

# LAN IP スキャナー

この Python スクリプトは、ローカルネットワーク上のアクティブな IP アドレスをスキャンします。ホストが到達可能かどうかを確認するために ping コマンドを使用し、スキャン処理の高速化のためにマルチスレッド処理を採用しています。システムへの過剰な負荷を避けるために、セマフォによって同時実行されるスレッド数を制限しています。スクリプトはネットワークアドレス（例：「192.168.1.0/24」）を入力として受け取り、ネットワーク内の各 IP アドレスが稼働中か停止中かを印字します。

```
import subprocess
import ipaddress
import threading
import os

MAX_THREADS = 255 # 使用するスレッドの最大数

def is_host_up(host):
    """
    ping を使用してホストが稼働中かどうかを確認します。
    ホストが稼働中の場合は True、それ以外の場合は False を返します。
    """
    try:
        # -c 1: パケットを 1 つだけ送信
        # -W 1: 応答を 1 秒間待つ
        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):
    """
    単一の IP アドレスをスキャンし、そのステータスを出力します。
    """
    if is_host_up(ip_str):
        print(f"{ip_str} is up")
    else:
        print(f"{ip_str} is down")

def scan_network(network):
```

```
"""
```

スレッドを使用してライブホストをネットワークでスキャンし、同時実行されるスレッド数を制限します。

```
"""
```

```
print(f"Scanning network: {network}")
threads = []
semaphore = threading.Semaphore(MAX_THREADS) # 同時実行されるスレッド数を制限

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" # これをあなたのネットワークに変更してください
    scan_network(network_to_scan)
```