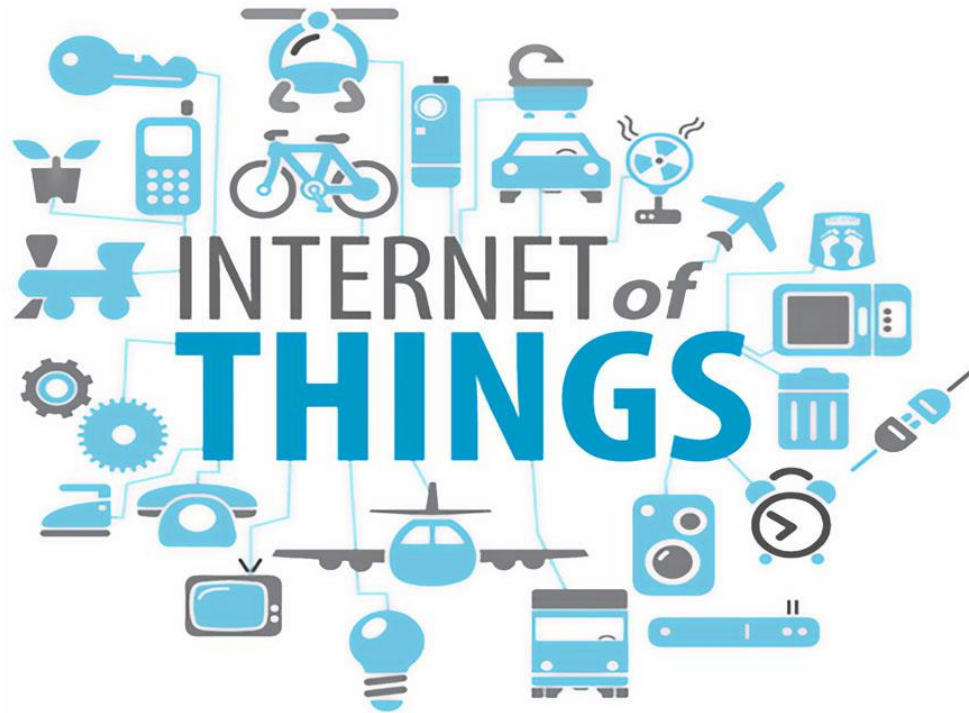




2021

ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΕΡΓΑΣΙΑ ΕΡΓΑΣΤΗΡΙΟΥ



ΔΕΥΤΕΡΑ 14.00-16.00

ΧΡΙΣΤΟΦΟΡΙΔΟΥ ΜΑΡΙΑ 47131

ΠΑΔΑ-ΜΠΥ

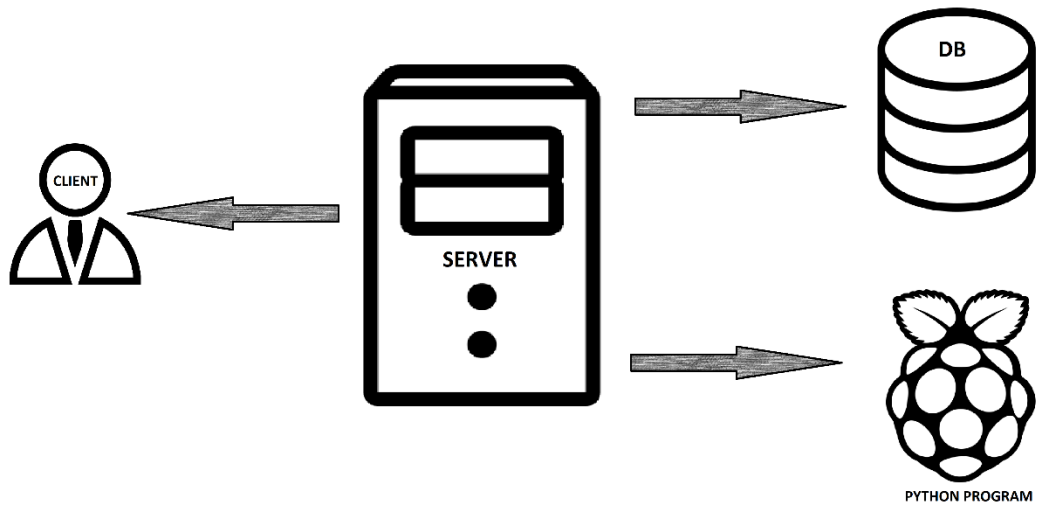
1/24/2021

Contents

Σχήμα αρχιτεκτονικής	2
Ερώτημα 1	3
Ερώτημα 2	5
Αποτελέσματα	6

Σχήμα αρχιτεκτονικής

Το σύστημα που ζητήθηκε να υλοποιηθεί φαίνεται στο παρακάτω σχήμα.



Υπάρχει ο server ο οποίος παίρνει δεδομένα από ένα πρόγραμμα που εκτελείται σε πλατφόρμα Raspberry Pi. Έπειτα τα δεδομένα αυτά αποθηκεύονται σε μία βάση δεδομένων. Τέλος, ο server μπορεί να εμφανίσει γραφικά τα αποτελέσματα των δεδομένων στο interface των πελατών του.

Ερώτημα 1

Για το ερώτημα αυτό δημιουργήθηκε κώδικας Python που υλοποιεί έναν εικονικό περιβαλλοντικό σταθμό. Ο κώδικας φαίνεται στη συνέχεια.

```
1 import random, requests, json
2 from time import sleep
3
4 url = 'http://localhost:3000'
5 previous = [0, 0, 0, 0] #holding the previous values
6 current = [0, 0, 0, 0] #holding the current values
7 data_names = ["temperature", "air humidity", "ground humidity", "beaufort"]
8 first = 0
9 percentage_dif = 0
10 timer = 0
11
12 while True:
13     temperature = random.randint(-100,100) #temperature
14     air_hum = random.randint(0,100) #air humidity
15     ground_hum = random.randint(0,100) #ground humidity
16     wind = random.randint(0,12) #Beaufort scale for wind
17
18     print("Temperature:", temperature, "C")
19     print("Air humidity:", air_hum, "%")
20     print("Ground humidity:", ground_hum, "%")
21     print("Wind:", wind,"Beaufort\n")
22
23     current[0]=temperature #putting the current values in the list
24     current[1]=air_hum
25     current[2]=ground_hum
26     current[3]=wind
27
28     if first == 1: #this code works for the second time so as to have current and previous values in the lists
29         print("Current temperature: ",current[0], ", Previous temperature: ", previous[0])
30         print("Current air humidity: ",current[1], ", Previous air humidity: ", previous[1])
31         print("Current ground humidity: ",current[2], ", Previous ground humidity: ", previous[2])
32         print("Current wind: ",current[3], ", Previous wind: ", previous[3],"\n")
33
34         if timer==10: #sending current data after 5 minutes of not sending anything
35             print("Sending everything to server...")
36         else:
37             for i in range(4):
38                 if (current[i] == 0) and (previous[i] == 0): #if both values are 0, move on to the next element
39                     continue
40                 elif previous[i] == 0: #cannot have division by 0
41                     percentage_dif = 11 #giving a random value >10 so as to send the current value to the server,
42                     #since the previous one was 0 and any change is important to send
43                 else: #calculating percentage difference of values
44                     percentage_dif = round(((current[i]-previous[i])/abs(previous[i]))*100,2)
45                     print("Percentage difference between previous and current value of", data_names[i], ":", percentage_dif)
46
47                 if (percentage_dif > 10) or (percentage_dif < -10): #if the difference is more than 10%, current value is sent to server
48                     print("Sending current value to server...")
49
50                     data = {0:current[i]} #sending previous value
51                     r = requests.post(url, data) #sending json object to specified url
52                     timer = 0 #when data is sent to server, make timer 0
53
54             print("\n\n")
55
56     sleep(30) #to generate the data every 30 seconds
57     timer+=1
58
59     previous[0]=temperature #putting the current values in the list holding the previous values
60     previous[1]=air_hum
61     previous[2]=ground_hum
62     previous[3]=wind
63
64     first = 1
65
```

Αρχικά, δημιουργούνται 4 ψευδοτυχαίες τιμές για να προσομοιωθεί η λήψη μετρήσεων για τη θερμοκρασία, την υγρασία του αέρα, την υγρασία του εδάφους και την ένταση του αέρα. Στη συνέχεια αυτές οι τιμές τοποθετούνται σε μια λίστα (current) που κρατάει τις τρέχουσες τιμές των 4 αυτών στοιχείων.

Χρησιμοποιείται μια δομή επανάληψης για να γίνει έλεγχος για όλες τις θέσεις της λίστας current αλλά και της λίστας previous που περιέχει της προηγούμενες τιμές καθενός από τα 4 στοιχεία. Αν το κάθε στοιχείο είναι 0 και στις δυο λίστες, η επανάληψη προχωράει στο επόμενο, χωρίς να στείλει στον server

την τρέχουσα τιμή του στοιχείου αυτού. Αν η προηγούμενη τιμή του στοιχείου είναι 0, δεν μπορεί να χρησιμοποιηθεί ο τύπος για τη μέτρηση της ποσοστιαίας μεταβολής και χρησιμοποιείται η μεταβλητή που αλλιώς θα κρατούσε τη μεταβολή αυτή (percentage_dif), έτσι ώστε να αποσταλεί η τρέχουσα τιμή του στοιχείου αυτού.

Μετά από αυτούς τους ελέγχους γίνεται υπολογισμός της ποσοστιαίας μεταβολής για κάθε στοιχείο με τη βοήθεια του τύπου $\frac{current - previous}{|previous|} * 100$. Αν η μεταβολή είναι μεγαλύτερη του 10 γίνεται αποστολή δεδομένων στον server με τη μέθοδο post της βιβλιοθήκης request της http.

Παρακάτω στον κώδικα φαίνεται η χρήση του sleep(30), έτσι ώστε να παράγονται ψευδοτυχαίες τιμές κάθε 30 δευτερόλεπτα. Τέλος, οι τρέχουσες τιμές μπαίνουν στη λίστα που κρατάει τις προηγούμενες έτσι ώστε να κρατηθούν και να αποθηκευτούν οι επόμενες τιμές. Επίσης υπάρχει ένας timer ο οποίος μετά από 10 επαναλήψεις(10*30sec=300sec=5min), θα πρέπει να αποστέλλει τα δεδομένα στον server, εφόσον δεν έχουν αποσταλεί μέσα στο χρονικό διάστημα των 5 λεπτών. Το flag first χρησιμοποιείται έτσι ώστε ο κώδικας που θα εκτελεστεί να εκτελεστεί αφού έχουν παραχθεί τιμές για 2^η φορά για να υπάρχουν προηγούμενες και τρέχουσες τιμές.

Ερώτημα 2

Για το ερώτημα αυτό δημιουργήθηκε server για να συλλέγει τα δεδομένα που αποστέλλει το πρόγραμμα Python που λειτουργεί ως η εικονική συσκευή. Ο κώδικας φαίνεται στη συνέχεια.

```
1 const express = require('express');  
2 const app = express();  
3  
4 app.listen(3000, () => console.log('Server listening on port 3000'));
```

Αρχικά γίνεται δήλωση της μεταβλητής `express` για να χρησιμοποιηθεί το node package `express`. Έπειτα, δημιουργείται ένα web application καλώντας τη συνάρτηση `express`, το `app`, το οποίο χρησιμοποιεί και εκτελεί πλέον το πακέτο `express`. Τέλος, προσδιορίζεται το `port` στο οποίο ακούει ο server για clients.

Αποτελέσματα

Τα αποτελέσματα του παραπάνω κώδικα είναι τα εξής.

```
marlach@marlach:~/Desktop$ python3 env_station.py
Temperature: -82 C
Air humidity: 96 %
Ground humidity: 99 %
Wind: 5 Beaufort

Temperature: 0 C
Air humidity: 13 %
Ground humidity: 60 %
Wind: 11 Beaufort

Current temperature: 0 , Previous temperature: -82
Current air humidity: 13 , Previous air humidity: 96
Current ground humidity: 60 , Previous ground humidity: 99
Current wind: 11 , Previous wind: 5

Percentage difference between previous and current value of temperature : 100.0
Sending current value to server...
Percentage difference between previous and current value of air humidity : -86.46
Sending current value to server...
Percentage difference between previous and current value of ground humidity : -39.3
9
Sending current value to server...
Percentage difference between previous and current value of beaufort : 120.0
Sending current value to server...

^CTraceback (most recent call last):
  File "env_station.py", line 56, in <module>
    sleep(30) #to generate the data every 30 seconds
KeyboardInterrupt
marlach@marlach:~/Desktop$
```

```
marlach@marlach:~/Desktop$ node server.js
Server listening on port 3000
```

Αριστερά φαίνεται το παράθυρο του περιβαλλοντικού σταθμού και δεξιά το παράθυρο του server. Στο σταθμό οι μετρήσεις δημιουργούνται επ' άπειρο μέχρι να σταματήσει χειροκίνητα με ctrl+C λόγω του κώδικα env_station.py. Εμφανίζονται οι πρώτες μετρήσεις, οι δεύτερες, στη συνέχεια γίνεται η εκτύπωση της ποσοστιαίας μεταβολής μεταξύ των 2 τιμών και τέλος η εκτύπωση του μηνύματος πως η τρέχουσα τιμή αποστέλλεται στον server. Έπειτα, φαίνονται οι επόμενες μετρήσεις για τις οποίες γίνεται η ίδια διαδικασία κ.ο.κ. Στο παράθυρο του server εμφανίζεται το μήνυμα πως ο server «ακούει» στο port 3000.