

# Proyecto #1

**Instituto Tecnológico de Costa Rica**

**Carrera:** Bachillerato en Ingeniería en Computación

**Curso:** Principios de Sistemas Operativos

**Profesor:** Kevin Moraga García

**Alumnos:** Alberto Zumbado Abarca Jonathan Quesada Salas

**Proyecto 1:** MultiDisplay Animator

---

## Introducción

Primeramente en dicha asignación se establece realizar la implementación de la biblioteca de pthreads en el espacio de usuario, esto permite al estudiante comprender cómo se puede programar un subsistema del sistema operativo, sin tener que hacer cambios en el kernel.

Para ello se deberá realizar una re-implementación de la biblioteca de pthreads, llamada mypthread, de las siguientes funciones:

- my\_thread\_create
- my\_thread\_end
- my\_thread\_yield
- my\_thread\_join
- my\_thread\_detach
- my\_mutex\_init
- my\_mutex\_destroy
- my\_mutex\_lock
- my\_mutex\_unlock
- my\_mutex\_trylock

Por otra se encuentra la parte de Schedulers, la cual deberá contemplar los siguientes apartados:

- **Scheduler RoundRobin:** Se debe de realizar la implementación del scheduler siguiendo un algoritmo de RoundRobin
- **Scheduler Sorteo:** Se debe de realizar la implementación del scheduler siguiendo un algoritmo de Sorteo.
- **Scheduler de Tiempo Real:** Se debe de realizar la implementación del scheduler siguiendo un algoritmo de Tiempo Real.

Adicionalmente un objetivo principal de dicho proyecto es en realizar un sistema de animación consiste en una especie de Flash para ASCII que permite crear animaciones. Para esto se debe de establecer un propio lenguaje de programación para poder describir cualquier tipo de animación. El cual deberá contemplar los siguientes aspectos:

- **Monitores de despliegue:** Permitirá correrse en distintas PCs. Cada computadora aportará su monitor como mecanismo de despliegue.
- **Tamaño del canvas:** deberá de poder establecer el tamaño del canvas, este tamaño incluye: qué sección del canvas corresponde a cual de los monitores.

- **Descripción de Objetos:**

- **Tipos:** Se establecerá un mecanismo para definir el tipo de objeto.
- **Forma:** El lenguaje permitirá la creación de nuevas formas, basadas en ASCII art,

- **Restricciones de tiempo de inicio y final:** consiste en el momento en que el objeto entra en escena y cuando es el momento máximo para que el objeto salga de la escena.

- **Descripción de movimiento de Objetos:** podrán moverse en cualquier dirección en el canvas, además se podrán realizar rotaciones de 0, 90, 180, 270 grados.

- **Límite de espacio de objetos:** Ningún objeto puede utilizar el espacio, en el canvas, que otro objeto ya posea.

- **Creación de una animación de prueba:** Se debe de crear una animación de prueba que permita mostrar el funcionamiento de todos los requerimientos anteriores.

---

## Ambiente de desarrollo

Con lo que respecta al ambiente de desarrollo se usará el sistema operativo Ubuntu por parte de Jonathan Quesada Salas y Alberto Zumbado Abarca, específicamente la versión:

```
Ubuntu 20.04.2 LTS
```

En cuanto a los IDLE se que usarán serán el Visual Studio Code para el desarrollo de la tarea y adicionalmente IntelliJ para la funcionalidad del debugger que tiene dicho IDLE para el lenguaje de programación Rust.

Se usará el lenguaje de programación Rust para la resolución de dicha asignación de proyecto número uno de principios de sistemas operativos.

---

## Estructuras de datos usadas y funciones

Primeramente con lo que respecta a las funcionalidades de `my_pthread`, se establece 2 estructuras principales, las cuales son **ThreadControlBlock** y **ThreadPool**:

- **ThreadControlBlock:** Esta estructura de datos se va a encargar de inicializar un determinado control en los hilos, la cual contendrá: el identificador del hilo, la función que se utiliza, los argumentos de la misma y el número de tickets.
- **ThreadPool:** Esta estructura actuará como una piscina de la información de los hilos y estos mismo tendrán un contador de los hilos creados, el identificador del scheduler específicamente el de lottery, los hilos que tenga por parte del scheduler de RoundRobin, también el de RealTime y el Lottery.

En cuanto respecta a las funciones que tienen como tal la biblioteca de `my_pthread`, se encuentran las siguientes:

- **my\_thread\_create:** Esta función tiene como entradas el scheduler a crear con el hilo, la función que se quiera ingresa, adicionalmente se tiene los argumentos de la misma función y por último tendrá el número de tickets que tendrá el hilo a crear. Esta función principalmente se encarga de función inicia un nuevo hilo en la llamada proceso.

- **my\_thread\_end:** Esta función tiene como principal objetivo terminar el proceso de ejecución de un hilo.
- **my\_thread\_yield:** La función permite que un subproceso ceda el control de un procesador para que otro subproceso pueda tener la oportunidad de ejecutarse. El parámetro de la función debe ser NULL, porque los valores que no son NULL están reservados.
- **my\_thread\_join:** El hilo que espera, debe llamar a la función. Esta llamada hace que el hilo se "duerma" hasta que el otro hilo termine. Si el otro hilo ya había terminado, la función sale inmediatamente.
- **my\_pthread\_detach** La función indica que los recursos del sistema para el subproceso especificado deben reclamarse cuando finaliza el subproceso. Si el subproceso ya finalizó, los recursos se reclaman inmediatamente. Esta rutina no hace que el hilo termine.
- **my\_mutex\_init:** La función inicializa el mutex al que hace referencia mp con el tipo especificado por type . Tras una inicialización exitosa, el estado del mutex se inicializa y desbloquea.
- **my\_mutex\_destroy:** La función destruirá el objeto mutex al que hace referencia mutex; el objeto mutex queda, en efecto, sin inicializar. Una implementación puede hacer que establezca el objeto al que hace referencia mutex en un valor no válido.
- **my\_mutex\_lock:** El bloqueo mutex solo lo liberará el subproceso que lo bloqueó. Por lo tanto, esto garantiza que una vez que un subproceso haya bloqueado un fragmento de código, ningún otro subproceso podrá ejecutar la misma región hasta que el subproceso que lo bloqueó lo desbloquee.
- **my\_mutex\_unlock:** Si hay subprocesos bloqueados en el objeto mutex cuando se llaman, lo que hace que el mutex esté disponible, la política de programación se usa para determinar qué subproceso adquiere el mutex.
- **my\_mutex\_trylock:** Se permite que esta función falle espuriamente y devuelva falso incluso si el mutex no está actualmente bloqueado por ningún otro subproceso.
- **my\_thread\_chsched:** Se encarga de cambiar el tipo scheduling del hilo.

Por otro lado se necesita definir las funciones de scheduler que se llegaron a aplicar, las cuales son las siguientes:

- **round\_robin:** Este scheduler se caracteriza por la espera circular que tiene, ya que tenemos una cola donde los procesos están ordenados por orden de llegada. En este caso se tiene que validar si la cantidad actual de los hilos esta vacía, para saber de ser lo contrario se determine un vector que contendrá los hilos de round robin.
- **lottery:** Es un algoritmo de programación probabilística para procesos en un sistema operativo. A cada proceso se le asigna una cierta cantidad de boletos de lotería, y el planificador extrae un boleto al azar para seleccionar el siguiente proceso. La distribución de boletos no necesita ser uniforme; otorgar a un proceso más boletos le brinda una probabilidad relativamente mayor de selección.
- **real\_time:** Este scheduler tiene una similitud con lo que respecta a lottey, ya que como lottery asigna aleatoriamente tickets a los hilos, para que estos mismos se ejecuten, en real time esta presente el factor de prioridad que va a tener cada proceso.

Como se pudo contemplar anteriormente se puede ver las funciones que se solicitaron en cuanto a la especificación del proyecto 1 del curso de sistemas operativos, ya

teniendo en cuenta lo anterior se procede a explicar funciones adicionales que se llegaron a necesitar:

- **signal\_handler:** Esta función se encargará de realizar un cambio en el contexto a los respectivos planificadores que puedan llegar a ejecutar un hilo, dicho contexto se extraerá con ayuda de `ucontext`.
- **set\_round\_robin\_contexts:** Esta función tendrá como entrada un vector que tiene todo el bloque de control de los hilos de round robin y adicionalmente una respectiva parada, en esta función se establece que a partir de un vector se pueda obtener un respectivo contexto de los hilos del planificador, para que de esta manera se pueda realizar un contexto, haciendo que pueda seguir avanzando con el siguiente proceso sucesivamente.
- **set\_lottery\_contexts:** Esta función tendrá como entrada un vector que tiene todo el bloque de control de los hilos de lottery y adicionalmente una respectiva parada, en esta función se establece que a partir de un vector se pueda obtener un respectivo contexto de los hilos del planificador, para que de esta manera se pueda realizar un contexto, haciendo que pueda seguir avanzando con el siguiente proceso sucesivamente.
- **set\_signal:** Esta función se encargará de manejar una señal para poder determinar si un proceso en el planificador se haya terminado, de manejar que no se reinicie y vuelva a inicializarse con un valor incorrecto de dirección de memoria.
- **scheduler:** Es la función encargada de almacenar en un vector la información del hilo de manera que este mismo vector pueda ser tratado con un respectivo planificador y que este mismo pueda sumarse a su vector principal de almacenamiento donde el planificador tendrá todos los hilos respectivos.
- **scheduler\_context:** Determina el contexto del planificador.
- **get\_actual\_round\_robin\_thread:** Obtiene el hilo actual del planificador de round robin.
- **get\_actual\_lottery\_thread:** Obtiene el hilo actual del planificador de lottery.
- **stack:** Es la capacidad que tiene la pila en el proyecto.

Adicionalmente a otras funciones que no tienen que ver con `my_pthread` se pueden encontrar las siguientes:

- **determinate\_txt:** Se encargará de leer las posiciones y inicializar cada una de las posiciones que deberá tener el txt que pueda tener de entrada.
- **parser:** Se encargará de abrir el archivo y leer todas las líneas del mismo.
- **animator:** Se encargará de ejecutar la animación de manera que primeramente se va a inicializar un determinando canvas con el ancho y el alto que se ingrese en el lenguaje, para que de esta manera se puedan determinar los respectivos movimientos que pueda tener la figura.
- **figure\_1:** Contendrá los movimientos respectivos con las figuras rotadas previamente vistas en el txt, para de esta manera darle un movimiento característico.
- **figure\_2:** Contendrá los movimientos respectivos con las figuras rotadas previamente vistas en el txt, para de esta manera darle un movimiento característico. Solamente que esta función contiene una figura diferente.

---

## Instrucciones para ejecutar el programa

1. Primeramente ingrese a la carpeta del proyecto
2. Ingrese el siguiente comando para la ejecución del programa animar:

```
./animar -c config.txt
```

## Actividades realizadas por estudiante

### ALBERTO ZUMBADO ABARCA

| Fecha    | Inicio   | Fin      | Avance Realizado                                  |
|----------|----------|----------|---|
| 21/09/22 | 7:00 pm  | 9:00 pm  | Lectura Proceso e Hilos                           |
| 24/09/22 | 10:00 am | 3:00 pm  | Investigación ucontext e implementación           |
| 25/09/22 | 7:00 pm  | 10:00 pm | Implementación algoritmos scheduling              |
| 28/09/22 | 9:00 am  | 12:00 pm | Investigación e implementación de signals y timer |
| 20/10/22 | 6:00 pm  | 10:00 pm | Programar el signal handler                       |
| 20/10/22 | 5:00 pm  | 9:00 pm  | Implementar mutex de manejar funcional            |
| 28/10/22 | 7:00 pm  | 1:00 am  | Aplicación mutex, documetación externa e interna  |

### JONATHAN QUESADA SALAS

| Fecha      | Inicio  | Fin      | Avance Realizado                                 |
|------------|---------|----------|--|
| 20/09/2022 | 7:00 pm | 10:00 pm | Leer otra vez el capítulo de Procesos E Hilos    |
| 28/09/2022 | 7:00 pm | 11:00 pm | Avance inicial del my_pthread                    |
| 03/10/2022 | 3:00 pm | 7:00 pm  | Avance de los planificadores                     |
| 08/10/2022 | 7:00 pm | 10:00 pm | Avance de canvas                                 |
| 14/10/2022 | 7:00 pm | 10:00 pm | Establecer lenguaje y avance en el parser        |
| 23/10/2022 | 7:00 pm | 10:00 pm | Animación con ncurses                            |
| 27/10/2022 | 8:00 am | 11:00 am | Animación en X Y con canvas                      |
| 28/10/22   | 7:00 pm | 1:00 am  | Aplicación mutex, documetación externa e interna |

## Autoevaluación

## Estado final

Con lo que respecta al estado final de este proyecto, se pueden tomar los siguientes principales puntos como incompletos:

1. Monitores de despliegue
2. Restricciones de tiempo de inicio y final
3. Límite de espacio de Objetos

No se pudo completar satisfactoriamente el inciso de la rúbrica de Multidisplay por temas de conocimientos y principalmente por tiempo.

## Problemas y limitaciones

Con lo que respecta a los problemas y limitaciones que se presentaron en el proyecto fueron las siguientes:

1. Primeramente un vacío de conocimiento con lo que respecta a las implementaciones de `my_pthread` ya que si bien es cierto que dicha biblioteca se usó en la tarea 3, no se llegó a investigar de más dicha biblioteca.
2. Por otra parte en la implementación de `my_pthread`, específicamente con lo que respecta al `signal` para que este mismo pueda cambiar de contexto sin que la dirección de memoria sea cambiada ya que cuando se almacenaba el contexto de un hilo y se le hacía un cambio de contexto para que este mismo pueda ser recorrido por un determinado planificador se cambiaba la dirección de donde fue establecido previamente.
3. En cuanto respecta a la implementación de la animación al inicio, se podía solo mover en el eje `x` o en el `y`, sin embargo no se podía implementar en conjunto, por lo cual se procedió a realizar los pasos respectivos en caso que una figura se mueva a la izquierda, derecha, arriba o abajo, por otra parte las rotaciones se realizan de manera a la dirección asignada, para esto mismo dichas figuras rotadas deben de estar en el archivo de extensión `.txt` para su respectivo manejo. Entonces como conclusión se contempló la limitación de hacer que genuinamente la figura en `ascii` llegara a rotar en un momento determinado de la animación
4. Establecer el tiempo de ejecución de la figura, ya que por motivos de tiempo se procuró realizar lo esencial de la animación
5. En el apartado del proyecto llamado Multidisplay se presentó una limitación marcada sobre el conocimiento que este rubro implica, ya que este mismo se tiene una noción de como realizarlo, sin embargo una implementación puntual no pudo ser desarrollada de manera satisfactoria.

## Reporte de commits

```
commit 3e9fb13a97921b10926b5f41964fd444a0363ee0 (HEAD -> mypthread, origin/mypthread)
Author: Nuwidra <nuwidra@gmail.com>
Date:   Fri Oct 28 23:51:57 2022 -0600
```

[+] Documentación Interna

```
commit ba856739365cbf3835db784902548c345390c403 (HEAD -> mypthread, origin/mypthread)
Author: Nuwidra <nuwidra@gmail.com>
Date:   Fri Oct 28 23:16:15 2022 -0600
```

[+] Últimas modificaciones para animator e ingreso de comando en terminal

commit 0d7a56b0b29541eb43dd68de3dddb33e3dbc9ce2

Author: Nuwidra <nuwidra@gmail.com>

Date: Fri Oct 28 22:06:02 2022 -0600

[+] Parte de la documentación interna

commit 42a1552ac7e48944af5efe2b63119a1855e41cd9

Author: Alberto Zumbado <zumbado.abarca.alberto@gmail.com>

Date: Fri Oct 28 11:19:11 2022 -0600

[+] animation wiht two figures

commit df0b787bbc05ad8d3ea1dbd0d62eec4e81bd0137

Author: Alberto Zumbado <zumbado.abarca.alberto@gmail.com>

Date: Thu Oct 27 21:26:53 2022 -0600

[+] mutex

commit 00288e0e57a016b44edf727dbb9631ea7a558479

Author: Alberto Zumbado <zumbado.abarca.alberto@gmail.com>

Date: Mon Oct 24 22:57:20 2022 -0600

[+] threads con varios hilos

commit 96e1bd04c70f358463e19abd61fb24a2eb65ca9c

Author: Nuwidra <nuwidra@gmail.com>

Date: Sun Oct 23 10:41:30 2022 -0600

[+] Animacion funcional (faltan retoques)

commit 0ee128b350d7fa71f5babe869abead6527306d5a

Author: Nuwidra <nuwidra@gmail.com>

Date: Fri Oct 14 19:34:53 2022 -0600

[+] Lenguaje para animar (txt) y avance en el parser

commit 6a532780e1821a0038f7da5beef7031beb93b48b

Author: Nuwidra <nuwidra@gmail.com>

Date: Sun Oct 9 11:53:25 2022 -0600

Todo ordenadito

commit 6ed9eba3e7d290df70e09afaba435e4cceb190d4

Author: Jonathan Quesada Salas <70598288+Nuwidra@users.noreply.github.com>

Date: Sun Oct 9 11:25:42 2022 -0600

Crear el canvas

commit 8a40dcc4d2b293cc2b7abb8d4358a31039dca795

Author: Nuwidra <nuwidra@gmail.com>

Date: Sun Oct 9 11:15:18 2022 -0600

#### Crear el canvas

commit a7b90cc31a42f0905a8c5c7506939c1ced337071  
Author: Nuwidra <nuwidra@gmail.com>  
Date: Sat Oct 8 16:48:17 2022 -0600

#### Estructura para el canva

commit 967e22f6fdb92376385f500829dbffee5fc65030  
Author: Nuwidra <<https://github.com/Nuwidra/IC-2101-P00-2020-ii->>  
Date: Tue Oct 4 20:04:59 2022 -0600

#### Detalle en scheduler\_change

commit cc000660a5599ed210fefa840cd9c163439f97eb  
Author: Nuwidra <<https://github.com/Nuwidra/IC-2101-P00-2020-ii->>  
Date: Tue Oct 4 19:40:50 2022 -0600

#### Avance en scheduler\_change

commit 29e4c49b590a58bb9499b8e1c038b6709f5fd5a3  
Merge: befdcab cdfdc16  
Author: Nuwidra <<https://github.com/Nuwidra/IC-2101-P00-2020-ii->>  
Date: Tue Oct 4 19:39:49 2022 -0600

Merge branch 'master' of [https://github.com/Nuwidra/IC-6600\\_Multi-Display-Animator\\_Proyecto-1](https://github.com/Nuwidra/IC-6600_Multi-Display-Animator_Proyecto-1)

commit befdcab3c08b483027ec85878991622a6e657ca  
Author: Nuwidra <<https://github.com/Nuwidra/IC-2101-P00-2020-ii->>  
Date: Tue Oct 4 19:38:28 2022 -0600

#### Avance en scheduler\_change

commit cdfdc1676a6534f604218f99fd04901d37df1335  
Author: Jonathan Quesada Salas <70598288+Nuwidra@users.noreply.github.com>  
Date: Mon Oct 3 12:16:31 2022 -0600

#### Update schedulers.rs

commit 9b7afebd55bdc43c55f195e8c6e450f5542565a6  
Author: Nuwidra <<https://github.com/Nuwidra/IC-2101-P00-2020-ii->>  
Date: Mon Oct 3 12:14:56 2022 -0600

#### Sorteo e intento de my\_thread\_chsched

commit 1e713dbfc37078f6137a9d2d5df3452f6ef2fefc  
Author: Nuwidra <<https://github.com/Nuwidra/IC-2101-P00-2020-ii->>  
Date: Mon Oct 3 10:33:43 2022 -0600

#### Avance en Round Robin y Real Time



```
commit 70457b0635603d8c986dbb8ad1028bdc3ec740af
Author: Nuwidra <https://github.com/Nuwidra/IC-2101-P00-2020-ii->
Date: Thu Sep 29 19:24:04 2022 -0600
```

Avance de mypthread

```
commit 2bc59100a440fceb8164a9c0e3f115f47df4f662
Author: Nuwidra <https://github.com/Nuwidra/IC-2101-P00-2020-ii->
Date: Wed Sep 28 21:43:25 2022 -0600
```

Avance inicial de mypthread  
(END)

### Calificación

| Rubro   | Porcentaje |
|---|------------|
| Scheduler RoundRobin                          | 5%         |
| Scheduler Sorteo                              | 5%         |
| Scheduler en Tiempo Real                      | 5%         |
| Cambio de Scheduler                           | 5%         |
| Funciones de la biblioteca pthreads           | 7%         |
| Documentación utilizando Markdown o Latex-PDF | 20%        |
| Diseño de lenguaje                            | 10%        |
| Implementación de la animación                | 15%        |
| Funcionamiento en Multiples Displays          | 0%         |
| Extra   | 0%         |
| Kick-off                                      | 5%         |
| TOTAL   | 77%        |

### Autoevaluación

#### Auto-Evaluación de Jonathan

| Rubro                             | Escala |
|-----------------------------------|--------|
| Aprendizaje de Round Robin        | 5      |
| Aprendizaje de Tiempo Real        | 2      |
| Aprendizaje de Cambio de contexto | 3      |
| Aprendizaje de Sorteo             | 5      |

#### Auto-Evaluación de Alberto

| Rubro | Escala |
|-------|--------|
|-------|--------|

|                                   |   |
|-----------------------------------|---|
| Aprendizaje de Round Robin        | 5 |
| Aprendizaje de Tiempo Real        | 2 |
| Aprendizaje de Cambio de contexto | 5 |
| Aprendizaje de Sorteo             | 5 |

---

## Lecciones Aprendidas

1. Primeramente inicie desde una vez con el proyecto a investigar apenas que lo asignan, ya que este proyecto requiere mucho tiempo de investigación y criterio para el desarrollo de los hilos y todo lo que ello implique, ya que estos mismos se deben de tener un conocimiento claro de su concepto y funcionamiento.
2. Se debe de realizar una investigación sobre posibles implementaciones de los métodos que se deben de utilizar, estudiar código en la red es fundamental para poder avanzar con el proyecto, pero es necesario entenderlo hasta la última línea.
3. Tenga en algún apartado de su computadora una sección para poder almacenar todos los vídeos tutoriales, sitios sobre aplicaciones sobre funcionalidades útiles, sitios webs en general, esto ya que es necesario para poder tener un respectiva bitácora para investigación e ir progresivamente avanzando.
4. Para el tema de la animación no se preocupe, ya que existe gracias a Dios ncurses es una biblioteca que es muy útil para las animaciones y la definición del canvas (perímetro), entonces primeramente enfoque toda su atención a `my_pthread` y todo lo que este mismo implica.
5. Con lo que respecta a `my_pthread` su mayor atención debe de estar en las funciones principales de `my_pthread` y los planificadores y el cambio del mismo ya que requieren de un mayor tiempo de inversión por parte del equipo para que estos funcionen, básicamente los mutexs son más fáciles de implementar porque se pueden ver como banderas, o bien se puede implementar código en asm en rust, ya que hay una biblioteca que lo hace, esto mismo porque en el libro que se ve en el curso están las respectivas implementaciones de dichas funciones.

---

## Bibliografía

- [1] Q. Lyles-Woods. "Terminal Sunday- 3". Medium.  
<https://quindarius.medium.com/terminal-sunday-3-78f92396e8e2>
- [2] "Inline assembly - Rust By Example". Learn Rust - Rust Programming Language.  
<https://doc.rust-lang.org/rust-by-example/unsafe/asm.html>
- [3] "Spin Lock using xchg". Stack Overflow.  
<https://stackoverflow.com/questions/28968552/spin-lock-using-xchg>
- [4] M. Gattozzi. "Global Uninitialized Statics in Rust". Rage Against the State Machine. <https://blog.mgattozzi.dev/global-uninitialized/>
- [5] "pthreads: src/mutex.c File Reference - doxygen documentation | Fossies Dox". Fossies - The Fresh Open Source Software Archive. [https://fossies.org/dox/pthreads-3.14/mutex\\_8c.html#a12809c44718c21544ceb280dc1a53573](https://fossies.org/dox/pthreads-3.14/mutex_8c.html#a12809c44718c21544ceb280dc1a53573)
- [6] "nix::ucontext::UContext - Rust". Docs.rs.  
<https://docs.rs/nix/0.15.0/nix/ucontext/struct.UContext.html>

[7] Página Principal.

[https://www.um.es/earlyadopters/actividades/a3/PCD\\_Activity3\\_Session1.pdf](https://www.um.es/earlyadopters/actividades/a3/PCD_Activity3_Session1.pdf)

[8] "Function Index (The GNU C Library)". The GNU Operating System and the Free Software Movement. [https://www.gnu.org/software/libc/manual/html\\_node/Function-Index.html](https://www.gnu.org/software/libc/manual/html_node/Function-Index.html)

[9] "Grafiati: Generador automático de citas online". Grafiati: Оформити списки використаних джерел онлайн. <https://www.grafiati.com/es/>

[10] Casual Coder. Ncurses Tutorial 0 - Hello World (initscr, endwin, refresh, getch, printw). (11 de diciembre de 2016). Accedido el 28 de octubre de 2022. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=1V-0PQhPvSM>

[11] Casual Coder. Ncurses Tutorial 1 - Moving Cursors (move, mvprintw, clear). (11 de diciembre de 2016). Accedido el 28 de octubre de 2022. [Video en línea]. Disponible: [https://www.youtube.com/watch?v=A5lX1h\\_2zy0](https://www.youtube.com/watch?v=A5lX1h_2zy0)

[12] Casual Coder. Ncurses Tutorial 2 - Basics of Windows (WINDOW\*, newwin, box, refresh, wrefresh, wprintw mvprintw). (11 de diciembre de 2016). Accedido el 28 de octubre de 2022. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=pjT5wg11ZSE>

[13] Casual Coder. Ncurses Tutorial 14.1 - Creating a Menubar (part 1). (27 de noviembre de 2020). Accedido el 28 de octubre de 2022. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=g7Woz3YVgvQ>

[14] Quin'darius Lyles-Woods. Animation in ncurses: x-axis. (25 de abril de 2021). Accedido el 28 de octubre de 2022. [Video en línea]. Disponible: <https://www.youtube.com/watch?v=Ms1l70scQac>

[15] "Working with c\_void in an FFI". Stack Overflow. <https://stackoverflow.com/questions/24191249/working-with-c-void-in-an-ffi>

[16] "Is there ever a good reason to have a function parameter be a void pointer in C or C++?" Quora. <https://www.quora.com/Is-there-ever-a-good-reason-to-have-a-function-parameter-be-a-void-pointer-in-C-or-C>

[17] "Function pointers - Unsafe Code Guidelines Reference". Page not found · GitHub Pages. <https://rust-lang.github.io/unsafe-code-guidelines/layout/function-pointers.html>

[18] M. Gattozzi. "Global Uninitialized Statics in Rust". Rage Against the State Machine. <https://blog.mgattozzi.dev/global-uninitialized/>

[19] "nix::sys::signal - Rust". Docs.rs. <https://docs.rs/nix/latest/nix/sys/signal/index.html>

[20] "sigevent(7) - Linux manual page". Michael Kerrisk - man7.org. <https://man7.org/linux/man-pages/man7/sigevent.7.html>