

Sway Recorder

Andres Felipe Gamba Algarra
Facultad de Ingeniería
Universidad Nacional de Colombia
Bogotá D.C., Colombia
angambaa@unal.edu.co

Abstract—Se logra crear Sway Recorder, una interfaz gráfica para el comando wf-recorder con el fin de facilitar y simplificar las capturas de video de pantalla en sistemas basados en wl-roots. A pesar de lograr un buen resultado gracias a la librería GTK, se recalca la dificultad de un primer acercamiento a este toolkit debido a la falta de recursos de aprendizaje.

Index Terms—GTK, wf-recorder, interfaz gráfica, styling, insert

I. INTRODUCCIÓN

Dentro del vasto mundo de GNU/Linux es posible encontrar alternativas para casi cada software, incluso hasta llegar al punto de encontrar distribuciones sin GNU Core Utilities, véase el caso de Alpine Linux [1]. Esto se sigue también dentro del terreno de los display servers, donde X Window System ha sido el estándar hace décadas [2] pero que poco a poco viene decayendo debido a la aparición y el buen desarrollo de Wayland, que en verdad viene a ser un display server protocol en vez de un display server. Se espera que X Window System sea eventualmente remplazado por Wayland [3], y esto nos lo empieza a demostrar aquellas distribuciones de Linux que ya lo han incorporado por defecto, algunos ejemplos son: Ubuntu, Manjaro, Debian y Fedora [4].

Si bien, la implementación de Wayland implica la ruptura con el paradigma de X Window System, no quiere decir que se comience desde cero, pues por lo general Wayland tiene compatibilidad con aplicaciones que corren nativamente en X, y si no, suele haber alguna alternativa similar. En otras palabras, lo que estábamos acostumbrados a usar en X, lo podemos seguir usando en Wayland bien sea bajo mantenimiento del mismo desarrollador, o de algún otro que haya decidido hacer un fork o una creación propia, y por tanto, esta transición entre display servers no debería repercutir importantemente en términos de usabilidad.

En el caso de los Windows Managers está Sway, que es un clon de I3 (nativo de X) con algunas adiciones, y según lo que he visto es el más popular si nos basamos en la cantidad de estrellas que tiene en Github. Sway ha desarrollado el protocolo Wlroots que como ellos describen, es un "proyecto que provee una base modular para Sway y otros compositors de Wayland" [5]. Así lo han dicho ellos y así ha sucedido, otros Windows Managers reconocidos de X han sido inspiración para nuevos Windows Managers en Wayland y han terminado basándose en Wlroots, entre ellos están: dwl y wayfire [6].

II. PROBLEMA

Luego de varios meses de usar Sway como Window Manager principal en mi computador, me encontré con un problema al cual no le vi solución en su momento. Según comenté anteriormente, Sway es el Window Manager más popular pero sigue siendo algo de nicho si se le compara con los Desktop Enviroments como Gnome [7]. Es por esto que aún tiene muchos aspectos a mejorar, por ejemplo la compatibilidad con GPU's Nvidia, o como trataré en este artículo: la ausencia de una aplicación sencilla y práctica para tomar capturas de video de la pantalla.

Solo encontré dos programas que podían cumplir esta funcionalidad en Sway. El primero, el reconocido programa OBS junto con el plugin wlrobs, y el segundo, wf-recorder [8]. Mi necesidad era poder grabar lo que ocurría en mi computador de manera fácil, rápida y espontánea, algo así como cuando necesito un screenshot, solo oprimo un keybinding y ya tengo lo que necesito. OBS parecía ser muy complejo y a veces demorado de configurar (desde la perspectiva de alguien que sabe casi nada sobre edición de videos). Por otro lado, wf-recorder parecía ser perfectamente simple y liviano pero menos intuitivo que OBS.

Wf-recorder no es más que un comando para correr desde la terminal, que con varios argumentos puede dar un excelente resultado. No obstante, al ser un programa para ejecutar desde la terminal, conlleva a que haya errores de digitación, a que toque escribir mucho para ejecutar algo relativamente simple y a que el usuario olvide cuales son los argumentos y parámetros que el programa tiene; en últimas, wf-recorder no es práctico para usar. Este es un ejemplo de lo que habría que escribir para hacer una grabación con audio (por defecto graba sin audio) en la carpeta "HOME/Videos":

```
wf-recorder --audio=easyeffects_sink.monitor  
-f ~/Videos/migrabacion.mp4 --output=DP-1  
--bframes=4
```

Ante esta falencia vi la oportunidad de hacer una interfaz gráfica que fuera un wrapper de wf-recorder, igualmente muy liviana, simple y además intuitiva y fácil de usar.

III. OBJETIVOS

Decidí llamar al proyecto "Sway recorder" aunque debería ser capaz de funcionar en cualquier Window Manager basado en Wlroots ya que wf-recorder lo es, sin embargo, tendría

mejor compatibilidad con Sway ya que es donde lo probé. Entre sus funciones básicas debería:

- Detectar automáticamente las fuentes de audio del sistema y autoseleccionar la fuente de salida predeterminada.
- Tener una casilla donde se pueda ingresar el nombre de la grabación cuando sea exportada.
- En caso de que no fuera ingresado ningún nombre, exportar la grabación bajo el nombre "FECHA HORA", siendo FECHA y HORA el momento en el que se inició la grabación.
- Detectar automáticamente las pantallas del sistema y autoseleccionar la pantalla predeterminada.
- Permitir al usuario escoger la carpeta destino de exportación por medio de un explorador de archivos.
- En caso de que no fuera seleccionado una carpeta destino, exportar la grabación en la carpeta "HOME/Videos".
- Permitir al usuario escoger si grabar la pantalla completa o solo un fragmento de ella.
- Tener un botón que sirva para activar o detener la grabación.
- Permitir al usuario determinar un tiempo de rezago desde que se oprime el botón de grabar hasta que realmente se empieza a grabar la pantalla.
- Integrarse con los "temas de color" Gtk del ordenador.

IV. MARCO TEÓRICO

Como se estableció en los objetivos, la aplicación debería integrarse con los colores Gtk del ordenador. La mayoría de Desktop Enviroments y Window Managers permiten escoger una combinación de colores que se van a integrar con las aplicaciones desarrolladas con Gtk (GIMP Toolkit), a esta combinación de colores se le conoce como "Color theme", en inglés. Considero una bondad desarrollar una interfaz gráfica con GTK ya que ayuda a mantener la armonía de colores entre las distintas aplicaciones del sistema. Entonces, al usar GTK no es necesario declarar el color que se va a usar en cada widget ya que automáticamente se va a ajustar al tema de colores predeterminado del sistema [9].

Adicionalmente, GTK es un toolkit que funciona en cualquier sistema operativo, por lo cual, si alguien quisiera hacer una aplicación para grabar videos en MacOS o Microsoft Windows, podría fácilmente tomar mi código de la interfaz gráfica y hacer las modificaciones necesarias para que el resto del programa ande correctamente.

La aproximación a este toolkit fue mediante objetos, que es la manera más intuitiva y fácil de organizar el código. Cada objeto corresponde a un elemento gráfico (widget), que puede invocar una función si la interacción del usuario lo amerita. Para acceder a GTK se hace desde la librería GObject (import gi). Otras librerías usadas en menor medida fueron:

- subprocess: permite ejecutar procesos externos a python dentro del sistema
- os: permite ejecutar procesos externos a python y es capaz de obtener "environmental variables" del shell que use el sistema operativo.

- time: se usó para determinar un tiempo t de retraso desde el llamado de una función hasta que fue ejecutada.
- json: se usó para convertir datos en formato .json en un dictado de python.

V. METODOLOGÍA

Hubo un reto considerable al intentar comprender cómo usar la librería de GTK. Fue imprescindible una preparación adicional sobre la temática de objetos, particularmente en herencias. La manera como funciona GTK es primero generando un objeto llamado "window" que es el contenedor principal, luego sobre este se añaden otros contenedores invisibles cuya función es crear patrones de organización sobre los cuales se añaden otros objetos que corresponden a los widgets. Los widgets son visibles, pueden ser legibles y el usuario puede interactuar con ellos, por ejemplo están los botones, comboboxes, checkboxes, entre otros. Entonces el flujo del desarrollo del código fue primero crear la ventana; segundo, crear una cuadrícula invisible de referencia; tercero, asignar cada widget a una coordenada de la cuadrícula; cuarto, asignar funciones a cada widget que retornaran un string correspondiente a un argumento del comando "wf-recorder"; último, configurar el botón "Start recording" para que inicie o detenga el comando 'wf-recorder' con los argumentos determinados por los otros widgets, de esta manera, es posible iniciar y detener la grabación solo con un par de clicks. En el código se puede ver con más detalle el funcionamiento de cada objeto tanto en la organización gráfica, así como en el funcionamiento del proceso principal de grabar la pantalla.

Fue un poco difícil obtener información concisa sobre como usar el toolkit desde mi perspectiva de principiante en la programación. Gtk es una librería que no esta hecha para un lenguaje en específico. Puede ser aprovechada de igual manera con C++, javascript, java, etc., por lo que, la documentación oficial está escrita en términos genéricos y las pocas referencias a ejemplos de código están en C++, no obstante, fue un recurso principal en el proceso de programación pero no el más deseable. Dado este panorama, mi mejor recurso para aprender a usar GTK fue google, el subreddit oficial de Gtk y otros proyectos open source que implementaran los mismos widgets que yo quería. En verdad me gastó más tiempo aprender qué widgets y métodos usar y cómo hacerlo a comparación de lo que gasté ideando y organizando los algoritmos del programa.

VI. RESULTADOS

El programa cumplió el objetivo principal: facilitar las capturas de video de pantalla en Sway y otros Windows Managers basados en Wlroots. Es capaz de asignar argumentos predeterminados precisos y eficaces al comando "wf-recorder" y desplegar distintas opciones de argumentos igual de precisas. Es muy ligero, sencillo y se entiende su uso con tan solo un vistazo.

Por otro lado, hay un bug que llama la atención y que sigue pendiente para ser solucionado. El argumento "-g" de wf-recorder requiere un string entre comillas, por ejemplo, "300,300 700x300", que indica el área del segmento de

la pantalla que se va a grabar. Entonces, al pasarse por `subprocess.Popen` para iniciar la grabación, la función queda de la siguiente manera: `"subprocess.Popen(...,-g "300,300 700x300")"`. Parece ser que `subprocess.Popen` es incapaz de procesar strings con comillas dentro, resultando en que no se ejecute dicho argumento y que el programa no pueda grabar un segmento de la pantalla, sino la pantalla completa solamente.

VII. CONCLUSIONES

Si no se encuentra un software o alguna alternativa de este en GNU/Linux, inevitablemente se hará, o mejor aun, lo podemos hacer. A veces simplemente, es tomar lo que ya está hecho y expresarlo en otros términos, algo tan sencillo puede mejorar la experiencia del usuario final.

La implementación exitosa de la mayoría de los objetivos de este proyecto deja un camino abierto a añadir más funcionalidades, pero tal vez ubicadas en un menú de configuración desplegable, de tal manera que no se sature la ventana principal y que el programa mantenga su esencia simple. Asimismo, una vez se consideren retroalimentaciones sobre el proyecto, el código de Sway Recorder sea mejor y se supere el bug actual, será un paso a seguir empaquetarlo para ser agregado a algún repositorio de software.

Respecto a los aspectos metodológicos, la falta de recursos de aprendizaje para el desarrollo de apps GTK con python inspira a crear una guía para principiantes, o al menos un cheatsheet.

REFERENCES

- [1] "Small. simple. secure.," about — Alpine Linux. [Online]. Available: <https://www.alpinelinux.org/about/>. [Accessed: 21-Jun-2022].
- [2] "X window system?," Habr, 27-Feb-2017. [Online]. Available: <https://habr.com/en/post/321470/>. [Accessed: 21-Jun-2022].
- [3] Wayland to replace the X Window System. [Online]. Available: <https://weekly-geekly.imtqy.com/articles/322580/index.html>. [Accessed: 21-Jun-2022].
- [4] M. Ashar, "Distros which adopted Wayland in 2021," Ubuntu Buzz ! : Unofficial Ubuntu Blog. [Online]. Available: <https://www.ubuntubuzz.com/2021/10/distros-which-adopted-wayland-in-2021.html>. [Accessed: 21-Jun-2022].
- [5] Sway. [Online]. Available: <https://swaywm.org/>. [Accessed: 21-Jun-2022].
- [6] "Wayland," Wayland - ArchWiki. [Online]. Available: <https://wiki.archlinux.org/title/WaylandCompositors>. [Accessed: 21-Jun-2022].
- [7] "Distrowatch.com: Put the fun back into computing. use linux, BSD." [Online]. Available: <https://distrowatch.com/dwres.php?resource=popularity>. [Accessed: 22-Jun-2022].
- [8] Natpen, "Natpen/awesome-wayland: A curated list of Wayland Code and resources.," GitHub. [Online]. Available: <https://github.com/natpen/awesome-wayland>. [Accessed: 21-Jun-2022].
- [9] "GTK-3.0," GTK Documentation. [Online]. Available: <https://docs.gtk.org/gtk3/>. [Accessed: 21-Jun-2022].