

# Monitoraggio della Disponibilità di Host tramite ICMP (con Python)

## Descrizione

Questo script Python permette di monitorare lo stato di uno o più host di una rete, controllando la loro disponibilità tramite il protocollo ICMP (ping). Lo script consente all'utente di specificare gli indirizzi IP degli host da monitorare e visualizzare lo stato (online/offline) di ciascun host.

## Requisiti

- Python 3.x
- Librerie standard di Python:
  - subprocess
  - platform
  - time

## Struttura dello script

### Importazione delle librerie necessarie

```
import subprocess
import platform
import time
```

Queste librerie sono utilizzate per eseguire comandi di sistema (*subprocess*), determinare il sistema operativo (*platform*), e gestire le pause (*time*).

### Funzione 'ping host'

```
def ping_host(ip: str) -> bool:
    param = '-n' if platform.system().lower() == 'windows' else '-c'
    command = ['ping', param, '1', ip]

    try:
        response = subprocess.run(command, stdout=subprocess.PIPE,
stderr=subprocess.PIPE, timeout=5)
        if response.returncode == 0:
            return True # Host Online
        else:
            return False # Host Offline
    except subprocess.TimeoutExpired:
        return False # Timeout (Considero l'host offline)
```

La funzione prende come unico parametro una stringa contenente l'indirizzo IP da monitorare e restituisce un booleano.

La funzione per prima, in base al sistema operativo, determina l'opzione corretta per il comando 'ping'.

Poi, all'interno di un try, viene chiamato subprocess.run per eseguire il comando ping, catturando l'output e impostando un timeout di 5 secondi.

Infine controllo con un if il *returncode code* (codice di ritorno) per determinare se l'host è online (0) o offline (diverso da 0).

In caso di Timeout l'host viene considerato offline nello script.

### Funzione 'main'

```
def main():
    hosts = input("Inserisci gli indirizzi IP degli host da monitorare,
separati da spazio: ").split()

    while True:
        for ip in hosts:
            print(f"Verifica dello stato per l'host {ip}...")
            if ping_host(ip):
                print(f"L'host {ip} è online.")
            else:
                print(f"L'host {ip} è offline.")
            time.sleep(1) # Pausa per non sovraccaricare la rete

        command = input("\nPremi 'q' e invio per uscire, \n'n' per
inserire nuovi host, \no solo invio per monitorare nuovamente gli IP
inseriti: ").strip().lower()
        if command == 'n':
            hosts = input("Inserisci gli indirizzi IP degli host da
monitorare, separati da spazio: ").split()
        elif command == 'q':
            print("Interruzione del monitoraggio.")
            break
```

Nella funzione *main* viene richiesto all'utente di inserire gli indirizzi IP da monitorare separati da uno spazio e salvati in una lista.

Poi parte il ciclo *while* per consentire all'utente di continuare il monitoraggio fino a quando non viene interrotto dall'utente.

All'interno del *while*, il ciclo *for* per ogni indirizzo IP nella lista esegue il ping utilizzando la funzione *ping\_host* e viene stampato lo stato dell'host (online/offline). Viene inoltre introdotta una pausa di 1 secondo per evitare il sovraccaricamento della rete.

Alla fine del *for*, lo script chiede all'utente di inserire un comando:

- 'q' per interrompere il monitoraggio,
- 'n' per inserire e monitorare nuovi indirizzi IP da monitorare (gli IP già nella lista vengono sovrascritti),
- Premere semplicemente 'Invio' per continuare con il monitoraggio degli host già inseriti

## Utilizzo

1. Salvare lo script in una certa directory,
2. Aprire un terminale e navigare fino alla cartella in cui il file è salvato,
3. Eseguire lo script con '*python3 monitor\_rete.py*' (eventualmente cambiando il nome dello script se è stato salvato con un altro nome),
4. Quando richiesto, inserire gli indirizzi IP degli host da monitorare, separati da spazi,
5. Lo script eseguirà il ping degli host in un loop continuo.  
Per interrompere il loop, inserire 'q' quando richiesto e premere 'Invio',  
per monitorare nuovi indirizzi IP, inserisci 'n' e poi premi 'Invio',  
o premi semplicemente 'Invio' (senza inserire nulla, ovvero inserendo un qualsiasi altro carattere o comando diverso da 'q' ed 'n') per continuare a monitorare gli host già passati allo script.

## Considerazioni Aggiuntive

- **Permessi di Amministratore:** In alcuni sistemi operativi, potrebbe essere necessario eseguire il terminale con permessi di amministratore per poter utilizzare il comando *ping*,
- **Timeout:** Lo script utilizza un timeout di 5 secondi per il comando ping e nel caso l'host è considerato offline (anche se potrebbe non esserlo veramente). Questo valore può comunque essere modificato all'interno della funzione *ping\_host*, se necessario,
- **Compatibilità:** Lo script è progettato per essere compatibile con Windows, Linux e macOS.