioridad.
- **Tiempo promedio de ejecución de los procesos en un nivel de prioridad dado.**
- **Tiempo promedio de ejecución de todos los procesos en el sistema.**
- **Nivel de prioridad con mayor o menor carga de procesos ejecutados.**

### Estructuras de Datos a Implementar

Además de las estructuras de la práctica anterior, se implementarán las siguientes:

1. **TAD ListaProcesos<sup>1</sup>:** Contendrá una lista de los procesos
2. **TAD Árbol Binario de Búsqueda ABBProcesos<sup>2</sup>:** Cada nodo representará un nivel de prioridad, conteniendo la lista de procesos asignados con dicha prioridad y sus tiempos de ejecución acumulados.

Cada nodo del ABB tendrá los siguientes datos:

- **Prioridad del proceso** (el ABB estará ordenado por nivel de prioridad).
- **ListaProcesos:** Lista de procesos asignados por prioridad en orden de llegada. Para cada proceso se almacenará el tiempo total de ejecución.

Para facilitar la organización, el ABB se inicializará con un nodo raíz ficticio que representará una prioridad genérica, con nombre "P0" y una lista vacía de procesos. P0 representa un nivel de prioridad genérico. Los procesos se almacenarán en el **ABBProcesos** cuando terminen su ejecución, una vez hayan sido atendidos. Se creará un nodo para el nivel de prioridad del proceso (con el proceso en la lista de dicho nodo) si este nivel de prioridad aún no existe en el árbol, o se insertará en la lista correspondiente al nodo de esa prioridad si ya existe en el árbol.

---

<sup>1</sup> Con las operaciones básicas y las que sean necesarias para las opciones del menú de la práctica.

<sup>2</sup> Puede utilizarse la clase proceso definida en la práctica anterior, almacenando/actualizando el tiempo de ejecución en el atributo que almacenaba el tiempo (ahora almacena el tiempo total que ha estado el proceso en el sistema).

El programa funcionará mediante un **menú con** al menos, las siguientes opciones (en **negrita las modificaciones** respecto a las mismas opciones de la parte 2 de la práctica 1):

1. Crear la pila de procesos que se iniciarán en el sistema. Los datos se crearán manualmente en el código.
2. Mostrar los procesos que se iniciarán (pila de procesos).
3. Borrar la pila de procesos que se van a ejecutar en el sistema operativo.
4. Simular que han pasado N minutos (N leído de teclado). Se simulará el paso del tiempo, escribiendo en pantalla los datos de los eventos de llegada y/o salida de procesos que vayan realizándose (minuto, PID, PPID, id del núcleo que ejecuta el proceso, estado de los distintos núcleos en ese momento). **Los procesos que son ejecutados se insertan en el ABBProcesos** (según se ha descrito).
5. Mostrar los detalles de los procesos que se encuentran en ejecución en cada uno de los núcleos. (PID del proceso, PPID del proceso, y tiempo)
6. **Consultar** en cualquier momento qué **núcleo** tiene **menos procesos** y cuál es el **más** ocupado.
7. **Consultar** en cualquier momento el **número de núcleos operativos**.
8. **Añadir un proceso** directamente al **ABBProcesos**, leyendo sus datos de teclado.
9. **Mostrar** los datos almacenados en el **ABBProcesos**, ordenados por prioridad. Para cada nivel de prioridad se mostrará el PID del proceso y la lista de procesos con la misma prioridad (PID del proceso y tiempo que ha vivo en el sistema operativo).
10. **Mostrar los procesos con la prioridad dada:** Permite consultar y mostrar todos los procesos que tienen un nivel de prioridad específico.
11. **Mostrar todos los niveles de prioridad que han tenido al menos un proceso ejecutado, en orden numérico:** Lista todos los niveles de prioridad en los que ha habido al menos un proceso atendido, ordenados de menor a mayor prioridad.
12. **Calcular y mostrar el nivel de prioridad con el mayor número de procesos y el de menor número:** Permite identificar los niveles de prioridad que han tenido la mayor y menor cantidad de procesos atendidos.
13. **Calcular y mostrar el tiempo promedio de ejecución de los procesos con una prioridad específica:** Calcula el tiempo medio que los procesos de un nivel de prioridad dado han estado en ejecución.
14. **Calcular y mostrar el tiempo promedio de ejecución de los procesos en cada nivel de prioridad, recorriendo el árbol en preorden:** Calcula y muestra el tiempo medio de ejecución para cada nivel de prioridad, recorriendo el árbol de prioridades en orden preorden.

**OBSERVACIONES:**

- Aquí se está describiendo el funcionamiento básico del programa y las estructuras correspondientes, los alumnos deben tomar decisiones sobre los datos, su implementación, la interface del programa, etc, que deben ser justificadas en la documentación.
- Es decisión de cada grupo la forma en que se mostrará en pantalla el contenido de las estructuras y la información sobre el funcionamiento de la simulación, se valorará la claridad de la misma.
- Una vez entregada la práctica, se realizará una **defensa** en la que el programa debe **funcionar correctamente con los datos de prueba** que se facilitarán. La defensa, **individual y obligatoria para ser calificado**, constará de dos partes:
- Defensa escrita (se realiza sin máquina) se puede traer, si se desea, el código en papel.
- Defensa oral: para su realización es necesario traer el código entregado en el portátil o en un pen drive (en caso de no traer el portátil).

### **NORMAS PARA LA REALIZACIÓN Y ENTREGA DE LA PRÁCTICA**

1. Las prácticas se realizarán en grupos de **dos alumnos** que deberán ser los mismos para las dos prácticas de la asignatura. En caso de que, por motivos justificados, no sea así debe escribirse un correo a [carlos.cilleruelo@uah.es](mailto:carlos.cilleruelo@uah.es) antes del día 15 de noviembre de 2024.
2. La práctica se implementará en C++. Debe entregarse un fichero comprimido incluyendo todos los ficheros fuente del proyecto C++ y el documento descrito en el punto 4. Se subirá un fichero por grupo a la plataforma antes de la fecha indicada. El nombre del fichero será el `nombreaapellido1_nombreaapellido1` de ambos miembros del grupo.
3. En la defensa individual y obligatoria de la práctica se verificará la autoría de la práctica entregada y será calificada con APTO/NO APTO, siendo necesaria la calificación de APTO para poder ser evaluado de la práctica.
4. La entrega de prácticas copiadas supondrá el suspenso de la asignatura en esta convocatoria para todos los alumnos implicados.
5. La documentación que se subirá en un fichero .pdf junto con el proyecto (tendrá un peso del 10% de la nota), deberá tener al menos los siguientes apartados:
  - a. Nombre y DNI de los alumnos del grupo.
  - b. Detalles y justificación de la implementación:
    - b.1 TAD`s creados.
    - b.2 Definición de las operaciones del TAD (Nombre, argumentos y retorno).
    - b.3 Solución adoptada: descripción de las dificultades encontradas.
    - b.4 Diseño de la relación entre las clases de los TAD implementados.
    - b.5 Explicación de los métodos más destacados.
  - c. Explicación del comportamiento del programa.
  - d. Bibliografía y referencias utilizadas.