

Proyecto 2 - Crazy Ants

Emanuel Esquivel López, Fabricio Elizondo Fernández, Roger Valderrama
ema11412@estudiantec.cr, faelizondo@estudiantec.cr roger.andres18@estudiantec.cr

Área Académica de Ingeniería en Computadores
Instituto Tecnológico de Costa Rica

I. INTRODUCCIÓN

Los sistemas operativos actualmente son muy codiciados en la actualidad, por lo que la implementación y manejo de estos de una correcta forma es muy importante, como sabemos son los encargados que el usuario interactúe de manera fácil con software y hardware, sin estos hoy en día no todos podrían utilizar un computador, por lo que en general es mucha responsabilidad la que tienen los desarrolladores en la actualidad.

Una fuerte de los sistemas operativos es la capacidad de interactuar de manera adecuada con el hardware y además de esto un aspecto muy importante, los procesos ya que estos son capaces de ser ejecutados en un computador, de manera secuencial o paralela, por lo que el manejo de estos es muy importante ya que un fallo en esto puede llegar a ser fatal para el sistema, por lo que existen distintos métodos para el control de procesos, tanto como para calendarizar estos mismos, lo que quiere decir es que en que orden deben ejecutarse los procesos, así como cuando hay choque entre procesos, existen algoritmos encargados de controlar cuales procesos deben ser ejecutados, definidos con mediante distintas métricas, por ejemplo un método muy conocido es el semáforo, el cual identifica intersecciones entre procesos, los cuales prioriza al proceso en estado disponible.

II. AMBIENTE DE DESARROLLO

El proyecto fue desarrollado en linux, por lo que mucha de su implementación y elaboración fue con el uso de comandos en la terminal del mismo, esta fue desarrollada en el lenguaje de programación C.

Se llevo a cabo en una computadora de arquitectura 64 bits, sistema operativo Linux Kali linux 2020.2, 16 Gb RAM, procesador Ryzen 7 2700x, además de la edición de código en VS Code.

III. ATRIBUTOS

Durante este proyecto recalco la importancia de la creación de servidores, esto permitió el aspecto técnico fuera reforzado ya que se requirió un repaso previo de los lenguajes a utilizar, además de que el uso de nuevas herramientas como SDL.

El compromiso es un atributo muy importante el cual estuvo en juego en la elaboración del proyecto por lo que siempre la constancia y perseverancia estaban siempre ahí para la realización del proyecto, por lo que siempre el trabajo se realizó de manera comprometida.

Se fortaleció el atributo de análisis de problemas y herramientas de ingeniería ya que en un inicio no se contaba con la infracción necesaria para la resolución del mismo por lo que se llevo a cabo la investigación necesaria.

IV. DETALLES DE DISEÑO

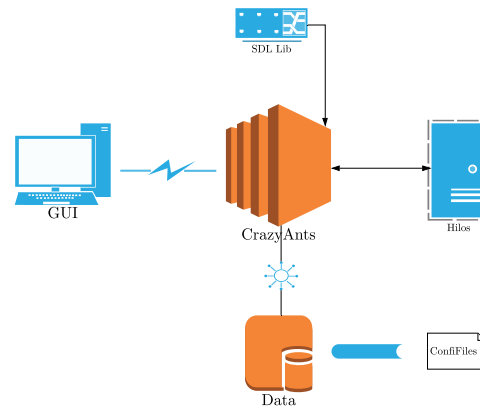


Figura 1. Diagrama de arquitectura del sistema

En este diagrama se puede ver la interacción principal entre todos los elementos, como parte principal o main esta el Crazy Ant's, la cual interactúa directamente con la biblioteca de hilos *CEthreads*, con la cual se hace la generación de las hormigas en las listas, las cuales se generan en el main, esto siempre llamando a una data central la cual carga los elementos de los config files los cuales son los necesarios para que se de el comportamiento deseado por los usuarios.

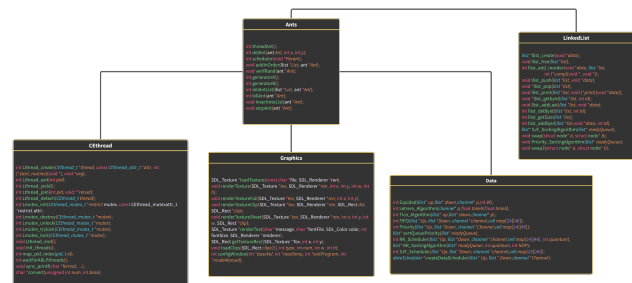


Figura 2. Diagrama de clases del sistema

V. INSTRUCCIONES DE USO

1. Instalación:

Para esta parte es importante tener los archivos necesario por lo que se debe ejecutar el siguiente comando como administrador para realizar la instalación:

```
1 sudo ./install.sh
```

2. Compilación:

Para esta etapa se corre el script run.sh el cual genera el ejecutable llamado Ants en la carpeta **bin**

```
1 ./run.sh
```

3. Ejecución:

Al abrir el programa, se muestra un menú como en la figura 3, en el cual se debe seleccionar un modo de operación entre automático o manual, se considera modificar opciones, y se da click en Iniciar. De primera entrada, se muestran las instrucciones del juego el cual es similar a la figura 4, en el cual a partir de las teclas de los números del 1 al 6, se crean las hormigas, como se aprecia en la figura 5.



Figura 3. Diagrama de instrucciones del sistema

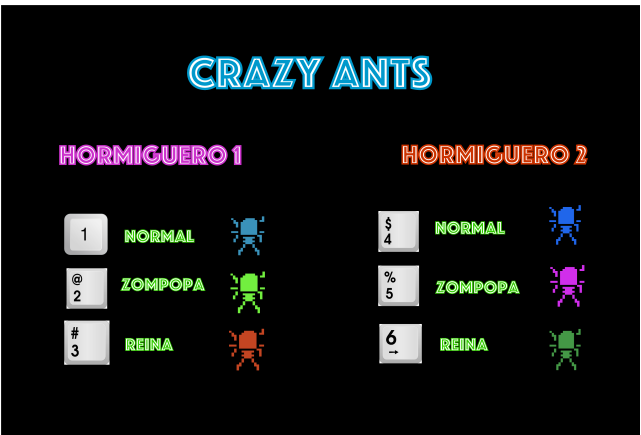


Figura 4. Diagrama de instrucciones del sistema



Figura 5. Sistema en ejecución

VI. TABLA DE ACTIVIDADES

VI-A. Emanuel Esquivel

Actividad	Horas
Investigación sobre hilos	5
Investigación sobre SDL	2
Investigación de creación de lock y multiproceso	5
Implementación de la biblioteca de hilos CETHreads	15
Implementación de la parte gráfica	15
Documentación	2
Total	44

VI-B. Roger Valderrama

Actividad	Horas
Investigación sobre SDL	6
Investigación sobre recursos visuales	5
Implementación del Menu Inicial	6
Implementación del Mapa	20
Documentación	2
Total	39

VI-C. Fabricio Elizondo Fernández

Actividad	Horas
Investigación sobre hilos	5
Investigación sobre Calendarizadores	10
Implementación de la biblioteca CETHreads	3
Implementación de los algoritmos de calendarización	15
Implementación de la lógica de movimiento	5
Documentación	3
Total	41

VII. CONCLUSIONES

Los calendarizadores como se vio son de gran ayuda en el orden natural de los procesos ya que en una computadora es muy común el choque entre procesos, por lo que ordenarlos definiendo varios criterios es muy importante, así como también el poder ordenar de manera correcta y adecuada los distintos procesos presentes en una computadora. En adición a los calendarizadores, en este proyecto se contempló la importancia de generar threads para una mejor administración de los procesos en el sistema, que en este caso correspondían con las hormigas que se generan. Los threads son procesos livianos, que nos permiten generar una adecuada independencia, de manera que no existan conflictos o problemas a la hora de ejecutar procesos concurrentes en una misma aplicación. Por supuesto, es necesario una correcta sincronización y calendarización, para asignar los tiempos y el orden de ejecución adecuados, de manera que los distintos algoritmos implementados en este proyecto, nos demuestran la importancia de planificar procesos en el ámbito de los sistemas operativos, con el objetivo de evitar riesgos que lleguen a afectar componentes esenciales de una computadora, como memorias, procesador, periféricos, etc.

VIII. SUGERENCIAS Y RECOMENDACIONES

Se recomienda utilizar la biblioteca SDL para el manejo de la interfaz gráfica ya que esta es muy sencilla, y de fácil implementación, por lo que en general esto es de mucho beneficio para el manejo de matrices. Por otra parte, este proyecto contaba con el requerimiento de implementar una biblioteca personalizada de threads, lo cual conlleva muchos cuidados a la hora de manejar errores de segmentación de memoria. Lo recomendable en este caso, es realizar la codificación del proyecto utilizando una biblioteca ya implementada, como es el caso de *pthreads* en C, probando cada una de las funcionalidades de la aplicación. Posteriormente, se realiza una implementación de las funciones de *pthreads* en una biblioteca personalizada utilizando las directivas de C.

REFERENCIAS

- [1] Generating random numbers of exponential distribution. (s. f.). Stack Overflow. Recuperado 22 de abril de 2021, de <https://stackoverflow.com/questions/34558230/generating-random-numbers-of-exponential-distribution>
- [2] Memoria compartida en C para Linux. (s. f.). chuidiang. Recuperado 20 de abril de 2021, de http://www.chuidiang.org/linux/ipcs/mem_comp.php
- [3] Use C language to generate Poisson distribution random number instance source code. (s. f.). Alibabacloud. Recuperado 22 de abril de 2021, de https://topic.alibabacloud.com/a/use-c-language-to-generate-poisson-distribution-random-number-instance-source-code_1_31_20012031.html