

Laboratorio 4: Sistemas Operativos

Profesor: Viktor Tapia

Ayudantes de cátedra: Muryel Constanzo y Nicolás Schiaffino

Ayudantes de Laboratorio: Ian Rossi y Luciano Yevenes

9 de noviembre de 2025

1. Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje Python 3 (o superior). Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso y ejecución de sus programas junto a cualquier indicación que sea necesaria.

2. Tarea

¡Finalmente lo lograron! Despues de escalar el volcán y llegar a la cima, el equipo de rescate los ha localizado. Sin embargo, hay un problema: debido a las condiciones meteorológicas adversas, el helicóptero no podrá aterrizar hasta dentro de 6 horas.

Mientras esperan, el líder del equipo nota algo extraordinario: en el cráter del volcán hay cristales volcánicos raros de alto valor científico y comercial. Estos cristales podrían financiar futuras expediciones e investigaciones.

El equipo decide que no pueden desperdiciar esta oportunidad. El problema es que los cristales están distribuidos en zonas de alta actividad geotérmica, donde las temperaturas y los gases tóxicos varían constantemente. Además, cada miembro del equipo tiene una resistencia limitada a estas condiciones extremas.

3. Especificaciones

Deben simular un sistema de extracción coordinada donde múltiples trabajadores (representados por hebras) compiten y colaboran para extraer la mayor cantidad de recursos posible antes de que llegue el rescate, sin comprometer su seguridad.

3.1. Fases del experimento

3.1.1. Fase de Inicialización del Equipo

Se crean 64 hebras, cada una representando a un trabajador del equipo. Cada trabajador tiene:

- **Resistencia inicial:** 100 puntos (representa su capacidad para trabajar en condiciones extremas)
- **Mochila:** Capacidad para almacenar hasta 10 cristales

- **Estado:** Activo, Agotado o Intoxicado

Al inicio, se distribuyen aleatoriamente 256 cristales en 16 zonas de extracción diferentes (cada zona tiene entre 10-20 cristales).

3.1.2. Fase de Exploración y Extracción

Los trabajadores se mueven de forma aleatoria entre las zonas de extracción. En cada ronda:

1. Cada trabajador intenta entrar a una zona para extraer cristales
2. **Las zonas tienen capacidad limitada:** máximo 8 trabajadores simultáneamente
3. Cada trabajador puede intentar extraer entre 1-3 cristales por ronda (aleatorio)
4. **Costo de extracción:** Cada cristal extraído reduce la resistencia del trabajador:
 - 1 cristal: -5 puntos de resistencia
 - 2 cristales: -12 puntos de resistencia
 - 3 cristales: -20 puntos de resistencia
5. **Condiciones ambientales:** Cada zona tiene un nivel de toxicidad (10-50 puntos) que afecta a todos los trabajadores presentes:
 - Si un trabajador permanece en una zona, pierde puntos de resistencia igual a: $\text{toxicidad}/\text{número_de_trabajadores_en_zona}$
 - Esto simula que trabajar en grupo distribuye la exposición
6. Competencia por recursos: Si múltiples trabajadores intentan extraer del mismo yacimiento:
 - 70 % de probabilidad de extracción exitosa
 - 30 % de probabilidad de conflicto (nadie extrae, ambos pierden 10 puntos de resistencia adicionales)

Cada ronda dura 15 segundos.

3.1.3. Fase de Gestión de Estados

- Si un trabajador alcanza resistencia ≤ 20 , queda **Agotado** y debe regresar al campamento base (deja de trabajar por 1 ronda para recuperar 30 puntos)
- Si un trabajador alcanza resistencia ≤ 0 , queda **Intoxicado** y es evacuado permanentemente (su hebra termina)
- **Los trabajadores pueden compartir recursos:** si un trabajador **Activo** se encuentra con uno Agotado, puede darle hasta 3 cristales para motivarlo (recupera 15 puntos de resistencia)

3.1.4. Fase de Recolección Final

El proceso se repite por 8 rondas (simulando las 6 horas de espera). Al final:

- Se contabiliza el total de cristales extraídos por el equipo
- Se evalúa cuántos trabajadores sobrevivieron
- Se calcula la eficiencia: $(\text{cristales_extraídos} * \text{trabajadores_vivos}) / 256 \text{ cristales totales}$

3.2. Outputs

- **inicializacion.txt:** Lista de trabajadores, zonas y distribución inicial de cristales
- **ronda_X.txt:** Registro de:
 - Qué trabajador entró a qué zona
 - Cuántos cristales extrajo
 - Resistencia actual
 - Conflictos ocurridos
 - Trabajadores agotados/intoxicados
- **intercambios.txt:** Registro de cristales compartidos entre trabajadores
- **extraccion_final.txt:**
 - Total de cristales extraídos
 - Cristales por trabajador
 - Trabajadores evacuados
 - Eficiencia del equipo

3.3. Requisitos de Implementación

- Usar hebras (threads) para representar cada trabajador
- Controlar la sincronización con locks para:
 - Acceso a las zonas (capacidad limitada)
 - Extracción de cristales (evitar que dos trabajadores extraigan el mismo cristal)
 - Cambios de estado (Activo → Agotado → Intoxicado)
- Evitar condiciones de carrera al actualizar:
 - Inventario de zonas
 - Mochila de trabajadores
 - Resistencia
- Cada hebra debe mantener su historial: cristales extraídos, zonas visitadas, rondas trabajadas
- Asegurarse de que todas las hebras finalicen correctamente antes de terminar el programa
- **Implementar un sistema de prioridad:** trabajadores con menos cristales tienen prioridad para entrar a zonas con más recursos

3.4. Implementación con threading

Se debe hacer uso de esta librería (y no otra) para la implementación de las hebras en la tarea.
Se recomienda usar:

- `threading.Lock()` para proteger recursos compartidos
- `threading.Semaphore()` para controlar el acceso limitado a las zonas
- `threading.Barrier()` para sincronizar el inicio de cada ronda

4. README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los algoritmos y desarrollo realizado.
- Instrucciones de como compilar y correr el código.
- Estrategias de sincronización implementadas
- Supuestos utilizados.

5. Consideraciones Generales

- El uso de librerías está estrictamente limitado a aquellas esenciales para la implementación del laboratorio. Se permite el uso de:
 - Bibliotecas estándar básicas como `time`, `math`, `random` en Python3.
 - `threading` para la implementación de hebras.

En caso de requerir alguna librería adicional no listada explícitamente, su uso debe ser consultado previamente con los ayudantes y autorizado formalmente.

Otras consideraciones:

- Se deberá trabajar **OBLIGATORIAMENTE** en parejas.
- Deberá estar subido al Github correspondiente a mas tardar el día 23 de Noviembre a las 23:59 horas.
- Se descontarán 10 puntos por cada hora o fracción de atraso.
- Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje Python. Se asume que usted sabe programar en este lenguaje, ha tenido vivencias con él, o que aprende con rapidez.
- El código debe ser entregado en forma de **1 archivo .py** nombrado en base al formato "`LAB4_ApellidoIntegrante1_ApellidoIntegrante2`"
- Los códigos serán pasados por un software anti-plagio, en caso de ser detectada copia o uso de una IA para la totalidad del Laboratorio, se pondrá nota 0, hasta que se realice una reunión con los grupos involucrados.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- Las tareas serán ejecutadas en Linux, cualquier tarea que no se pueda ejecutar en dicho sistema operativo, partirá de nota máxima 60.
- Las preguntas deben ser hechas por Aula a través del foro o servidor de Discord. De esta forma los demás grupos pueden verse beneficiados también.

- Si no se entrega README, o si su programa no funciona, la nota es 0 hasta la corrección.
- Se descontarán hasta 50 puntos por:
 - No respetar el formato de entrega.
 - No respetar el formato del README.
 - Solicitar edición de código al momento de revisar.
 - Código poco prolijo y mal estructurado (ausencia de indentación adecuada, falta de consistencia en el estilo, nombres de variables confusos o poco descriptivos, comentarios insuficientes o irrelevantes).
- **Una vez publicadas las notas tendrán 5 días para apelar con el corrector que les revisó, después de este plazo las notas no tendrán ningún tipo de cambio.**