



# Fawry | ステップ

## Task 1

```
#!/bin/bash

if [ "$1" == "--help" ]; then
    echo "How to use:"
    echo " $0 [options] word_to_find file_name"
    echo "Options:"
    echo " -n Show line numbers"
    echo " -v Show lines that do NOT match"
    exit 0
fi

if [ "$#" -lt 2 ]; then
    echo "Not enough arguments!"
    echo "Usage: $0 [options] word_to_find file_name"
    exit 1
fi
```

```

option=""
word_to_find=""
file_name=""

if [[ "$1" == -* ]]; then
    option="$1"
    word_to_find="$2"
    file_name="$3"
else
    word_to_find="$1"
    file_name="$2"
fi

if [ -z "$word_to_find" ] || [ -z "$file_name" ]; then
    echo "You forgot to give the word or the file!"
    exit 1
fi

if [ ! -f "$file_name" ]; then
    echo "Oops! File '$file_name' does not exist."
    exit 1
fi

line_number=0

while IFS= read -r current_line; do
    line_number=$((line_number + 1))

    line_in_small=$(echo "$current_line" | tr '[:upper:]' '[:lower:]')
    word_in_small=$(echo "$word_to_find" | tr '[:upper:]' '[:lower:]')

    if [[ "$line_in_small" == *"$word_in_small"* ]]; then
        is_match=true
    else
        is_match=false
    fi

```

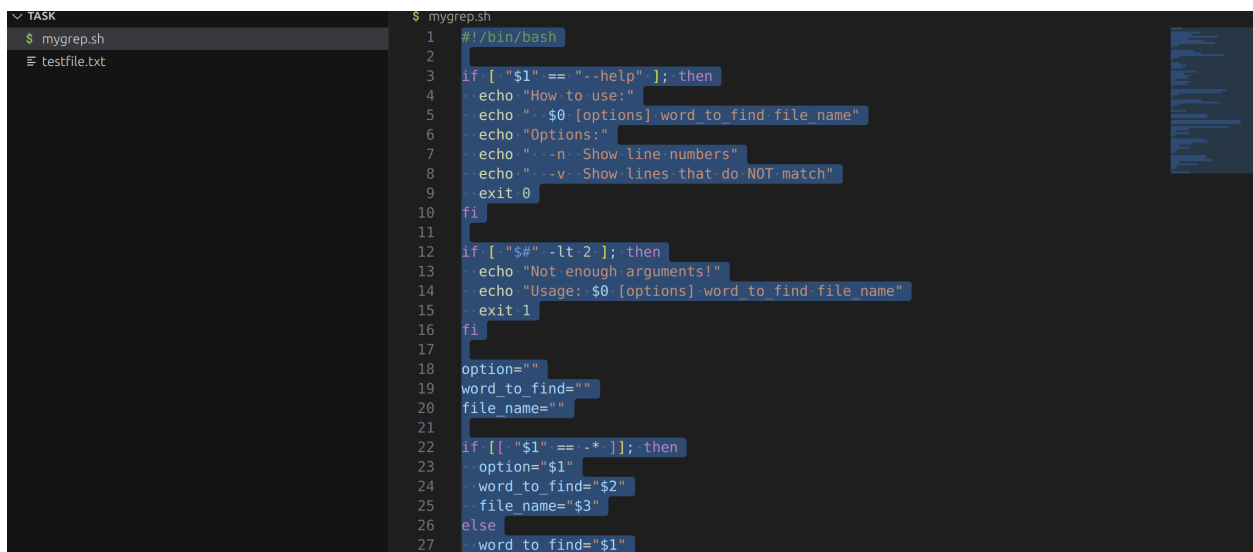
```

if [[ "$option" == *v* ]]; then
    if [ "$is_match" == true ]; then
        is_match=false
    else
        is_match=true
    fi
fi

if [ "$is_match" == true ]; then
    if [[ "$option" == *n* ]]; then
        echo "${line_number}:$current_line"
    else
        echo "$current_line"
    fi
fi

done < "$file_name"

```



```

TASK
$ mygrep.sh
testfile.txt

$ mygrep.sh
1  #!/bin/bash
2
3  if [ "$1" == "--help" ]; then
4      echo "How to use:"
5      echo "  $0 [options] word to find file name"
6      echo "Options:"
7      echo "  -n Show line numbers"
8      echo "  -v Show lines that do NOT match"
9      exit 0
10 fi
11
12 if [ "$#" -lt 2 ]; then
13     echo "Not enough arguments!"
14     echo "Usage: $0 [options] word to find file name"
15     exit 1
16 fi
17
18 option=""
19 word_to_find=""
20 file_name=""
21
22 if [[ "$1" == -* ]]; then
23     option="$1"
24     word_to_find="$2"
25     file_name="$3"
26 else
27     word_to_find="$1"

```

```

## test file
Hello world
This is a test

```

another test line  
HELLO AGAIN  
Don't match this line  
Testing one two three

```
~/Task
./mygrep.sh hello testfile.txt
Hello world
HELLO AGAIN

~/Task
./mygrep.sh -n hello testfile.txt
1:Hello world
4:HELLO AGAIN

~/Task
./mygrep.sh -vn hello testfile.txt
2:This is a test
3:another test line
5:Don't match this line
6:Testing one two three

~/Task
./mygrep.sh -v testfile.txt
You forgot to give the word or the file!

ERROR
```

```
~/Task
./mygrep.sh --help
How to use:
./mygrep.sh [options] word_to_find file_name
Options:
-n Show line numbers
-v Show lines that do NOT match

ERROR
```

## 1. How my script handles arguments and options:

First, the script checks if the user typed `--help` and shows how to use it. Then it makes sure there are enough arguments. It looks if the first thing is an option (like `-n` or `-v`). If yes, it saves it. After that, it reads the file line by line. It checks if each line has the word we are looking for (ignoring big/small letters). Depending on the

option, it either shows matching lines, shows lines that don't match ( `-v` ), and adds line numbers if needed ( `-n` ).

## 2. How I would change it for regex or `-i`/`-c`/`-l` options:

I would add more options when reading the input. For regex, I would use a different way to match lines (like `[[ $line =~ $pattern ]]` ). For `-i` , I would handle case sensitivity better. For `-c` , I would count how many matches and print the number at the end. For `-l` , I would just print the filename if there is a match.

## 3. Hardest part and why:

The hardest part was handling the options correctly, like making sure `-n` and `-v` work together. Also making sure it shows the right lines or skips them, without messing up.

# Task 2 | ステップ 2

## 1. Verify DNS Resolution

**Goal:** Check if `internal.example.com` is resolvable via:

### Commands:

```
# Check which DNS servers you are using
cat /etc/resolv.conf
```

```
# Try resolving using system DNS
nslookup internal.example.com
```

```
# Now manually specify Google's DNS
nslookup internal.example.com 8.8.8.8
```

```
cat /etc/resolv.conf | grep nameserver
nameserver 127.0.0.53

nslookup internal.example.com
Server:      127.0.0.53
Address:     127.0.0.53#53

** server can't find internal.example.com: NXDOMAIN

nslookup internal.example.com 8.8.8.8
Server:      8.8.8.8
Address:     8.8.8.8#53

** server can't find internal.example.com: NXDOMAIN
```

## 2. Diagnose Service Reachability

### Goal:

check if the service is reachable.

### Commands:

```
## i want to do it but there is no server to test
```

```
# Ping the resolved IP
```

```
ping <IP_ADDRESS>
```

```
# Check if ports 80 (HTTP) or 443 (HTTPS) are open
```

```
telnet <IP_ADDRESS> 80
```

```
telnet <IP_ADDRESS> 443
```

```
# OR
nc -zv <IP_ADDRESS> 80
nc -zv <IP_ADDRESS> 443

# OR use curl to try to connect
curl -I http://internal.example.com
curl -I https://internal.example.com

# On the server itself, check if service is listening
ss -tln | grep -E ':80|:443'
```

Layer	Possible Cause
DNS	- Wrong DNS entry- Missing A record- DNS server misconfigured- Local /etc/resolv.conf wrong- Firewall blocking DNS queries
Network	- Firewall blocking HTTP/HTTPS- Routing issues (wrong routes/subnet)
Service Level	- Web server (nginx/httpd) not bound to 0.0.0.0- Service listening only on localhost- Service down on specific node
Client Side	- /etc/hosts has wrong mapping- Cache issues (stale DNS entries)

## a) DNS Issues

- **How to Confirm:**

`nslookup` from system DNS but works with `8.8.8.8`.

- **Commands to Fix:**

- If /etc/resolv.conf wrong:

```
sudo nano /etc/resolv.conf
```

```
# Fix or add correct nameserver entries
```

```
nameserver 127.0.0.53  
options edns0 trust-ad  
search .
```

- need to flush local DNS cache:

```
sudo systemd-resolve --flush-caches
```

## b) Firewall Issues

- **How to Confirm:**










- `nc -zv <IP> 80` or `443` times out.
- `telnet` fails to connect.

- **Commands to Fix:**

with `ufw`:

```
sudo ufw allow 80/tcp  
sudo ufw allow 443/tcp  
sudo ufw reload
```



```
-   ~ 1m 0.556s ⌚ ✓ 19:24:40 ⌚  
- sudo ufw allow 80/tcp  
rules updated  
rules updated (v6)  
-   ~ 635ms ⌚ ✓ 19:26:00 ⌚  
- sudo ufw allow 443/tcp  
rules updated  
rules updated (v6)  
-   ~ ✓ 19:26:06 ⌚  
- sudo ufw reload  
firewall not enabled (skipping reload)  
-   ~ ✓ 19:26:10 ⌚  
- 
```

## d) Routing/Subnet Issues

- Use `traceroute` :

```
traceroute <IP_ADDRESS>
```

## Bonus

### Configure Local /etc/hosts Entry

#### Commands

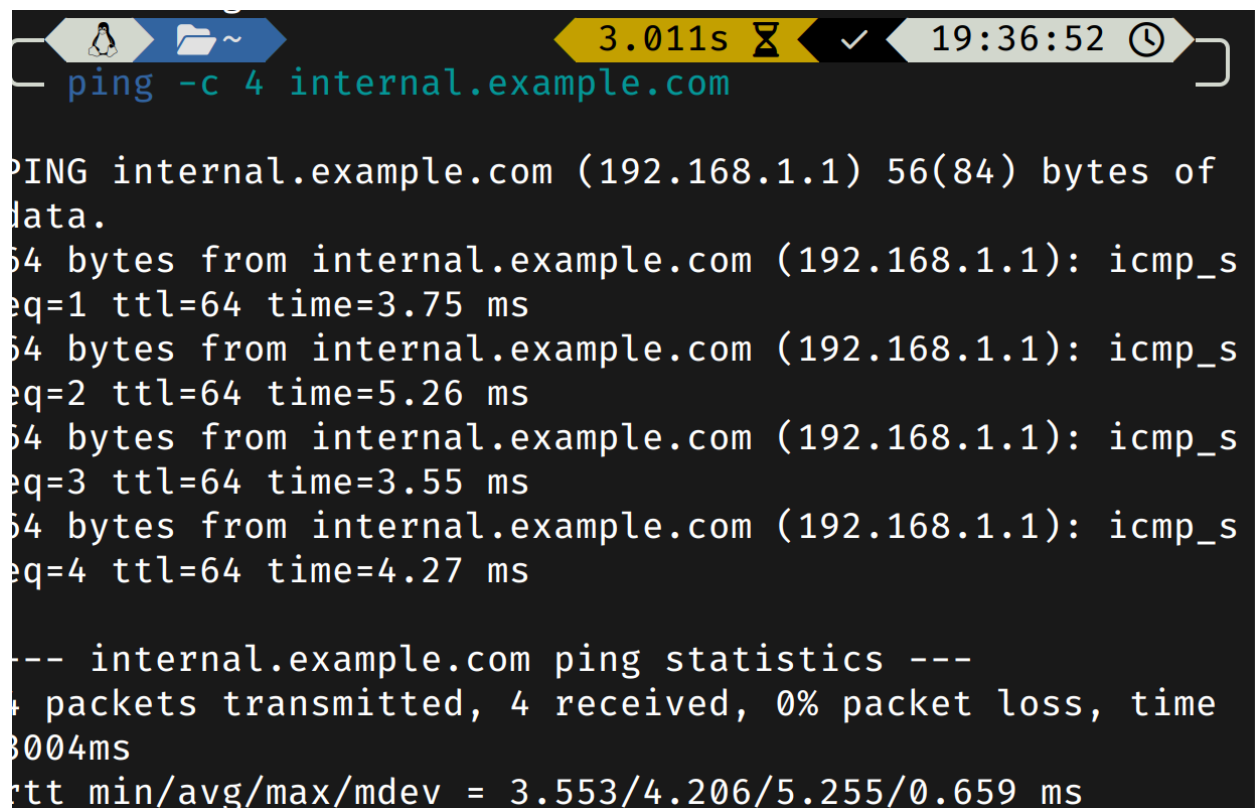
```
sudo nano /etc/hosts
```

Add:

```
192.168.1.1 internal.example.com
```

```
ping internal.example.com  
curl http://internal.example.com
```

Test:



A terminal window screenshot showing a successful ping command. The terminal has a dark background with light-colored text. At the top, there's a status bar with icons for a terminal, a folder, a timer showing 3.011s, a checkmark, and a clock showing 19:36:52. The command entered is `ping -c 4 internal.example.com`. The output shows four successful ping requests to 192.168.1.1 with varying response times. At the bottom, it displays the ping statistics: 4 packets transmitted, 4 received, 0% packet loss, and a total time of 3004ms. The round-trip times (rtt) are listed as min/avg/max/mdev = 3.553/4.206/5.255/0.659 ms.

```
ping -c 4 internal.example.com  
  
PING internal.example.com (192.168.1.1) 56(84) bytes of  
data.  
64 bytes from internal.example.com (192.168.1.1): icmp_s  
eq=1 ttl=64 time=3.75 ms  
64 bytes from internal.example.com (192.168.1.1): icmp_s  
eq=2 ttl=64 time=5.26 ms  
64 bytes from internal.example.com (192.168.1.1): icmp_s  
eq=3 ttl=64 time=3.55 ms  
64 bytes from internal.example.com (192.168.1.1): icmp_s  
eq=4 ttl=64 time=4.27 ms  
  
--- internal.example.com ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time  
3004ms  
rtt min/avg/max/mdev = 3.553/4.206/5.255/0.659 ms
```

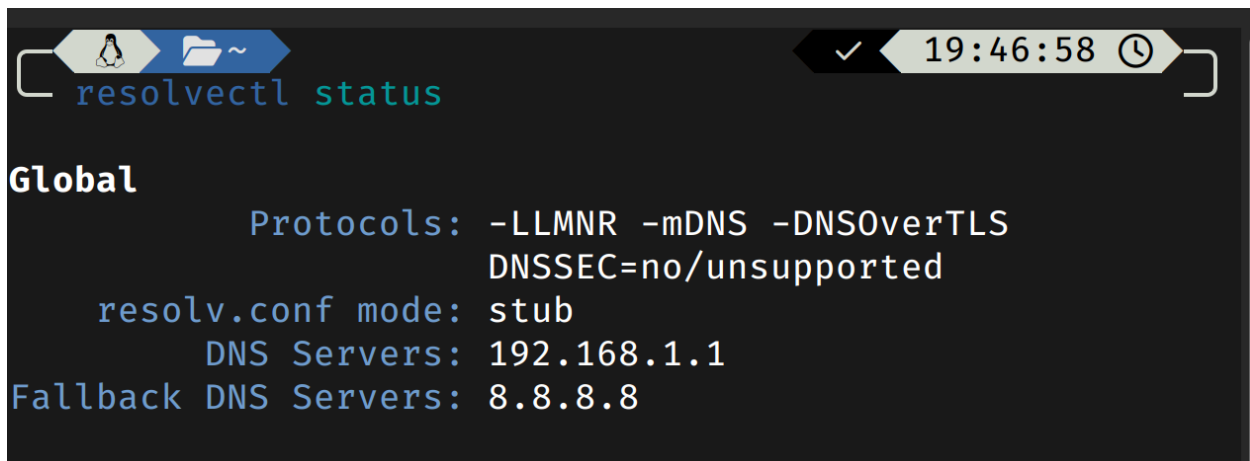
**Persist DNS Settings with systemd-resolved or NetworkManager**

```
# Check status
systemctl status systemd-resolved

# Set DNS servers persistently
sudo nano /etc/systemd/resolved.conf

[Resolve]
DNS=192.168.1.1
FallbackDNS=8.8.8.8

sudo systemctl restart systemd-resolved
```



```
resolvectl status

Global
      Protocols: -LLMNR -mDNS -DNSOverTLS
                DNSSEC=no/unsupported
      resolv.conf mode: stub
      DNS Servers: 192.168.1.1
Fallback DNS Servers: 8.8.8.8
```