ΑΝΑΦΟΡΑ ΜΑΘΗΜΑΤΟΣ ΣΥΣΤΗΜΑΤΑ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ ΜΙΧΑΗΛΙΔΗΣ ΓΙΩΡΓΟΣ 7480

Θα εξηγήσω αναλυτικά τον τρόπο διασύνδεσης των raspberry με τους υπολογιστές ,τον τρόπο αντιμετώπισης του προβλήματος,

καθώς και πράγματα σχετικά με τον κώδικα που έγραψα.

Χρησιμοποίησα περιβάλλον-λειτουργικό σύστημα Linux-Ubuntu.

Αναλυτικότερα:

1. ### Τρόπος Αντιμετώπισης προβλήματος

   Χρησιμοποίησα μοντέλο peer to peer όπου το κάθε raspberry είναι και client και server..
   Έγινε χρήση adhoc δικτύου όπου δεν χρησιμοποιήθηκαν rooters αλλά σύνδεση έγινε μηχάνημα μηχάνημα. Και πιο συγκεκριμένα:
   Χρησιμοποιήθηκαν 2 raspberrys και 2 υπολογιστές για αυτό το σκοπό.
   Τα 2 raspberry επικοινώνησαν μεταξύ τους μέσω adhoc Wifi δικτύου(ασύρματα) όπου δημιούργησε το κάθε raspberry με κωδικό SSID piZero-ESPX
   Επίσης συνδέθηκε το ένα raspberry μέσω της εντολής ssh με τον έναν υπολογιστή μέσω της θύρας του raspberry micro-USB (ενσύρματα) και ένα ακόμα raspberry αντίστοιχα με έναν ακόμα υπολογιστή(ενσύρματα)..
   Ο στόχος ήταν να ανοίξει το command line του κάθε raspberry στον κάθε υπολογιστή ώστε με τις κατάλληλες εντολές και τις κατάλληλες ρυθμίσεις που θα αναλυθούν παρακάτω να επικοινωνήσουν τα raspberry μεταξύ τους χωρίς τη χρήση rooter δημιουργώντας μεταξύ τους ένα adhoc δίκτυο.(όχι οι υπολογιστές μεταξύ τους.. τα raspberrys)
   Επίσης το κάθε laptop με το κάθε raspberry συνδέθηκε μέσω ethernet με καλώδιο συνεπώς στο αντίστοιχο αρχείο του raspberry μπήκαν διευθύνσεις 192.168.0.[2-254](πεδίο address στο αντίστοιχο αρχείο) ενώ στου υπολογιστή 192.168.0.1/24(πεδίο gateway στο αντίστοιχο αρχείο)

Ενώ τα raspberry μεταξύ τους συνδέθηκαν μέσω wifi protocol επομένως μπήκαν διευθύνσεις 10.0.0.1/8(ασύρματα)
Για την επικοινωνία των συσκευών μέσω WiFi χρησιμοποίησα το port 2288
Και στα δύο raspberry έτρεξα έναν κοινό κώδικα.
Από οδηγίες στο site του Raspbian έβαλα το σωστό αρχείο μέσα στην κάρτα SD.
Δημιούργησα 2 threads 1 για το client και έναν για τον server στην main.
Από εκεί και πέρα δημιουργήθηκε ένα thread και για το connection ώστε να μπορούν παραπάνω από 1 client να συνδεθούν στο socket ενός server και να επικοινωνήσουν παράλληλα μαζί του..
Επίσης δημιουργήθηκε ένας κυκλικός buffer ο οποίος είναι 1 για το κάθε ενσωματωμένο και στον οποίο έχει πρόσβαση τόσο η συνάρτηση client που καλέστηκε μέσα στο thread client όσο και η συνάρτηση server που καλέστηκε μέσα στο thread server.
Έλεγξα τα μηνύματα πριν τα αποστείλω ή και κατά τη λήψη τους ώστε να τα προσθέσω ή όχι στον buffer.Ο έλεγχος ήταν σε σχέση με το αν έχουν αποσταλεί ή όχι άλλη φορά. Κάτι που εξηγώ πιο αναλυτικά στα σχόλια του κώδικα. Ουσιαστικά υλοποιήθηκε αυτό που μας ζητείται. Δηλαδή αποστέλλεται όποιο μήνυμα δεν έχει αποσταλεί σ αυτό τον παραλήπτη αλλά και λαμβάνονται μηνύματα που έχει αποστείλει η συγκεκριμένη συσκευή και μπαίνει στη λίστα.
Παρήγαγα τυχαία με random τρόπο τα μηνύματα.
Για IP χρησιμοποίησα κωδικοποιημένα τα ΑΕΜ κάποιων φοιτητών..
Τέλος φυσικά χρησιμοποιήθηκαν TCP sockets τα οποία άνοιξαν ώστε να αποσταλούν τα μηνύματα μεταξύ των ενσωματωμένων. Τα μηνύματα μετατράπηκαν σε strings ώστε να αποσταλούν μέσω των sockets.

2. Σύνδεση Rasberry

Πιο συγκεκριμένα ακολουθήθηκαν με τη σειρά οι παρακάτω εντολές

Έγιναν τα εξής βήματα:
Τα παρακάτω έγιναν 2 φορές. Μια για να συνδεθεί μέσω ssh το ένα raspberry με τον 1 υπολογιστή και μια για να συνδεθεί το άλλο raspberry με τον άλλο υπολογιστή.
Βήματα:

1.Φορτώθηκε στην κάθε μία SD κάρτα του κάθε raspberry η εικόνα του Raspbian από το site του Raspbian
2. Για την εγκατάσταση του σε μία SD κάρτα, ακολουθήθηκαν οι παρακάτω εντολές στο Terminal του Linux του κάθε laptop:

```
curl -o ESPX-rasp.tar.gz -LJ \
"https://www.dropbox.com/s/0sp5a1s6r5ee3kw/ESPX-rasp.tar.gz?dl=1"

tar xf ESPX-rasp.tar.gz

dd bs=4M if=ESPX-rasp.img of=/dev/sdX status=progress conv=fsync
```

όπου αντικαταστάθηκε το \dev\sdX με το όνομα συσκευής της SD κάρτας μου.
3. Επίσης η IP του raspberry αλλάχθηκε, διότι εάν δύο raspberry βρισκόταν εντός εμβέλειας, θα συνδεόταν μεταξύ τους και θα υπήρχε σύγκρουση των διευθύνσεών τους. Η αλλαγή της διεύθυνσης πραγματοποιήθηκε με αλλαγή του
αρχείου /etc/network/interfaces , το οποίο έγίνε απευθείας στην SD κάρτα, πριν γίνει η πρώτη σύνδεση.
4. Συνδέθηκε το 1 raspberry με ένα καλώδιο USB στο 1 laptop, άνοιξα ένα παράθυρο terminal και εκτέλεσα την εντολη "iwconfig".(sudo iwconfig wlan0 ssid pizero=ESPX) Εμφανίστηκαν κάποια ονόματα και εγώ χρησιμοποίησα το"enpYsYYuY" (όπου Y αριθμός).
5. Έπειτα άνοιξα ένα παράθυρο terminal και εκτέλεσα την ακόλουθη εντολή: "sudo nano /etc/network/interfaces" και στο παράθυρο που εμφανίστηκε:

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
```

6. Σε αυτό το παράθυρο έγραψα δίπλα από το iface:

```
iface enpYsYYuY inet static
        address 192.168.0.[2-254]
        netmask 255.0.0.0
        gateway 192.168.0.1
```

όπου:
address: η στατική διεύθυνση του laptop που πρέπει να είναι στο εύρος 192.168.0.[2-254]
netmask: 255.0.0.0

gateway: η διεύθυνση του Raspberry.Βάζουμε την default που έχει και αργότερα την αλλάξαμε.
7. Από τις ρυθμίσεις αναζήτησα την καρτέλα με όνομα "Network". Έπειτα μετά τη σύνδεση του Raspberry στον υπολογιστή πάτησα "USB Ethernet". Στις ρυθμίσεις πηγαίνοντας στην καρτέλα "IPv4", πάτησα "Manual" και έβαλα τις διευθύνσεις του προηγούμενου βήματος.
Αφού έδειξε ότι είναι "Connected", έτρεξα την εντολή ssh για να συνδεθεί(ssh 192.168.0.1 -lroot και password: espx2019)
Το τελευταίο βήμα πρέπει να επαναληφθεί κάθε φορά που αποσυνδέεται το Raspberry.

Υλοποίηση Κώδικα:

1. Χρησιμοποίησα 2 threads όπως λέω και παραπάνω
2. Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι: <stdio.h>,<sys/types.h>, <sys/socket.h>,<netinet/in.h>,<netdb.h>,<stdlib.h>,<string.h>,<unistd.h>,<arpa/inet.h>,<time.h>,<sys/time.h>,<pthread.h>,<sys/time.h>,<sys/sysinfo.h>,<inttypes.h>,<signal.h>
3. Στο αρχείο Statistics.txt αποθήκευσα τα αποτελέσματα.
4. Χρησιμοποίησα το Port 2288
5. Χρησιμοποίησα τον πίνακα από strings *circ_buffer[2000] που όρισα καθολικά για την υλοποίηση του κυκλικού buffer που θα έχει το κάθε ενσωματωμένο ώστε το κάθε string είναι ένα μήνυμα και είναι και το ίδιο ένας μονοδιάστατος πίνακας-διάνυσμα από chars.
6. Όρισα τα threads όπως και τα mutex,lock και cond
7. Χρησιμοποίησα τις εξής συναρτήσεις:
8. int main(int argc,char *argv[])
   Η main είναι υπεύθυνη για τη δημιουργία threads μέσω της pthread_create και μέσα στο καθένα από αυτά δημιουργείται socket με το οποίο γίνεται η σύνδεση.Επίσης χρησιμοποιείται καταγραφή στατιστικών στοιχείων(διεύθυνση αποστολής, χρόνος σύνδεσης των Raspberry κτλ).Τα threads εκτελούνται με τη σειρά.
9. void*client()
   Η συνάρτηση αυτή αποστέλει τα μηνύματα στο Raspberry.
10. void*connection(void*socket)

Αυτή τη συνάρτηση την χρησιμοποιώ για να ελέγξω για διπλότυπα μηνύματα κάνοντας τους απαραίτητους ελέγχους που ζητούνται.

11.char*random message ()

Η random message () δημιουργεί τυχαία μηνύματα που θα σταλούν μέσω του socket

12.int *random_time()

Δημιουργεί ένα τυχαίο χρονικό διάστημα μέσα στο οποίο θα δημιουργηθεί τυχαίο μήνυμα.

13.int circ_bbuf_pop(char **data1) και int circ_bbuf_push(char **data1)

Με τις συναρτήσεις αυτές παραπάνω γίνεται η αποθήκευση των νέων μηνυμάτων που έρχονται όπως και η διαγραφή των προηγούμενων αν η λίστα είναι γεμάτη και θέλουμε να αποθηκεύσουμε ένα μήνυμα.

15. void Statistics (int sockfd , char*IP, socklen_t client_length,struct sockaddr_in*c)

Εδώ καταγράφω τα στατιστικά στοιχεία σε ένα αρχείο txt.