

Mini TP - Threads y Semáforos

El Gran Asado de SOR



Propósito y sentido de la actividad	1
Producto final de la actividad	1
Evaluación y entrega	1
Insumos necesarios para el TP	2
Ayuda y consultas	2
Enunciado	3

Propósito y sentido de la actividad

En este tp se practican los conceptos de sincronización y exclusión mutua entre threads. Estos conceptos son importantes para evitar errores y problemas que surgen cuando se trabaja con procesos concurrentes y variables compartidas.

Producto final de la actividad

Al finalizar esta actividad tendremos:

- Un ejemplo de un programa concurrente dividido en threads que realiza una tarea compleja.
- Sabremos compilar y ejecutar nuestro programa dentro del sistema GNU/Linux.

Evaluación y entrega

Para acreditar esta actividad se solicita:

- el código fuente de su implementación en formato de texto plano (archivo .c)
- un informe en pdf del trabajo realizado punto por punto, dificultades encontradas y soluciones propuestas.

Esta actividad es grupal, obligatoria y con nota.

El presente trabajo se califica con las notas:

- ★ I (insuficiente)
- ★ A- (aprobado menos, no puede tener dos A- en la cursada)
- ★ A (aprobado)
- ★ A+ (aprobado más, redondea para arriba la nota final en caso de promocionar)

Fecha de entrega: Consultar el calendario.

Espacio de entrega: Por Moodle

Recuperatorio: En caso de no aprobar tiene un plazo de una semana para entregar el TP con las correcciones indicadas más ejercicios adicionales que se agregaran al enunciado. En recuperatorio no se pone A +.

Insumos necesarios para el TP

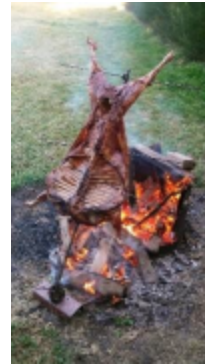
Para realizar esta actividad es necesario trabajar sobre un sistema GNU/Linux.

Ayuda y consultas

Para realizar consultas sobre algún error que surja durante los experimentos debe enviar una captura de pantalla del error y la secuencia de pasos que realizaron para obtener ese error. Tendremos disponible un canal de consultas en Telegram, así como también los mails de los profesores.

Enunciado

Manucho que cursó en el segundo cuatrimestre de 2023 la materia Sistemas Operativos y Redes, decide hacer un gran asado en su casa para festejar que promocionó la materia. Como además Manucho hace poco ganó el Loto plus y no escatima en gastos, decide contratar un servicio de catering.



El servicio consiste de 1 asador y M mozos. A la casa de Manucho asistirán N invitados. Los comensales se sientan a la mesa, siendo Manucho el último en sentarse. Una vez que se sienta a la mesa Manucho, los M mozos sirven la comida. Por problemas de organización de la empresa de catering, la cantidad de mozos es menor a la cantidad de invitados. Los mozos no deben servir más de $N+1$ platos. Cualquier comensal incluido nuestro querido Manucho puede comenzar a comer, incluso si no han terminado de servir a todos los invitados. A Manucho le encanta hablar sobre fútbol en especial sobre el mundial. Cuando Manucho termina de comer hace una pregunta sobre los favoritos a ser campeones del mundo. Esta pregunta puede ser respondida por cualquier invitado/a que haya terminado de comer. Solo un invitado debe responder la pregunta. Como es de esperarse en un asado, habrá vino así que los ánimos sobre cuestiones de fútbol se exacerban y Manucho se enoja muchísimo por la respuesta un invitado/a se levanta y abandona el quincho. Si Manucho ya se ha levantado, los invitados pueden hacer lo mismo, pero solo si ya han terminado de comer.

Las primitivas disponibles son:

Sentarse()

Servircomida()

Enojarse()

Levantarse()

Lanzar_pregunta_mundialista()

Lanzar_respuesta_mundialista()

Consignas:

1. Realizar una solución usando semáforos para el problema descrito
2. Explicar brevemente cómo funciona la solución indicando que semáforos necesitó y para qué; y cuántos procesos diferentes tuvo.
3. ¿En algún punto de la solución se puede producir inanición? ¿Dónde? ¿Por qué?
4. Implementar la solución en lenguaje C utilizando semáforos y threads

Nota: los docentes también podríamos comer asado.