

# المحليّة الشبكيّة ماسح

إذا مما للتتحقق ping الأمر يستخدم نشطة. عنوانين عن بحث الم المحليّة الشبكيّة يفحص بآي ثون بلغة النصي البرنامنج هذا إثقال لتجنب الم تزامنة الخيوط عدد مرور إشارة يحدد الم فحص. عملية لتسريع الخيوط تعدد ويستخدم متاح، الم ضييف كان في عنوانين كل كانت إذا ما ويطبع كمدخل، 192.168.1.0 24 مثل الم الشبكيّة عنوان النصي البرنامنج يأخذ الم ظام. كا هل لـ. أم تعميل الم الشبكيّة

```
import subprocess
import ipaddress
import threading
import os

MAX_THREADS = 255 #


def is_host_up(host):
    """
        ping.
        True      .
    """
    try:
        # -c 1:
        # -W 1:
        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):
    """
        IP      .
    """
    if is_host_up(ip_str):
        print(f"{ip_str} is up")
    else:
        print(f"{ip_str} is down")


def scan_network(network):
    """
```

```
"""

print(f"Scanning network: {network}")

threads = []
semaphore = threading.Semaphore(MAX_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" #
    scan_network(network_to_scan)
```