

Tarea N° 2
Sistemas Operativos

Fecha: 08 de Mayo del 2023

Plazo: 3 semanas

Mochila mochila 🎵 ...

El problema de la mochila es un problema clásico en programación, este consiste en un excursionista que debe preparar su mochila, la cual tiene una capacidad limitada y por tanto no le permite llevar todos los artículos que quisiera tener en la excursión. Cada ítem que el excursionista puede incluir en la mochila le reporta una determinada utilidad. Luego el problema consiste en seleccionar un subconjunto de ítems de forma tal que se maximice la utilidad que el excursionista obtiene, pero sin sobrepasar el peso máximo de la mochila

Como el excursionista tiene que prepararse para la salida, le pide a usted que le ayude a decidir qué artículos llevar en la mochila. Para ello deberá:

- 1) Crear un arreglo en el cual guardará todos los objetos que el excursionista puede almacenar en su mochila, la información de los artículos está en el archivo mochila.txt, el cual sigue el siguiente formato:

Número-de-Items(N)	Peso-Máximo-Mochila(K)
ganancia-item1	peso-item1
ganancia-item2	peso-item2
...	
ganancia-itemN	peso-itemN

También deberá tener un registro de la ganancia que tiene la mochila (inicializada en 0) y deberá modificarla en caso de que encuentre una ganancia mejor **(5 puntos)**

- 2) Deberá crear una función que agregue los ítems a la mochila (considere para esta pregunta que un ítem N_i solo se puede agregar 1 vez a la mochila por cada llamado a la función). La función deberá agregar ítems de manera aleatoria. Si al escoger un ítem se sobrepasa el peso máximo de la mochila, este ítem no se agrega y se termina el proceso de llenado. En caso de que se encuentre una ganancia mejor a la actual, la función deberá reportar dicho valor. Considere que esta función deberá ser llamada durante un tiempo de K segundos, en donde K es un parámetro ingresado por el usuario. **(15 puntos)**
- 3) Ahora deberá crear N threads que llamen a la función, en donde cada uno de ellos llene su propia mochila, sin embargo deberá controlar que si un ítem es agregado en una mochila, este no podrá ser agregado en ninguna otra hasta que se termine de usar, es decir, que los artículos deben ser tomados de manera **sincronizada**. Al igual que la pregunta anterior, se deberá imprimir la ganancia solo si esta supera a la mejor

encontrada hasta dicho punto del proceso de búsqueda. Este proceso se deberá repetir por un tiempo de K segundos. **(25 puntos)**

- 4) Finalmente, deberá crear una función, la cual será utilizada por N threads, en donde se podrán agregar sincronizadamente hasta 4 artículos **a la vez** del mismo ítem en distintas mochilas. Al igual que en los puntos anteriores, se deberá reportar la ganancia en caso de mejorar la mejor ganancia encontrada hasta dicho punto de la búsqueda además de repetir el proceso por un tiempo de K segundos. **(10 puntos)**
- 5) Reporte y compare resultados para las tres estrategias utilizando para ello distintos valores de N y K. **(5 puntos)**

OBSERVACIÓN:

Para la resolución de los problemas debe usar herramientas de sincronización vistas en clases, cualquier otro método utilizado no será considerado y no tendrá puntaje.

Condiciones de entrega

Se indican las siguientes condiciones para la entrega de la tarea:

- La tarea se desarrolla de manera **individual**.
- Es requisito indispensable que los códigos compilen. De no cumplirse esta condición, la tarea tendrá un descuento inicial de 20 puntos.
- La tarea se entregará vía el Canvas del curso. La fecha y hora límite para la entrega es el 28 de Mayo a las 23.59. Cualquier entrega atrasada no se considerará y será calificada con la nota mínima de inmediato. **No habrá extensión de plazo.**
- Consultas al correo pablo.saez@mail.udp.cl