

Práctica Extra:

Emulación de videojuegos retro con la Raspberry Pi

Fundamentos de Sistemas Embebidos

Autores:
Pablo Axel Escalona Gallegos
Leonardo José Campos Rodríguez
Rodrigo Iván Olvera Martínez

1. Objetivo

El alumno aprenderá a hacer las configuraciones necesarias para lograr emular videojuegos retro en la Raspberry Pi.

2. Introducción

La presente práctica resume los pasos a seguir para desplegar en pantalla videojuegos, utilizando el emulador Mednafen y haciendo algunas configuraciones podremos utilizar un control ya sea de Playstation o Xbox para lograr jugar sin la necesidad de un teclado.

2.1. Mednafen

Mednafen es un emulador de múltiples sistemas de videojuegos retro. Su nombre es una abreviatura de "My Emulator Doesn't Need A Frickin' Excellent Name" (Mi emulador no necesita un nombre excelente). Es un software de código abierto y gratuito que permite al usuario jugar y disfrutar de juegos de consolas retro.

Mednafen admite una amplia gama de sistemas, incluidos NES, SNES, Game Boy, Game Boy Advance, Sega Genesis, Sega Saturn, Atari Lynx y muchas otras plataformas populares. El emulador se enfoca en la precisión y la emulación de hardware, lo que significa que intenta replicar fielmente el funcionamiento de las consolas originales y sus juegos.

Al utilizar Mednafen, el usuario puede cargar imágenes de ROM o ISO de juegos en el emulador y jugarlos en su computadora. El software ofrece varias opciones de configuración, como gráficos, sonido y controles, que permiten a los jugadores personalizar su experiencia de juego según sus preferencias.

3. Material

Se asume que el alumno cuenta con una Raspberry Pi con sistema operativo Raspbian e intérprete de Python instalado. Se aconseja encarecidamente el uso de *git* como programa de control de versiones.

- Control de PS4 o Xbox
- Cable usb v8 o el necesario si el control es inalámbrico.

4. Instrucciones

1. Instalar Mednafen.
2. Configuración.
3. Programa.

4.1. Paso 1: Instalar Mednafen.

Para este punto el alumno ya conectó debidamente a la red de internet la Raspberry, a continuación en la consola de Raspberry Pi OS insertar el comando “sudo apt update”

```
axel@raspberrypi:/home$ sudo apt update
Obj:1 http://security.debian.org/debian-security bullseye-security InRelease
Obj:2 http://deb.debian.org/debian bullseye InRelease
Obj:3 http://deb.debian.org/debian bullseye-updates InRelease
Obj:4 http://archive.raspberrypi.org/debian bullseye InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
axel@raspberrypi:/home$
```

Figura 1: comando update

Posteriormente insertar el comando “sudo apt install mednafen”

```
axel@raspberrypi:/home$ sudo apt install mednafen
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mednafen ya está en su versión más reciente (1.26.1+dfsg-1).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libfuse2
Utilice «sudo apt autoremove» para eliminarlo.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
axel@raspberrypi:/home$
```

Figura 2: comando para instalar mednafen

Dar si a la instalación y por último reiniciar la raspberry con “reboot”.

4.2. Paso 2: Configuración de archivos.

Vamos a crear un servicio para que la Raspberry inicie con una imagen o video personalizado. Creamos un archivo en el directorio raíz con el comando “sudo nano reproducción_arranque.sh”. Y se escribe lo siguiente:

```
#!/bin/bash
fbset -T 1 -noverbose -a /home/Inicio/splash.png
```

```
GNU nano 5.4      reproduccion_arranque.sh
#!/bin/bash
fbi -T 1 -noverbose -a /home/Inicio/splash.png

[ El fichero «reproduccion_arranque.sh» no es de escritura ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich.^\\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```

Figura 3: creación del archivo para la ejecución de imagen personalizada

Guardamos el archivo. Prestar atención en la dirección o ruta del archivo, para posteriormente guardarlo.

Después habilitar y activar el servicio de inicio

“sudo systemctl enable reproduccion_arranque”

“sudo systemctl start reproduccion_arranque”

Reiniciar la Raspberry, para que los cambios puedan verse, al reiniciar debería mostrarse en pantalla la imagen de arranque personalizada.

4.3. Paso 3: Copiar las ROMs a una ruta.

Buscar ROMs de la web y copiarlas a la ruta que sea de preferencia, en este caso “/home/rms/”, puede ser la que sea de conveniencia.

```
axel@raspberrypi:/home/rms$ ls
CAPAME.nes
'Chip '\''n Dale - Rescue Rangers 2 (USA).nes'
'Donkey Kong Classics (USA, Europe).nes'
'DuckTales (USA).nes'
'Flintstones 2 - The Surprise at Dinosaur Peak!, The (U).nes'
'Legend of Zelda, The (U) (PRG1) [!].nes'
'Metroid (U).nes'
'Ninja Gaiden (USA).nes'
SF3.nes
'Shadow Warriors 2 (E) [!].nes'
'Simpsons, The - Bartman Meets Radioactive Man (USA).nes'
SMB3.nes
SMW.nes
splash.png
'Super Mario Advance 2 - Super Mario World (USA, Australia).gba'
'Super Mario Kart (USA).sfc'
'Teenage Mutant Ninja Turtles (USA).nes'
'Terminator, The (USA, Europe).nes'
'Tiny Toon Adventures (USA).nes'
'Zelda 2 - The Adventure of Link (U).nes'
axel@raspberrypi:/home/rms$
```

Figura 5: ROMs copiadas a la ruta específica

4.4. Paso 4: Programa

El programa propuesto se guarda en la misma ruta o en la de conveniencia. Ejecutamos el comando “sudo nano inicio.py”

```

import os
import subprocess

def show_menu():
    print("Bienvenido a la interfaz ALR")
    print("1. Super Mario Bros 3")
    print("2. Super Mario World")
    print("3. Captan America")
    print("4. Chip 'n Dale - Rescue Rangers 2")
    print("5. Donkey Kong Classics")
    print("6. DuckTales")
    print("7. Flintstones 2")
    print("8. Ninja Gaiden")
    print("9. Shadow Warriors 2")
    print("10. Simpsons, The - Bartman Meets Radioactive Man")
    print("11. Teenage Mutant Ninja Turtles")
    print("12. Terminator")
    print("13. Tiny Toon Adventures")
    print("14. Legend of Zelda")
    print("15. Metroid")
    print("exit. APAGAR")
    print("reiniciar")

def play_rom(rom_path):
    print("Iniciando Mednafen...")
    subprocess.call(["mednafen", rom_path]) # Ejecutar Mednafen con la ruta de la ROM

def main():
    show_menu()

```

Figura 6: Programa

```

while True:
    choice = input("Ingrese su opción: ")

    if choice == "1":
        play_rom("/home/rms/SMB3.nes")
    elif choice == "2":
        play_rom("/home/rms/SMW.nes")
    elif choice == "3":
        play_rom("/home/rms/CAPAME.nes.nes")
    elif choice == "4":
        play_rom("/home/rms/Chip 'n Dale - Rescue Rangers 2 (USA).nes")
    elif choice == "5":
        play_rom("/home/rms/Donkey Kong Classics (USA, Europe).nes")
    elif choice == "6":
        play_rom("/home/rms/DuckTales (USA).nes")
    elif choice == "7":
        play_rom("/home/rms/Flintstones 2 - The Surprise at Dinosaur Peak!, The (U).nes")
    elif choice == "8":
        play_rom("/home/rms/Ninja Gaiden (USA).nes")
    elif choice == "9":
        play_rom("/home/rms/Shadow Warriors 2 (E) [!].nes")
    elif choice == "10":
        play_rom("/home/rms/Simpsons, The - Bartman Meets Radioactive Man (USA).nes")
    elif choice == "11":
        play_rom("/home/rms/Teenage Mutant Ninja Turtles (USA).nes")
    elif choice == "12":
        play_rom("/home/rms/Terminator, The (USA, Europe).nes")
    elif choice == "13":
        play_rom("/home/rms/Tiny Toon Adventures (USA).nes")
    elif choice == "14":
        play_rom("/home/rms/Legend of Zelda, The (U) (PRG1) [!].nes")
    elif choice == "15":
        play_rom("/home/rms/Metroid (U).nes")
    elif choice == "exit":
        print("Apagando...")
        os.system("sudo shutdown now") # Apagar la Raspberry Pi
        break
    elif choice == "reiniciar":
        print("Reiniciando...")
        os.system("sudo reboot") # Reiniciar la Raspberry Pi
        break
    else:
        print("Opción inválida. Intente nuevamente.")

```

Figura 7: Programa, continuación

Este programa se encarga de cargar las roms y con llamadas al sistema arrancamos mednafen automáticamente, verificar muy bien si las rutas y nombres de las roms son correctas, de lo contrario, no se ejecutará el juego.

4.5. Paso 5: Configuración de mando o control.

Al iniciar cualquier juego presionamos Ctrl+Shift+1 y seguimos las instrucciones en pantalla, se deben de presionar los botones mostrados para configurarlos y se guarden. Este paso solo se hace la primera vez, posteriormente ya no será necesario.

5. Experimentos

[5pt] Ejecute un juego y configure los botones para el control o mando que se haya elegido

[2pt] Modifique el programa para que al cerrar un juego se presente nuevamente todos los juegos disponibles.

[3pt] Agregue mínimo 5 ROMs más, se muestren en la interfaz y ejecute los juegos.

[+3pt] Haga las configuraciones para que al encender la raspberry se ejecute el programa sin necesidad de arrancarlo manualmente.

6. Cuestionario

¿Qué otras herramientas de emulación existen, para el propósito de esta práctica?

¿Podemos cargar juegos de playstation o gamecube con mednafen?

¿Se considera ilegal el reproducir copias de juegos de Nintendo de la Manera en la que se hacen en esta práctica?

7. Referencias

- [1] Foro insertmorecoins - Como se configura el mando en Mednafen? (s. f.). <https://insertmorecoins.es/foro/consolas/como-se-configura-el-mando-en-mednafen/> Ludmila Maceková. 1-wire-the technology for sensor networks. *Acta Electrotechnica et Informatica*, 12(4):52,2012. Last accessed 14/06/2023
- [2] *Mednafen General Documentation*. (s. f.). <https://mednafen.github.io/documentation/>. Last Accessed 13/06/2023
- [3] Emulators - Raspberry Pi Gaming - page 62. (s. f.). <http://what-when-how.com/Tutorial/topic-213lnr/Raspberry-Pi-Gaming-90.html> Last Accessed 11/06/2023
- [4] Ellingwood, J. (2020). Cómo usar Systemctl para gestionar servicios y unidades de Systemd. DigitalOcean. <https://www.digitalocean.com/community/tutorials/how-to-use-systemctl-to-manage-systemd-services-and-units-es>. Last Accessed 14/06/2023

8.Enlaces.

<https://youtu.be/2q4xXNHV9DE>

<https://github.com/Paxeles/PFFSEm2023-2>