aΣΥΣΤΗΜΑΤΑ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ

Στο παρών κείμενο θα εξηγηθεί αναλυτικά τόσο ο τρόπος αντιμετώπισης του προβλήματος όσο και ο τρόπος διασύνδεσης των raspberry με τους υπολογιστές

Καθώς και το σκεπτικό πίσω από τη γραφή του κώδικα.

Αναλυτικότερα:

1. Τρόπος Αντιμετώπισης προβλήματος

Το όλο σύστημα έτρεξε σε περιβάλλον-λειτουργικό σύστημα Linux-Ubuntu.

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 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

Αυτό έγινε διότι και στα δύο raspberry έτρεξα έναν κοινό κώδικα ο οποίο αναπαριστά το μοντέλο peer to peer και κάνει το κάθε raspberry τόσο client όσο και server..

Για να γίνει αυτό ακολούθησα πρώτα τις εντολές του εγγράφου που μας δόθηκε στο e-learning από τους διδάσκοντες του μαθήματος αλλά και από το αντίστοιχο site του raspbian.Ο τρόπος θα αναλυθεί παρακάτω.

Όσον αφορά τον κώδικα δημιουργήθηκαν 2 threads 1 για το client και έναν για τον server και αυτά δημιουργήθηκαν 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.

Επίσης έγιναν οι κατάλληλοι έλεγχοι πριν την αποστολή και λήψη ενός μηνύματος από ένα ενσωματωμένο ώστε να προστεθεί ή όχι στον buffer ανάλογα με τις συνθήκες ελέγχου που μας ζητούνται στην εκφώνηση

(Αποστέλλονται όλα τα μηνύματα που δεν έχουν σταλεί στον συγκεκριμένο παραλήπτη στο παρελθόν 2. Λαμβάνονται όλα τα μηνύματα που έχει να αποστείλει η νέα συσκευή και αποθηκεύονται στην λίστα, αφού πρώτα γίνει έλεγχος για διπλότυπα (όλα τα πεδία πρέπει να είναι ίδια για να χαρακτηριστεί μία εγγραφή διπλότυπη))

Τέλος φυσικά χρησιμοποιήθηκαν TCP sockets τα οποία άνοιξαν ώστε να αποσταλούν τα μηνύματα μεταξύ των ενσωματωμένων.

Ως ip δόθηκαν κωδικοποιημένα τα ΑΕΜ κάποιων φοιτητών όσο και το δικό μου τα οποία δόθηκαν στις κατάλληλες μεταβλητές.

Στο κάθε μήνυμα τα αντίστοιχα πεδία αφού συμπληρώθηκαν μετατράπηκαν σε strings ώστε να αποσταλούν μέσω των sockets.

Σαφώς επίσης τα μηνύματα παράχθηκαν τυχαία με random συναρτήσεις και τρόπο που θα αναλυθεί στην αντίστοιχη ενότητα.

1. 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 το"enpXsXXuX" (όπου Χ αριθμός) που μου ενδιέφερε το χρησιμοποίησα.

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 enpXsXXuX 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. Μετά άνοιξα τις ρυθμίσεις και έψαξα την καρτέλα που λέει "Network". Αφού συνδέθηκε το Raspberry στον υπολογιστή έβγαλε μια επιλογή που έλεγε "USB Ethernet". Άνοιξα τις ρυθμίσεις, στην καρτέλα που λέει "IPv4", επέλεξα "Manual" και εισήγαγα στα αντίστοιχα πλαίσια τις διευθύνσεις που όρισα στο προηγούμενο βήμα. Μόλις μου έδειξε "Connected", έκανα SSH και συνδέθηκε, εννοώντας εκτέλεσα την εντολή ssh και συνδέθηκε.( ssh 192.168.0.1 -lroot και password: espx2019)

Το τελευταίο βήμα επαναλαμβάνεται κάθε φορά που αποσυνδέεται το Raspberry και γίνεται προσπάθεια να επανασυνδεθεί.

Υλοποίηση Κώδικα:

Χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

1. <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>
2. Χρησιμοποιήθηκαν 4 threads όπως λέω και παραπάνω
3. Χρησιμοποιήθηκε το port 2288 το οποίο γράφω και παραπάνω το οποίο το όρισα με #define PORT 2288
4. Χρησιμοποιήθηκε το αρχείο Statistics.txt για να αποθηκευτούν τα αποτελέσματα
5. Χρησιμοποιήθηκε η δομή circ\_bbuf\_t που ορίστηκε καθολικά για την υλοποίηση του κυκλικού buffer που θα έχει το κάθε ενσωματωμένο ώστε να αποθηκευτούν σε αυτόν τα μηνύματα και μέσα σε αυτόν ορίστηκε ο buffer ως pointer σε pointer(\*\*) ώστε να αποτελέσει έναν πίνακα από strings που το κάθε string είναι ένα μήνυμα και είναι και το ίδιο ένας μονοδιάστατος πίνακας-διάνυσμα από chars.
6. Ορίστηκαν τα threads καθολικά αλλά και τα mutex,lock και cond όπως και ο αριθμός των threads και στον τύπο της δομής circ\_buff\_t ορίστηκε ένας pointer \*b(αντικείμενο b)