

Proyecto 2: Alien's Community.

1. Motivación.

Uno de los problemas con los que debe lidiar el SO es la asignación y calendarización de procesos, así como contabilizar en todo momento el inventario de recursos que posee el sistema. Generalmente el tema de calendarización y administración de los recursos no es un tema trivial, ya que involucra un alto componente de conocimiento sobre los conceptos básicos de SO.

El enfoque de este proyecto es aplicar los algoritmos y conceptos vistos en clase, así como investigar sobre la implementación de hilos.

2. Objetivo.

Aplicar los algoritmos de calendarización básicos, así como el uso de directivas para interactuar a bajo nivel con el procesador, con el fin de solucionar problemas de calendarización de procesos livianos.

3. Información general

- Valor del proyecto: 17%
- Nombre del programa: Alien's Community
- Este proyecto se debe implementar en grupos de 3 estudiantes.
- Cualquier fraude detectado significará una nota de cero y su proceso respectivo.
- El proyecto se debe de realizar utilizando sus propias bibliotecas de hilos (No se permite el uso de externas) y con el lenguaje de programación en C, en caso contrario la nota será cero.

4. Requerimientos funcionales.

A. LPthreads.

Se deberá reimplementar la biblioteca Pthreads (utilizando las directivas de C, se aclara que un wrapper o interface no es reimplementación) con el nombre **LPthreads**, la cual incluye las siguientes funciones.

- ✓ Lthread_create
- ✓ Lthread_end
- ✓ Lthread_yield
- ✓ Lthread_join
- ✓ Lthread_detach
- ✓ Lmutex_init

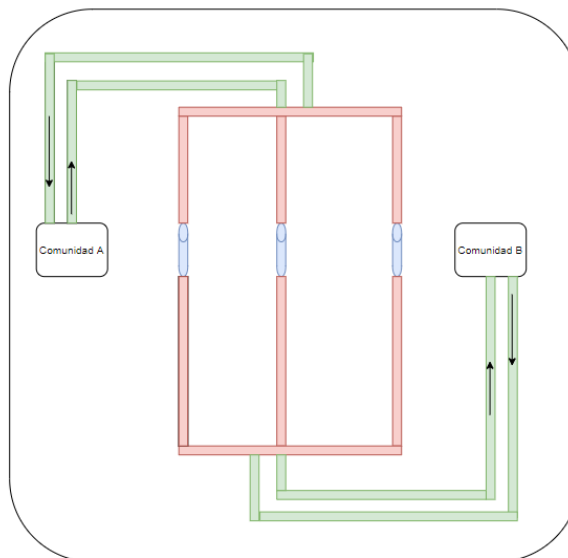
- ✓ Lmutex_destroy
- ✓ Lmutex_unlock
- ✓ Lmutex_trylock

La idea fundamental es que un código que se ejecute con Pthreads también lo haga con LPthreads.

B. Comunidades fantasma.

Existen dos comunidades de Alien, las cuales son llamadas Comunidad A y Comunidad B. En estos lugares es donde ocurre la generación de Aliens. Estos fantasmas tienen el objetivo de llegar a la otra comunidad opuesta de manera civilizada. Hay vías para entrar y salir de cada comunidad, por los cuales los Aliens deberán de desplazarse. Todas las demás vías son de doble sentido (una vía de norte-sur y una vía de sur-norte). En la pantalla principal deberá crearse un mecanismo para crear los diferentes tipos de Aliens en la comunidad deseada.

En la figura 1 se muestra un ejemplo de la distribución de los elementos básicos, se espera que la interfaz gráfica de este proyecto sea impecable, ya que es la manera de poder revisar los aspectos funcionales.



C. Puentes Colgantes.

Existen tres puentes llamados Oeste, Central y Este, los cuales regulan el tránsito en una única vía (de manera simultánea), ya sea hacia el norte o hacia el sur. La capacidad del puente dependerá de cuanto pesen los Aliens, por ejemplo, si un Alien pesa 30 y el puente soporta 60, entonces soportará 2 Aliens simultáneamente. El largo del puente y todos los parámetros necesarios para los mismos, serán configurables en un archivo de texto (uno por cada puente). El tránsito por los puentes (configurable) será regulado por alguno de los siguientes algoritmos:

- *Algoritmo Y:* Este algoritmo trata de dejar pasar Y (configurable) Aliens del lado norte y posteriormente Y Aliens del lado sur. En caso de que no hayan Aliens de un lado, el algoritmo debe dejar pasar Aliens del lado que si haya. En caso de que haya Aliens de ambos lados, pero cantidades menores a Y, se deberá dejar pasar del lado que haya más, se deberá mostrar en pantalla el contador de los Aliens que pasan.
- *Algoritmo Semáforos:* Este algoritmo consiste en dejar pasar Aliens, M segundos del lado norte y N segundos del lado sur. Deberá mostrarse semáforos que indiquen esta situación.
- *Algoritmo Supervivencia:* Este algoritmo consiste en que un Alien pueda pasar apenas pueda, sin ocasionar algún accidente con otro Alien.

D. Aliens

Existirá tres tipos de Alien:

- Alien normal: Tienen una velocidad moderada (configurable) y no hay tratamiento especial.
- Alien alfa: Tiene una velocidad de 20% mayor que los Aliens de tipo normal.

- Alien beta: Tienen prioridad máxima y un tiempo de ejecución máximo (configurable), debe de tener tratamiento de proceso en un RTOS (hard, en caso de no ejecutarse lanzará un mensaje de que si desea continuar o detener el programa). Tiene una velocidad de 50-200% mayor a la de los Aliens normales, caben destacar que tienen prioridad máxima sobre los otros Aliens.

E. Calendarizadores.

Los Aliens deberán estar ordenados en los extremos según el calendarizador de cada puente (configurable), la cantidad de Aliens que debe estar ordenados (en cada lado) también es configurable.

Los posibles calendarizadores son:

- ✓ Round Robin (expropiativo)
- ✓ Calendarizador por prioridad: Los hilos necesitaran un parámetro extra, el cual corresponderá a la prioridad.
- ✓ El más corto primero (expropiativo): Necesitará un parámetro extra por ejemplo cuánto dura el Alien en atravesar el puente (calculado con forme el largo del puente y la velocidad).
- ✓ Primero en llegar primero en salir.
- ✓ Lotería.

F. Generación de Aliens.

El programa proporcionará dos modos de generación (el usuario debe de elegir) de Aliens:

- Generación automática: La generación de Aliens será aleatoria siguiendo una distribución exponencial de tiempos (media configurable). El porcentaje del tipo de Aliens también es configurable en un archivo de texto independiente de los puentes, por ejemplo 50% normales, 30% beta y 20% alfa.
- Generación manual, se deberá crear un mecanismo de generación de Aliens, tomando en cuenta el tipo y la comunidad (el mecanismo puede ser por medio teclas o botones).

G. Interfaz gráfica.

En la interfaz gráfica se debe de visualizar todo lo descrito anteriormente, el movimiento de los Aliens debe ser animado y

diferenciar los tipos. El movimiento de los Aliens debe ser fluido para observar el comportamiento del software. Debe ser impecable, es decir, amigable al usuario.

H. Interrupciones.

El usuario podrá eliminar a un Alien específico presionando clic encima de la imagen.

I. Intersecciones.

Las comunidades de Aliens son “civilizadas” por lo que se deben controlar para que no ocurran accidentes. Las avenidas tienen la prioridad.

J. Implementación.

Cada Alien será un hilo, por lo que se debe plantear la incógnita si se necesita un único tipo de hilo o no.

K. Intruso.

El usuario podrá crear un intruso, siempre y cuando no haya uno en pantalla, cuya misión es eliminar a los Aliens alfa, tiene capacidades especiales como rebasar los Aliens que están adelante con el fin de alcanzar a los Aliens alfa. La ruta y el punto de partida es aleatorio (cada vez que se genere). La velocidad de este intruso será el doble de un Alien normal y podrá rebasar a los Aliens sin ocasionar accidentes con otros Aliens.

5. Requerimientos técnicos.

- ✓ Este proyecto se debe realizar en el lenguaje de programación C.
- ✓ Es inaceptable el error de segmentation fault o un deadlock.
- ✓ Debido a que es un problema de calendarización y sincronización, es necesario que el diseño de la solución sea muy robusta y que esté preparado para cualquier combinación de eventos, la idea fundamental es indicar al usuario algún error o problema ocurrido, siempre y cuando sea ocasionado por el mismo. En caso de que se produzca un fallo, el sistema debe de terminar de manera controlada y elegante.

6. Documentación (Estilo IEEE-Trans máximo 6 páginas).

- ✓ Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- ✓ Ambiente de desarrollo: Todos los detalles de implementación y herramientas durante el desarrollo del proyecto.
- ✓ *Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto y de qué manera (Documento aparte).
- ✓ Detalles del diseño del programa desarrollado, del software: **Diagramas UML (equivalente a clases), arquitectura**, dominio, funcionalidades, imágenes, **descripciones** entre otros.
- ✓ Manual de instrucción: instrucciones de cómo se debe de utilizar el programa.
- ✓ Tabla de actividades por cada estudiante: bitácora con total de horas.
- ✓ Conclusiones
- ✓ Sugerencias y recomendaciones.
- ✓ Referencias

7. Entregables

- ✓ Código fuente con documentación interna.
- ✓ Documentación.

8. Evaluación

- ✓ Puentes: 10%
- ✓ Generación 10 %
- ✓ Algoritmos: 20%
- ✓ Interfaz: 20%
- ✓ Biblioteca 20
- ✓ Documentación: 20%

9. Fecha de entrega.

- ✓ 26 de Julio 11:55, únicamente por tecdigital.

10. Otros aspectos administrativos.

- ✓ Para la revisión del proyecto se debe de entregar tanto la documentación como la implementación del software.
- ✓ El proyecto deberá ejecutarse en la máquina del profesor, por lo que debe proporcionar un make que realice todo lo necesario y genere los ejecutables con sus debidos permisos. El profesor no hará ningún tipo de configuración para ejecutar el proyecto. En caso de que no se pueda ejecutar la nota será cero.

- ✓ En caso de que no compile el proyecto la nota será cero.
- ✓ **No se reciben trabajos después de la hora indicada.**
- ✓ En la revisión del proyecto pueden estar presentes el coordinador y asistente.
- ✓ Es responsabilidad del estudiante proveer los medios para poder revisar la funcionalidad del software, en caso de que una funcionalidad no se pueda verificar la nota será cero en el rubro correspondiente.
- ✓ No se aceptan proyectos entregados por otros medios que no sea Tecdigital.

11. Atributos que evaluar.

Herramientas de ingeniería. Capacidad para crear, seleccionar, aplicar, y adaptar apropiadamente técnicas, recursos y herramientas modernas de ingeniería y de tecnologías de información, incluyendo predicción y modelado de problemas complejos de ingeniería, con la comprensión de las limitaciones asociadas.

Se debe de especificar qué herramientas, técnicas o recursos se utilizaron para dar la solución al problema.

Aprendizaje continuo. Capacidad para reconocer las necesidades propias de aprendizaje y la habilidad de vincularse en un proceso de aprendizaje independiente durante toda la vida, en un contexto de amplio cambio tecnológico.

Se debe de colocar todos los aspectos (Estrategias, modelos, técnicas entre otros) que se aprendieron nuevos, así como identificar aquellas áreas donde se debe de mejorar, tanto en habilidades duras como blandas y cómo piensan suplir estas deficiencias.