

The Network Management Department is in need of two tools: a ping sweeper and a port scanner. The ping sweeper checks if a set of hosts are reachable, using the **ping** utility. The port scanner checks which ports are active on a given address, using the **nc** utility. It is your task to develop shell scripts to implement these two tools.

Ping Sweeper Script

Write a ping sweeper script that meets the following requirements.

- a) The script receives the name of a text file that contains a list of ipv4 addresses (one address per line) that must be tested, as a parameter.
- b) The script asks for the filename, should the filename be not received as parameter.
- c) The script must check if the name refers to an existing regular file. If it is not, the user has 3 opportunities to enter the name of an existing regular file, before the script terminates.
- d) Before each ping, the script must check if the address conforms to the ipv4 format.
- e) The connectivity is tested using the following command: **ping -c 1 <ipv4_address>**
- f) Whenever the response string contains '**100% packet loss**', the test failed.
- g) The script must generate a results file, named **reachability_test.txt**, containing the following information:
 - a. Number and list of active addresses;
 - b. Number and list of inactive addresses;
 - c. Number and list of invalid addresses.

Port Scanner Script

Write a port scanner script that meets the following requirements.

- a) The script must receive two parameters: (i) one providing the list of hosts to test, and (ii) other providing the port/list of ports to scan.
 - i. The list of hosts is provided with option **-h**, followed by the name of a text file, containing multiple hosts (described by name or address), one host per line.
 - ii. To ports to scan are given by one of two possible options: **-n** and **-f**. Option **-n** is followed by the number of one single port (i.e. an integer in the [1; 65535] range); option **-f** is followed by the name of a text file, containing multiple ports, one port per line.

Examples: **portscanner.sh -h hostfile.txt -n 80**
 portscanner.sh -f portfile.txt -h hostfile.txt

- b) The script must verify the correct number of parameters. Additionally, the script must ensure that option **-n** is followed by an integer in the [1; 65535] range, and if options **-f** and **-h** are followed by existing files. Should these validations fail, the script must terminate.
- c) You may test a port with command line **nc -zvw1 <host> <port>**. Whenever the port is active, the response contains the string **"succeeded"**.

Example:

```
result=$(nc -zvw1 www.google.com 443 2>&1 | grep succeeded)
```

- d) The script must generate a new file, named **scan_report.txt**, reporting for each host which ports are active, inactive and invalid.