TCSS 562: Software Engineering for Cloud Computing
Fall 2021
http://faculty.washington.edu/wlloyd/courses/tcss562

School of Engineering and Technology
University of Washington – Tacoma

Tutorial 2 – Introduction to Bash Scripting

Disclaimer: Subject to updates as corrections are found
Version 0.10
Scoring: 20 pts maximum

The purpose of this tutorial is to introduce Bash scripting under the Linux operating system, while also introducing web services, and the use of curl as a command-line HTTP REST web service client. Complete this tutorial leveraging your Linux environment set up for Tutorial 1. Please review sections 1 – 8 of the online Bash Scripting Tutorial as needed to complete the scripting activity:

Bash Scripting Tutorial:
https://ryanstutorials.net/bash-scripting-tutorial/

Tutorial Sections include:
1. What is a BASH script?
2. Variables
3. Input
4. Arithmetic
5. If Statements
6. Loops
7. Functions
8. User Interface

At the conclusion of the online Bash tutorial, please complete the Bash scripting task described below. As needed, search the Internet to find BASH examples beyond Ryan's tutorial to help with the overall programming task. Submit your completed operational Bash script as a file called **weather.sh** online via Canvas. While it is possible to perform the implementation in Python, the goal here is to gain experience using Bash and curl.

BASH WEATHER FORECAST TOOL

Write a short Bash script that consumes two web services to obtain a localized weather forecast based on the latitude and longitude of your computer's Internet connection. To complete the script it is recommended to use the following commands:

Command	Description
curl	Command line http REST client for performing GET PUT POST requests, etc. If curl has not already been installed, it can be installed with: sudo apt install curl When using curl, please use the "-s" flag to request silent output without continuous status information
cut	Cut is simple parsing tool in Bash. Simply pipe text to cut, and specify a custom column delimiter with "-d", then specify the desired column with "-f".

jq	Jq is a Bash JSON parser. You will need to install jq as follows: **sudo apt install jq**
awk	Awk can easily parse individual columns of a file: (for column 2) **cat myfile.txt | awk '{ print $2 }'**

Use the following two web services to obtain a weather forecast using the geolocation (latitude and longtitude coordinates) of your internet providers IP address.

IP Address Location API **https://ipinfo.io/developers**

The "IP Address Location API" provides latitude and longitude based on the IP address of your incoming web service request. The API is free to use and does not require registration. The web service never sees your internal IP address inside the "firewall" of your business, school, or home. For example, at home you may receive an IP address from a Comcast modem of "10.0.0.50". This is an internal IP address. Internal IP addresses are described in the table:

Private IPv4 addresses

RFC1918 name	IP address range	host id size
24-bit block	10.0.0.0 – **10.255**.255.255	**24 bits**
20-bit block	172.16.0.0 – 172.31.255.255	**20 bits**
16-bit block	192.168.0.0 – 192.168.255.255	**16 bits**

The "IP Address Location API" will provide metadata regarding your Internet service provider in JSON format as follows:


```
{
        "ip": "71.209.4.118",
        "hostname": "c-71-209-4-118.hsd1.wa.comcast.net",
        "city": "Tacoma",
        "region": "Washington",
        "country": "US",
        "loc": "47.2529,-122.4417",
        "postal": "98402",
        "org": "AS33650 Comcast Cable Communications, LLC"
}
```


Using the "jq" function, parse the latitude and longitude, and pass this to the Weatherbit API.

Because you're limited to 1,000 free requests per day, you should cache this information using a local hidden file called ".myipaddr". In Bash, at the start of your script, check for the existence of a file called ".myipaddr". If the file does not exist, call the service to

obtain the JSON object. Save the JSON object to the disk. Read the JSON object into a Bash variable using the "cat" command. Then process the JSON with "jq".

Print out the following messages based on whether you're using a cached IP from ".myipaddr", or whether your script had to call the service to obtain the JSON describing your IP and location:

echo "CALLING API TO QUERY MY IP"

echo "IP READ FROM CACHE"

See the online documentation for more information: https://ipinfo.io/developers

Weatherbit API https://www.weatherbit.io/api

Use the Weatherbit API to obtain a 16-day weather forecast with High and Low temperature information. The Weatherbit API provides a detailed 16-day forecast on request. This API is also limited to 1,000 calls per day. If you are concerned that debugging your script will exceed 1,000 calls per day, please consider caching the JSON object as described above for the IP Address Location API. Then write your parsing code using the cached JSON object to save calls to the web service.

You will need to register to obtain an API Key to use the Weatherbit API. Keys are generally made available within a couple of minutes. Once having a key, you can check your daily usage quota with the following call:

curl -s -g https://api.weatherbit.io/v2.0/subscription/usage?key=[YOUR-API-KEY]

Specify your actual apikey (a combination of letters and numbers) in place of "[YOUR-API-KEY]".

To obtain the 16-day weather forecast, use the following API:
https://www.weatherbit.io/api/weather-forecast-16-day

It is recommended to use the fields "max_temp" and "min_temp" to obtain the forecast.

***** To call the weatherbit services with CURL put the URL in "quotes" *****

The following online JSON formatter is helpful to paste JSON into a web browser to rapidly make it more readable: **https://jsonlint.com/**

Your weather.sh script should produce output as below:

Sample Output – First Call:
CALLING API TO QUERY MY IP
Forecast for my lat=47.2529°, lon=-122.4443°
Forecast for 2021-10-07 HI: 11.5°c LOW: 8.7°c
Forecast for 2021-10-08 HI: 12.5°c LOW: 7.5°c
Forecast for 2021-10-09 HI: 12.5°c LOW: 7°c
Forecast for 2021-10-10 HI: 14.6°c LOW: 7.3°c

Forecast for 2021-10-11 HI: 13.8°c LOW: 4.4°c
Forecast for 2021-10-12 HI: 14.6°c LOW: 3.6°c
Forecast for 2021-10-13 HI: 13.5°c LOW: 7.3°c
Forecast for 2021-10-14 HI: 13.8°c LOW: 7.6°c
Forecast for 2021-10-15 HI: 10.3°c LOW: 9.1°c
Forecast for 2021-10-16 HI: 10.9°c LOW: 8.9°c
Forecast for 2021-10-17 HI: 10.9°c LOW: 8.7°c
Forecast for 2021-10-18 HI: 11.3°c LOW: 7.5°c
Forecast for 2021-10-19 HI: 10.3°c LOW: 6.2°c
Forecast for 2021-10-20 HI: 13.7°c LOW: 6.7°c
Forecast for 2021-10-21 HI: 15.1°c LOW: 9.9°c
Forecast for 2021-10-22 HI: 14.6°c LOW: 14.6°c

Sample Output – Subsequent Calls:
IP READ FROM CACHE
Forecast for my lat=47.2529°, lon=-122.4443°
Forecast for 2021-10-07 HI: 11.5°c LOW: 8.7°c
Forecast for 2021-10-08 HI: 12.5°c LOW: 7.5°c
Forecast for 2021-10-09 HI: 12.5°c LOW: 7°c
Forecast for 2021-10-10 HI: 14.6°c LOW: 7.3°c
Forecast for 2021-10-11 HI: 13.8°c LOW: 4.4°c
Forecast for 2021-10-12 HI: 14.6°c LOW: 3.6°c
Forecast for 2021-10-13 HI: 13.5°c LOW: 7.3°c
Forecast for 2021-10-14 HI: 13.8°c LOW: 7.6°c
Forecast for 2021-10-15 HI: 10.3°c LOW: 9.1°c
Forecast for 2021-10-16 HI: 10.9°c LOW: 8.9°c
Forecast for 2021-10-17 HI: 10.9°c LOW: 8.7°c
Forecast for 2021-10-18 HI: 11.3°c LOW: 7.5°c
Forecast for 2021-10-19 HI: 10.3°c LOW: 6.2°c
Forecast for 2021-10-20 HI: 13.7°c LOW: 6.7°c
Forecast for 2021-10-21 HI: 15.1°c LOW: 9.9°c
Forecast for 2021-10-22 HI: 14.6°c LOW: 14.6°c

How to type the degree symbol in Ubuntu/BASH:
To produce the degrees symbol in Ubuntu/BASH simply type [CONTROL]-[SHIFT]-[U] as a three-key combination and then quickly type “b0” and press [ENTER]. The degree symbol should appear: °

[CONTROL]-[SHIFT]-[U] enables Unicode characters to be entered such (4E91) as: 今

Scoring
The Bash weather script (weather.sh) will be scored out of 20 points:

- 10 points for being functionally correct. It should obtain and display a 16-day weather forecast for the lat/long of the internet service provider.
- 10 additional points are for proper formatting. Formatting should match the output as described in the tutorial specification.