

Forum:[Hack Forums](#)

Thread: [Run Various Linux Distros without Rebooting - Directly in Terminal](#)

G-DillahKillah - September 7, 2025, 1:22 pm

For anyone that can't quite pick between distros or wants to try them without needing to burn an image to a usb and boot into them, I wrote something to use various distros right from your terminal.

Your options are...

- Kali
- Parrot OS
- Fedora
- Kodachi (will update once the revamped release hits the scene)

I'm attaching the 3 files that will run and their virus total scans here.

PLEASE NOTE: I just had to be fucking fancy and add background music so I'd recommend the Github route if you want the full effect.

Otherwise....well....here you go LMFAO

If you have any issues getting it to run for some reason, or recommendations for Distro's to add just let know.

TO RUN...

```
Code pipenv shell
git clone https://github.com/Thr0wAway-n0w/EMily.git
cd EMily
python3 -m pip install -r requirements.txt
python3 EMily_Main.py
```

```
Code import os
import subprocess
import threading
import tkinter as tk
from tkinter import ttk
import requests
from PIL import Image, ImageTk
from io import BytesIO
os.system("pip install pygame")
os.system("sudo apt install python3-pygame")
import pygame
subprocess.run(["sudo", "chmod", "+x", "EMily.py"])
```

```
pygame.mixer.init()
```

```
# List of background music files
```

```
playlist = ["8-bit-dungeon-251388.mp3", "glitch-in-the-dark-306765.mp3", "circuit-blitz-387798.mp3",
"circuit-pathway-387799.mp3"]
```

```

def play_music_playlist():
    while True:
        for song in playlist:
            try:
                pygame.mixer.music.load(song)
                pygame.mixer.music.play()
                # Wait until the song finishes
                while pygame.mixer.music.get_busy():
                    pygame.time.wait(100)
            except Exception as e:
                print(f"Error playing {song}: {e}")

# Start the playlist in a separate thread so it doesn't block the GUI
music_thread = threading.Thread(target=play_music_playlist, daemon=True)
music_thread.start()
# Use a thread so the damn thing doesn't stop the progress bar
def run_emily_choice(choice):
    def run_script():
        try:
            process = subprocess.Popen(['python3', 'EMily.py'], stdin=subprocess.PIPE,
            stdout=subprocess.PIPE, stderr=subprocess.PIPE)
            stdout, stderr = process.communicate(input=f'{choice}\n'.encode())
            print(stdout.decode())
            if stderr:
                print(stderr.decode())
        except Exception as e:
            print(f"Error running EMily.py with choice {choice}: {e}")
    finally:
        # Take a guess dumb-ass
        progress_bar.stop()

# Hmmmm....I wonder
progress_bar.start()

# I love threading, but I fucking hate it
threading.Thread(target=run_script).start()

def create_gui(root):
    global progress_bar
    # Remove any fidget widgets
    for widget in root.winfo_children():
        widget.destroy()

    # How's she lookin'
    style = ttk.Style(root)
    style.configure('TButton', font=('Arial', 20, 'bold'), foreground='red', background='black')

    # Gif URL. Don't bother with ascii, this works fine
    url = "https://media.tenor.com/QYjTsemTldcAAAAM/pixel-devil-devil.gif"
    try:
        response = requests.get(url)
        response.raise_for_status()
        image_data = response.content

```

```

    image = Image.open(BytesIO(image_data))
except requests.exceptions.RequestException as e:
    print(f"Error fetching image: {e}")
    return
except Exception as e:
    print(f"Error processing image: {e}")
    return

# Yea...animating is kinda a bitch
frames = []
try:
    for frame in range(0, image.n_frames):
        image.seek(frame)
        frame_image = image.copy()
        frame_image = frame_image.resize((440, 400), Image.LANCZOS)
        frames.append(ImageTk.PhotoImage(frame_image))
except Exception as e:
    print(f"Error gett'n zee frames: {e}")
    return

# Label for animation
image_label = tk.Label(root, borderwidth=2, relief='flat', bg='black', highlightcolor='red',
highlightbackground='red')
image_label.grid(row=0, column=0, columnspan=2, padx=10, pady=10)

def animate(counter=0):
    frame = frames[counter]
    image_label.configure(image=frame)
    counter = (counter + 1) % len(frames)
    root.after(100, animate, counter)

animate()

button_width = 18

# Buttons to run EMily.py
tk.Button(root, text="Kali-Linux", command=lambda: run_emily_choice(1)).grid(row=1,
column=0, padx=15, pady=15)
tk.Button(root, text="Parrot OS", command=lambda: run_emily_choice(2)).grid(row=1,
column=1, padx=15, pady=15)
tk.Button(root, text="Fedora", command=lambda: run_emily_choice(3)).grid(row=2, column=0,
padx=15, pady=15)
tk.Button(root, text="Kodachi", command=lambda: run_emily_choice(4)).grid(row=2, column=1,
padx=15, pady=15)

# I couldn't figure out a true progress bar...the best you get is this animated loader...tough shit
progress_bar = ttk.Progressbar(root, mode='indeterminate', length=200)
progress_bar.grid(row=3, column=0, columnspan=2, pady=20)

# window size
root.geometry('475x650')

# Window boarder

```

```

root = tk.Tk()
root.title("~~~~~ EMilly-The Linux EMulator ~~~~~")
root.configure(background='black')

# Create GUI
create_gui(root)

# Run main loop
root.mainloop()

Code import os
import subprocess
import shlex
import shutil
os.system("sudo apt install qemu-system-x86")
os.system("sudo apt install lolcat")
os.system("sudo apt install aria2c")
os.system("sudo apt install axel")
# Ensure lolcat is installed
subprocess.run(["sudo", "apt-get", "install", "-y", "lolcat"])

def clear_screen():
    os.system('clear' if os.name == 'posix' else 'cls')

def display_menu():
    print("Pick a Distro:")
    print("1) Kali")
    print("2) Parrot")
    print("3) Fedora")
    print("4) Kodachi (RUNS SLOW)")
    choice = input("Enter your choice (1-4): ")
    return choice.strip()

def display_label(label):
    cmd = f'echo "{label}" | lolcat -a -d 20'
    subprocess.run(cmd, shell=True)

def ensure_iso_dir():
    subprocess.run(["sudo", "mkdir", "-p", "/mnt/iso"])

def iso_exists(path):
    return os.path.isfile(path)

def download_iso(url, filename):
    if not iso_exists(filename):
        print(f"Downloading {filename}...")
        # Use aria2c for faster multi-threaded download if available
        if shutil.which("aria2c"):
            subprocess.run(["aria2c", "-x", "16", "-s", "16", "-o", filename, url])
        else:
            subprocess.run(["axel", filename, url])
        subprocess.run(["sudo", "mv", filename, "/mnt/iso"])
    else:

```

```

print(f"{filename} already exists. Skipping download.")

def run_qemu(iso_path, terminal_emulator):
    # Enable KVM for better performance if available
    qemu_cmd = f"qemu-system-x86_64 -boot d -cdrom {shlex.quote(iso_path)} -m 2048 -smp cores=2"
    if terminal_emulator == "gnome-terminal":
        command = [
            "gnome-terminal", "--",
            "bash", "-c",
            f"{qemu_cmd}; exec bash"
        ]
    elif terminal_emulator == "xterm":
        command = [
            "xterm", "-hold", "-e",
            f"{qemu_cmd}"
        ]
    elif terminal_emulator == "konsole":
        command = [
            "konsole", "-e",
            f"{qemu_cmd}"
        ]
    else:
        print("Unsupported terminal emulator. Please modify the script.")
        return
    subprocess.Popen(command)

def main():
    clear_screen()
    display_label(
        """
?-----?
|         |
| ??????++ ++ |
| ?? ???|| +++ |
| ??? ?+-+ + |
|         |
?-----?
        """)
    display_label("Linux Distro Emulator")
    ensure_iso_dir()
    choice = display_menu()

    iso_info = {
        "1": {
            "name": "Kali Linux",
            "url": "https://cdimage.kali.org/kali-2025.2/kali-linux-2025.2-live-amd64.iso",
            "filename": "kali-linux-2025.2-live-amd64.iso"
        },
        "2": {
            "name": "Parrot Security",
            "url": "https://deb.parrot.sh/parrot/iso/6.4/Parrot-security-6.4_amd64.iso",
            "filename": "Parrot-security-6.4_amd64.iso"
        }
    }

```

```

    },
    "3": {
        "name": "Fedora",
        "url":
"https://download.fedoraproject.org/pub/alt/releases/42/Labs/x86_64/iso/Fedora-Security-Live-x86_64-42-1.1.iso",
        "filename": "Fedora-Security-Live-x86_64-42-1.1.iso"
    },
    "4": {
        "name": "Kodachi",
        "url": "https://sourceforge.net/projects/linuxkodachi/files/latest/download",
        "filename": "kodachi-8.27-64-kernel-6.2.iso"
    }
}

```

```

if choice not in iso_info:
    print("Invalid choice.")
    return

```

```

label = iso_info[choice]["name"].upper()
clear_screen()
display_label(label)

```

```

iso_path = f"/mnt/iso/{iso_info[choice]['filename']}"
download_iso(iso_info[choice]['url'], iso_info[choice]['filename'])

```

```

# Use your preferred terminal emulator
terminal_emulator = "gnome-terminal" # or "xterm", "konsole"
run_qemu(iso_path, terminal_emulator)

```

```

if __name__ == "__main__":
    main()

```

Code requests
Pillow
pygame
shutil

??????????? - September 17, 2025, 3:10 am

(Sep 7, 2025 01:22 PM)G-DillahKillah Wrote: [\[CITE\]](#) For anyone that can't quite pick between distros or wants to try them without needing to burn an image to a usb and boot into them, I wrote something to use various distros right from your terminal.

Your options are...

- Kali
- Parrot OS
- Fedora
- Kodachi (will update once the revamped release hits the scene)

I'm attaching the 3 files that will run and their virus total scans here.

PLEASE NOTE: I just had to be fucking fancy and add background music so I'd recommend the

Github route if you want the full effect.
Otherwise....well....here you go LMFAO

If you have any issues getting it to run for some reason, or recommendations for Distro's to add just let know.

TO RUN...

```
Code pipenv shell
git clone https://github.com/Thr0wAway-n0w/EMily.git
cd EMily
python3 -m pip install -r requirements.txt
python3 EMily_Main.py
```

```
Code import os
import subprocess
import threading
import tkinter as tk
from tkinter import ttk
import requests
from PIL import Image, ImageTk
from io import BytesIO
os.system("pip install pygame")
os.system("sudo apt install python3-pygame")
import pygame
subprocess.run(["sudo", "chmod", "+x", "EMily.py"])
```

```
pygame.mixer.init()
```

[h1]List of background music files[/h1]

```
playlist = ["8-bit-dungeon-251388.mp3", "glitch-in-the-dark-306765.mp3", "circuit-blitz-387798.mp3",
"circuit-pathway-387799.mp3"]
```

```
def play_music_playlist():
    while True:
        for song in playlist:
            try:
                pygame.mixer.music.load(song)
                pygame.mixer.music.play()
                # Wait until the song finishes
                while pygame.mixer.music.get_busy():
                    pygame.time.wait(100)
            except Exception as e:
                print(f"Error playing {song}: {e}")
```

[h1]Start the playlist in a separate thread so it doesn't block the GUI[/h1]

```
music_thread = threading.Thread(target=play_music_playlist, daemon=True)
music_thread.start()
```

[h1]Use a thread so the damn thing doesn't stop the progress bar[/h1]

```
def run_emily_choice(choice):
    def run_script():
        try:
            process = subprocess.Popen(['python3', 'EMily.py'], stdin=subprocess.PIPE,
            stdout=subprocess.PIPE, stderr=subprocess.PIPE)
```

```

        stdout, stderr = process.communicate(input=f'{choice}\n'.encode())
        print(stdout.decode())
        if stderr:
            print(stderr.decode())
    except Exception as e:
        print(f"Error running EMily.py with choice {choice}: {e}")
    finally:
        # Take a guess dumb-ass
        progress_bar.stop()

# Hmmmm....I wonder
progress_bar.start()

# I love threading, but I fucking hate it
threading.Thread(target=run_script).start()

def create_gui(root):
    global progress_bar
    # Remove any fidget widgets
    for widget in root.winfo_children():
        widget.destroy()

    # How's she lookin'
    style = ttk.Style(root)
    style.configure('TButton', font=('Arial', 20, 'bold'), foreground='red', background='black')

    # Gif URL. Don't bother with ascii, this works fine
    url = "https://media.tenor.com/QYjTsemTldcAAAAM/pixel-devil-devil.gif"
    try:
        response = requests.get(url)
        response.raise_for_status()
        image_data = response.content
        image = Image.open(BytesIO(image_data))
    except requests.exceptions.RequestException as e:
        print(f"Error fetching image: {e}")
        return
    except Exception as e:
        print(f"Error processing image: {e}")
        return

    # Yea...animating is kinda a bitch
    frames = []
    try:
        for frame in range(0, image.n_frames):
            image.seek(frame)
            frame_image = image.copy()
            frame_image = frame_image.resize((440, 400), Image.LANCZOS)
            frames.append(ImageTk.PhotoImage(frame_image))
    except Exception as e:
        print(f"Error gett'n zee frames: {e}")
        return

    # Label for animation

```

```

image_label = tk.Label(root, borderwidth=2, relief='flat', bg='black', highlightcolor='red',
highlightbackground='red')
image_label.grid(row=0, column=0, columnspan=2, padx=10, pady=10)

def animate(counter=0):
    frame = frames[counter]
    image_label.configure(image=frame)
    counter = (counter + 1) % len(frames)
    root.after(100, animate, counter)

animate()

button_width = 18

# Buttons to run EMily.py
tk.Button(root, text="Kali-Linux", command=lambda: run_emily_choice(1)).grid(row=1,
column=0, padx=15, pady=15)
tk.Button(root, text="Parrot OS", command=lambda: run_emily_choice(2)).grid(row=1,
column=1, padx=15, pady=15)
tk.Button(root, text="Fedora", command=lambda: run_emily_choice(3)).grid(row=2, column=0,
padx=15, pady=15)
tk.Button(root, text="Kodachi", command=lambda: run_emily_choice(4)).grid(row=2, column=1,
padx=15, pady=15)

# I couldn't figure out a true progress bar...the best you get is this animated loader...tough shit
progress_bar = tk.Progressbar(root, mode='indeterminate', length=200)
progress_bar.grid(row=3, column=0, columnspan=2, pady=20)

# window size
root.geometry('475x650')

[h1]Window boarder[/h1]
root = tk.Tk()
root.title("~~~~~ EMilly-The Linux EMulator ~~~~~")
root.configure(background='black')

[h1]Create GUI[/h1]
create_gui(root)

[h1]Run main loop[/h1]
root.mainloop()

Code import os
import subprocess
import shlex
import shutil
os.system("sudo apt install qemu-system-x86")
os.system("sudo apt install lolcat")
os.system("sudo apt install aria2c")
os.system("sudo apt install axel")
[h1]Ensure lolcat is installed[/h1]
subprocess.run(["sudo", "apt-get", "install", "-y", "lolcat"])

```

```

def clear_screen():
    os.system('clear' if os.name == 'posix' else 'cls')

def display_menu():
    print("Pick a Distro:")
    print("1) Kali")
    print("2) Parrot")
    print("3) Fedora")
    print("4) Kodachi (RUNS SLOW)")
    choice = input("Enter your choice (1-4): ")
    return choice.strip()

def display_label(label):
    cmd = f'echo "{label}" | lolcat -a -d 20'
    subprocess.run(cmd, shell=True)

def ensure_iso_dir():
    subprocess.run(["sudo", "mkdir", "-p", "/mnt/iso"])

def iso_exists(path):
    return os.path.isfile(path)

def download_iso(url, filename):
    if not iso_exists(filename):
        print(f"Downloading {filename}...")
        # Use aria2c for faster multi-threaded download if available
        if shutil.which("aria2c"):
            subprocess.run(["aria2c", "-x", 16, "-s", 16, "-o", filename, url])
        else:
            subprocess.run(["axel", filename, url])
            subprocess.run(["sudo", "mv", filename, "/mnt/iso"])
    else:
        print(f"{filename} already exists. Skipping download.")

def run_qemu(iso_path, terminal_emulator):
    # Enable KVM for better performance if available
    qemu_cmd = f"qemu-system-x86_64 -boot d -cdrom {shlex.quote(iso_path)} -m 2048 -smp\ncores=2"
    if terminal_emulator == "gnome-terminal":
        command = [
            "gnome-terminal", "--",
            "bash", "-c",
            f"{qemu_cmd}; exec bash"
        ]
    elif terminal_emulator == "xterm":
        command = [
            "xterm", "-hold", "-e",
            f"{qemu_cmd}"
        ]
    elif terminal_emulator == "konsole":
        command = [
            "konsole", "-e",
            f"{qemu_cmd}"
        ]

```

```

    ]
else:
    print("Unsupported terminal emulator. Please modify the script.")
    return
subprocess.Popen(command)

def main():
    clear_screen()
    display_label(
"""
?-----?
|           |
| ??????++ ++ |
| ?? ???|| +++ |
| ??? ?++-+ + |
|           |
?-----?
""")
    display_label("Linux Distro Emulator")
    ensure_iso_dir()
    choice = display_menu()

    iso_info = {
        "1": {
            "name": "Kali Linux",
            "url": "https://cdimage.kali.org/kali-2025.2/kali-linux-2025.2-live-amd64.iso",
            "filename": "kali-linux-2025.2-live-amd64.iso"
        },
        "2": {
            "name": "Parrot Security",
            "url": "https://deb.parrot.sh/parrot/iso/6.4/Parrot-security-6.4_amd64.iso",
            "filename": "Parrot-security-6.4_amd64.iso"
        },
        "3": {
            "name": "Fedora",
            "url":
"https://download.fedoraproject.org/pub/alt/releases/42/Labs/x86_64/iso/Fedora-Security-Live-x86_64-42-1.1.iso",
            "filename": "Fedora-Security-Live-x86_64-42-1.1.iso"
        },
        "4": {
            "name": "Kodachi",
            "url": "https://sourceforge.net/projects/linuxkodachi/files/latest/download",
            "filename": "kodachi-8.27-64-kernel-6.2.iso"
        }
    }

    if choice not in iso_info:
        print("Invalid choice.")
        return

    label = iso_info[choice]["name"].upper()
    clear_screen()

```

```
display_label(label)
```

```
iso_path = f"/mnt/iso/{iso_info[choice]['filename']}"  
download_iso(iso_info[choice]['url'], iso_info[choice]['filename'])
```

```
# Use your preferred terminal emulator  
terminal_emulator = "gnome-terminal" # or "xterm", "konsole"  
run_qemu(iso_path, terminal_emulator)
```

```
if __name__ == "__main__":  
    main()
```

Code requests

Pillow

pygame

shutil

These days kali also have multiple variants.

G-DillahKillah - October 2, 2025, 7:45 pm

(Sep 17, 2025 03:10 AM)????????????? Wrote: [\[CITE\]](#) These days kali also have multiple variants.
Like Kali Purple correct?

????????????? - October 3, 2025, 4:08 am

(Oct 2, 2025 07:45 PM)G-DillahKillah Wrote: [\[CITE\]](#) Like Kali Purple correct?
Purple for defensive and blue for offensive.

G-DillahKillah - October 6, 2025, 11:14 am

(Oct 3, 2025 04:08 AM)????????????? Wrote: [\[CITE\]](#) Purple for defensive and blue for offensive.
Correct me if I'm wrong, but Kali Purple IS for both offensive and defensive purposes.
Red (Team) + Blue (Team) = Purple

????????????? - October 6, 2025, 11:23 am

(Oct 6, 2025 11:14 AM)G-DillahKillah Wrote: [\[CITE\]](#) Correct me if I'm wrong, but Kali Purple IS for
both offensive and defensive purposes.
Red (Team) + Blue (Team) = Purple
Kali Purple usually emphasizes defensive security tools, which acts as a "SOC-in-a-box" and
supports blue and purple team exercises.

G-DillahKillah - October 6, 2025, 11:43 am

(Oct 6, 2025 11:23 AM)????????????? Wrote: [\[CITE\]](#) Kali Purple usually emphasizes defensive
security tools, which acts as a "SOC-in-a-box" and supports blue and purple team exercises.
Gotcha gotcha, I thought you were saying purple was strictly defensive tools. My misunderstanding
my Man.
I used to use Purple a lot but there seemed to be a number of malicious mirrors that were
troublesome. Although it did educate me heavily on rootkits and bootkits so I'm definitely grateful for
that (now LMFAO)

Have any other distro's of note?

????????????? - October 6, 2025, 11:51 am

(Oct 6, 2025 11:43 AM)G-DillahKillah Wrote: [\[CITE\]](#) Gotcha gotcha, I thought you were saying purple
was strictly defensive tools. My misunderstanding my Man.

I used to use Purple a lot but there seemed to be a number of malicious mirrors that were troublesome. Although it did educate me heavily on rootkits and bootkits so I'm definitely grateful for that (now LMFAO)

Have any other distro's of note?

Parrot OS is also a good flavor if you are into these things.

G-DillahKillah - October 6, 2025, 11:59 am

(Oct 6, 2025 11:51 AM)????????????? Wrote: [\[CITE\]](#) Parrot OS is also a good flavor if you are into these things.

True true but it's Already in the tool.

It has Kali Blue, Parrot os, Fedora and Kodachi linux

s0lo - October 6, 2025, 10:31 pm

never heard about Kodachi linux, interesting to try it out, been using mostly Kali and Qubes

G-DillahKillah - October 7, 2025, 8:34 pm

(Oct 6, 2025 10:31 PM)s0lo Wrote: [\[CITE\]](#) never heard about Kodachi linux, interesting to try it out, been using mostly Kali and Qubes

I love Kodachi but if you run it in a vm it's unbearably slow. It's reached it's end of life but nine is set to release soon. Always wanted to try Qubes.