Proyecto 2

Sistemas Operativos

Segundo Semestre 2022, Prof. Cecilia Hernández

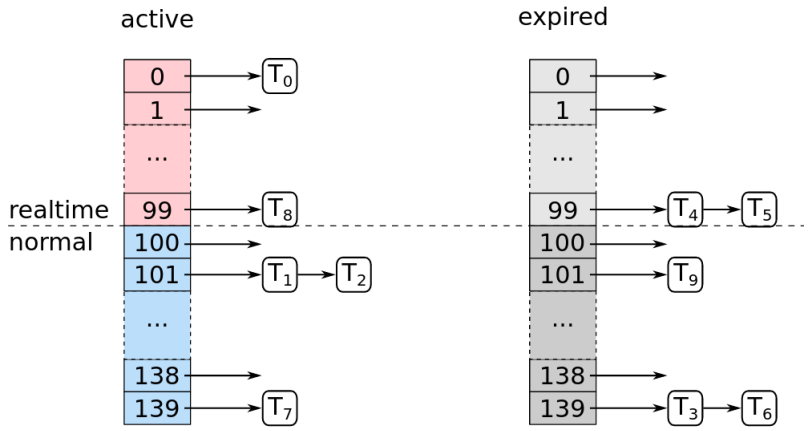
Fecha Inicio: Viernes 28 de Octubre, 2022.

Fecha Entrega: Viernes 25 de Noviembre, 2022 (23:59 hrs).

Trabajo en grupo: Integrado con 3 estudiantes. Una entrega por grupo.

Entrega: Archivo comprimido con readme.txt software e informe.

1. En este proyecto implementará un simulador multi-hebra de un algoritmo de planificación de procesos con M CPUs. Su simulador debe considerar los siguientes aspectos:
   - Debe planificar N procesos que se ejecutan concurrentemente y que simulan procesos a planificar. Cada proceso se debe simular con una hebra. Diremos que estas hebras son de tipo T.
   - Debe haber M hebras especializadas, donde cada una planifica procesos en una CPU específica. A estas hebras las denominaremos del tipo E.
   - Las hebras tipo T deben tener una prioridad dinámica y tiempo requerido para ejecutarse. Asuma que el rango de tiempo de ejecución está dado por un valor aleatorio en un rango $[a, b]$ y es simulado con *sleep*. Las prioridades deben estar entre 0 y 9 (valor mas bajo mayor prioridad). La figura proporcionada mas abajo da el rango de prioridades de linux, las que están definidas entre 0 y 139.
   - El simulador debe mantener una estructura de datos compuesta de dos runqueues, una la denominaremos activa y la otra expirada. El planificador mantendrá las hebras tipo T en estas runqueues, de donde las hebras E irán eligiendo una hebra T a la vez según su prioridad de la runqueue activa. Una vez que la hebra termine su tiempo de ejecución (simulado por sleep), la hebra E que la planificó obtiene el tiempo de cómputo para la siguiente periodo de ejecución y calcula su nueva prioridad. Luego, la agrega a la runqueue expirada en la prioridad correspondiente. Una vez que las hebras E terminen de planificar las hebras en la runqueue activa continuan en el siguiente periodo de planificación procesando las hebras almacenadas en la runqueue expirada, operando de manera similar, es decir, ir agregando las hebras a la runqueue activa. De esta manera, en cada periodo de planificación, el simulador planifica las hebras en runqueue activa, luego en siguiente periodo planifica las hebras en runqueue expirada, en el siguiente periodo planifica las hebras en runqueue activa y asi sucesivamente. La siguiente figura despliega como ejemplo las dos runqueues con prioridades de procesos de linux.



- Su simulador debe definir una forma de calcular prioridades dinámicas que deben ser acorde al tiempo de ejecución en cada periodo de planificación. Para ello debe considerar la prioridad actual y el tiempo de ejecución asignado en el siguiente periodo. A mayor tiempo de ejecución de una hebra en un periodo de planificación, menor debe ser la prioridad en el siguiente periodo.
- Manejo de las ruqueues.
  - Para simular las hebras a agregar a runqueue activa, debe crear hebras de tipo T concurrentemente y definir la runqueue con exclusión mutua para cada prioridad.
  - En cada periodo de planificación la hebras E deben procesar todas las hebras que están en la runqueue activa. Cada hebra procesada por las hebras E deben ser agregadas a la runqueue expirada, según la nueva prioridad calculada por su simulador. Durante este periodo las hebras procesadas deben agregarse a la runqueue activa para continuar con la planificación alternando las runqueues en cada periodo.
  - Asuma que si el tiempo de cómputo para el siguiente periodo de planificación de una hebra T es inferior a un umbral dado la hebra termina su ejecución.
  - Si la runqueue de turno queda vacía las hebras E deben esperar hasta que lleguen nuevas hebras a la runqueue. Para simular este efecto, asuma la existencia de una hebra G que espera por esta condición y genera nuevas hebras tipo T para continuar con la simulación.

2. Entregables
   - Para la entrega debe construir un informe, describiendo el funcionamiento de su simulador junto con los mecanismos concurrentes y de sincronización usados en sus implementación.
   - La entrega debe contener un readme.txt y un ejemplo de como utilizar su simulador.

3. Evaluación
   - Funcionamiento: 80 %. Incluye definición correcta de generación de hebras concurrentes, definición de accesos exclusivos a runqueues, criterios de espera y señalización correctos de coordinación entre hebras. Aspectos de implementación como calidad del código y estructura.
   - Informe: 20 %. Incluye descripción de los algoritmos y análisis de resultados. Adicionalmente, descripción de supuestos y limitaciones de su implementación.