Weather Project

Description: A project that gets weather data from a free open source and logs the daily temperature automatically.

Steps:

* Create a weather log file. A history of daily weather data

touch rx\_poc.log

* Adding headers to the report

header=$(echo -e "year\tmonth\tday\thour\tobs\_tmp\tfc\_temp")

echo $header>rx\_poc.log

-e allows for interpretation of backslash escape sequences

* Set a format for the weather data

today=$(date +%Y%m%d)

weather\_report=raw\_data\_$today

* Download the data for a city

city=Casablanca

curl wttr.in/$city --output $weather\_report

Tips: tr - trim repeated characters to a single character.

For example, to remove extra spaces from text:

echo "There are too many spaces in this sentence." | tr -s " "

xargs - can be used to trim leading and trailing spaces from a string.

For example, to remove the spaces from the begining and the end of text:

echo " Never start or end a sentence

rev - reverse the order of characters on a line of text.

Try entering the following commmand:

echo ".sdrawkcab saw ecnetnes sihT" | rev

You will find rev to be a useful operation to apply in combination with the cut command.  
For example, suppose you want to access the last field in a delimited string of fields:

# print the last field of the string

echo "three two one" | rev | cut -d " " -f 1 | rev

You will also find xargs to be a useful operation to apply in combination with the cut command.  
For example, suppose you want to access the last word in a sentence as above, but there happens to be an extra space at the end:

# Unfortunately, this prints the last field of the string, which is empty:

echo "three two one " | rev | cut -d " " -f 1 | rev

# But if you trim the trailing space first, you get the expected result:

echo "three two one " | xargs | rev | cut -d " " -f 1 | rev

* Extract the temp

obs\_tmp=$(head -1 temperatures.txt | tr -s " " | xargs | rev | cut -d " " -f2 | rev)

* Extract tomorrows temperature

fc\_temp=$(head -3 temperatures.txt | tail -1 | tr -s " " | xargs | cut -d "C" -f2 | rev | cut -d " " -f2 | rev)

* Extract current day, hour, month and year

hour=$(TZ='Morocco/Casablanca' date -u +%H)

day=$(TZ='Morocco/Casablanca' date -u +%d)

month=$(TZ='Morocco/Casablanca' date +%m)

year=$(TZ='Morocco/Casablanca' date +%Y)

* Merge the fields to form a record

record=$(echo -e "$year\t$month\t$day\t$hour\t$obs\_tmp\t$fc\_temp")

echo $record>>rx\_poc.log

* Create a bash file to be run periodically

touch rx\_poc.sh

* Modify the permissions

Chmod u+x rx\_poc.sh

* Determine the time of day to run the script

date

date -u

* Create a cron job

Crontab -e

A complete bash script:

#! /bin/bash

# create a datestamped filename for the raw wttr data:

today=$(date +%Y%m%d)

weather\_report=raw\_data\_$today

# download today's weather report from wttr.in:

city=Casablanca

curl wttr.in/$city --output $weather\_report

# use command substitution to store the current day, month, and year in corresponding shell variables:

hour=$(TZ='Morocco/Casablanca' date -u +%H)

day=$(TZ='Morocco/Casablanca' date -u +%d)

month=$(TZ='Morocco/Casablanca' date +%m)

year=$(TZ='Morocco/Casablanca' date +%Y)

# extract all lines containing temperatures from the weather report and write to file

grep °C $weather\_report > temperatures.txt

# extract the current temperature

obs\_tmp=$(head -1 temperatures.txt | tr -s " " | xargs | rev | cut -d " " -f2 | rev)

# extract the forecast for noon tomorrow

fc\_temp=$(head -3 temperatures.txt | tail -1 | tr -s " " | xargs | cut -d "C" -f2 | rev | cut -d " " -f2 |rev)

# create a tab-delimited record

# recall the header was created as follows:

# header=$(echo -e "year\tmonth\tday\thour\_UTC\tobs\_tmp\tfc\_temp")

# echo $header>rx\_poc.log

record=$(echo -e "$year\t$month\t$day\t$obs\_tmp\t$fc\_temp")

# append the record to rx\_poc.log

echo $record>>rx\_poc.log

An example of weather report in the shell

A screenshot of a computer screen

Description automatically generated