#!/bin/bash

# Function to run a command and handle errors

run\_command() {

echo -e "\n$1"

eval $2

if [ $? -ne 0 ]; then

echo "Warning: The command encountered an error, but continuing with the script."

fi

}

# Function to pause before closing

pause() {

read -p "Press any key to continue or Ctrl+C to exit..."

}

# Step 1: Install Dependencies

install\_dependencies() {

echo "Step 1: Installing necessary dependencies..."

run\_command "sudo apt update" "Updating package lists..."

run\_command "sudo apt upgrade -y" "Upgrading system packages..."

run\_command "sudo apt install cmake gcc g++ git build-essential libpng-dev wget curl -y" "Installing build tools and dependencies..."

}

# Step 2: Install Box64

setup\_box64() {

echo "Step 2: Setting up Box64..."

run\_command "git clone https://github.com/ptitSeb/box64.git" "Cloning Box64 repository..."

run\_command "cd box64 && mkdir build && cd build && cmake .. -DCMAKE\_BUILD\_TYPE=RelWithDebInfo && make" "Building Box64 from source..."

run\_command "cd box64/build && sudo make install" "Installing Box64..."

run\_command "sudo ldconfig" "Refreshing library paths for Box64..."

}

# Step 3: Install Box86

setup\_box86() {

echo "Step 3: Setting up Box86..."

run\_command "git clone https://github.com/ptitSeb/box86.git" "Cloning Box86 repository..."

run\_command "cd box86 && mkdir build && cd build && cmake .. -DCMAKE\_BUILD\_TYPE=RelWithDebInfo && make" "Building Box86 from source..."

run\_command "cd box86/build && sudo make install" "Installing Box86..."

run\_command "sudo ldconfig" "Refreshing library paths for Box86..."

}

# Step 4: Install Wine (only if it's available for ARM)

setup\_wine() {

echo "Step 4: Installing Wine..."

# Check if Wine is available for ARM

if ! command -v wine > /dev/null; then

run\_command "sudo dpkg --add-architecture arm64" "Adding arm64 architecture for Wine..."

run\_command "sudo mkdir -pm755 /etc/apt/keyrings" "Creating keyrings directory..."

run\_command "sudo wget -O /etc/apt/keyrings/winehq-archive.key https://dl.winehq.org/wine-builds/winehq.key" "Downloading WineHQ key..."

run\_command "sudo wget -NP /etc/apt/sources.list.d/ https://dl.winehq.org/wine-builds/ubuntu/dists/noble/winehq-noble.sources" "Setting up Wine repository..."

run\_command "sudo apt update" "Updating package list after adding Wine repository..."

if ! run\_command "sudo apt install --install-recommends winehq-stable -y" "Installing Wine Stable version..."; then

echo "Warning: Wine installation failed. It might not be available for your architecture."

fi

# Attempt to fix broken dependencies if any

run\_command "sudo apt-get install -f" "Fixing broken dependencies..."

else

echo "Wine is already installed."

fi

}

# Step 5: Install DXVK

setup\_dxvk() {

echo "Step 5: Installing DXVK..."

run\_command "wget https://github.com/doitsujin/dxvk/releases/download/v2.3/dxvk-2.3.tar.gz" "Downloading DXVK package..."

run\_command "tar -xvf dxvk-2.3.tar.gz" "Extracting DXVK package..."

run\_command "cd dxvk-2.3 && sudo ./setup\_dxvk.sh install" "Installing DXVK..."

}

# Step 6: Install Flatpak if not installed

install\_flatpak() {

echo "Checking for Flatpak installation..."

if ! command -v flatpak > /dev/null; then

echo "Flatpak not found. Installing Flatpak..."

run\_command "sudo apt install flatpak -y" "Installing Flatpak..."

else

echo "Flatpak is already installed."

fi

}

# Step 7: Add Flathub repository for Steam

add\_flathub() {

echo "Adding Flathub repository for Steam..."

run\_command "flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo" "Adding Flathub repository..."

}

# Step 8: Install Steam via Flatpak

install\_steam\_flatpak() {

echo "Installing Steam via Flatpak..."

# Attempt to install Steam

if flatpak install flathub com.valvesoftware.Steam -y; then

echo "Steam installation completed successfully."

else

echo "Error: Steam installation failed. Checking for errors..."

# Log the error details

echo "Steam installation failed. Please check the log for details." | tee -a steam\_installation\_error.log

echo "========================" >> steam\_installation\_error.log

echo "Installation Date: $(date)" >> steam\_installation\_error.log

echo "Command Output:" >> steam\_installation\_error.log

flatpak install flathub com.valvesoftware.Steam -y 2>> steam\_installation\_error.log

# Provide instructions for the user

echo "For more information, please check the 'steam\_installation\_error.log' file."

fi

}

# Step 9: Run Steam

run\_steam() {

echo "Launching Steam..."

run\_command "flatpak run com.valvesoftware.Steam" "Running Steam..."

}

# Main function

main() {

echo "Starting installation process..."

pause # Pause before starting the installation

install\_dependencies

setup\_box64

setup\_box86

setup\_wine

setup\_dxvk

install\_flatpak

add\_flathub

install\_steam\_flatpak

run\_steam

echo "Installation complete!"

pause # Pause before closing

}

main