

# LAN-IP-Scanner

Dieses Python-Skript scannt ein lokales Netzwerk nach aktiven IP-Adressen. Es verwendet den Befehl ping, um zu überprüfen, ob ein Host erreichbar ist, und setzt Multithreading ein, um den Scavorgang zu beschleunigen. Ein Semaphore begrenzt die Anzahl gleichzeitiger Threads, um eine Überlastung des Systems zu vermeiden. Das Skript nimmt eine Netzwerkadresse (z. B. "192.168.1.0/24") als Eingabe entgegen und gibt für jede IP-Adresse im Netzwerk aus, ob sie aktiv oder inaktiv ist.

```
import subprocess
import ipaddress
import threading
import os

MAX_THREADS = 255 # Maximale Anzahl zu verwendeter Threads

def is_host_up(host):
    """
    Überprüft mit ping, ob ein Host aktiv ist.
    Gibt True zurück, wenn der Host aktiv ist, andernfalls False.
    """
    try:
        # -c 1: Sendet nur 1 Paket
        # -W 1: Warte 1 Sekunde auf eine Antwort
        subprocess.check_output(["ping", "-c", "1", "-W", "1", host], timeout=1)
        return True
    except subprocess.CalledProcessError:
        return False
    except subprocess.TimeoutExpired:
        return False

def scan_ip(ip_str):
    """
    Scannt eine einzelne IP-Adresse und gibt ihren Status aus.
    """
    if is_host_up(ip_str):
        print(f"{ip_str} ist aktiv")
    else:
        print(f"{ip_str} ist inaktiv")

def scan_network(network):
```

"""

Scannt ein Netzwerk nach aktiven Hosts unter Verwendung von Threads und begrenzt die Anzahl gleichzeitiger  
Threads.

```
print(f"Scanne Netzwerk: {network}")
threads = []
semaphore = threading.Semaphore(MAX_THREADS) # Begrenzt die Anzahl gleichzeitiger Threads

def scan_ip_with_semaphore(ip_str):
    semaphore.acquire()
    try:
        scan_ip(ip_str)
    finally:
        semaphore.release()

for ip in ipaddress.IPv4Network(network):
    ip_str = str(ip)
    thread = threading.Thread(target=scan_ip_with_semaphore, args=(ip_str,))
    threads.append(thread)
    thread.start()

for thread in threads:
    thread.join()

if __name__ == "__main__":
    network_to_scan = "192.168.1.0/24" # Ändern Sie dies in Ihr Netzwerk
    scan_network(network_to_scan)
```