Ενσωματωμένα Συστήματα Πραγματικού Χρόνου

Τελική Εργασία

Συντάκτης: Κωνσταντινίδης Κωνσταντίνος, ΑΕΜ : 9162,

Στοιχεία επικοινωνίας: konkonstantinidis@ece.auth.gr

Α. Περιγραφή του τρόπου υλοποίησης

Παρακάτω θα εξηγηθεί, με όσες λεπτομέρειες κρίνεται πως χρειάζονται και τηρώντας τη σειρά εμφάνισης, ο τρόπος λειτουργίας και η χρήση των συναρτήσεων στο αρχείο covidTrace.h, το οποίο περιλαμβάνει και όλο το κώδικα που αναπτύχθηκε για την παρούσα εργασία.

Αρχικά, ορίζονται τα παρακάτω macro ως εξής:

- COVID_CHANCE: Η πιθανότητα (%) να επιστραφεί θετικό τεστ από την testCOVID(), για την εργασία ορίστηκε στο 1% ώστε να έχουμε μερικά θετικά τεστ κατά τη διάρκεια εκτέλεσης του προγράμματος.
- MAC_ARRAY_SIZE: Το μέγεθος του πίνακα τον οποίο θα γεμίσουμε στην αρχή του προγράμματος με τυχαίως παραγόμενες MAC διευθύνσεις, για την εργασία ορίστηκε στο 100 ώστε να υπάρχει μία πληθώρα πιθανών (κοντινών) επαφών.
- RUNTIME_NSECS: Ο χρόνος που θα εκτελείται το πρόγραμμα σε nanoseconds, έχει οριστεί στις 7 ώρες και 12 λεπτά = 25920000000000 nanoseconds.

Έπειτα, έχουμε τον ορισμό των παρακάτω struct:

- MACaddress: Δομή μεγέθους 48-bit που υλοποιεί μία διεύθυνση MAC, διαθέτει έναν πίνακα τύπου uint8_t με έξι θέσεις, αποθηκεύοντας έτσι μία διεύθυνση MAC 48-bit.
- contact: Δομή μεγέθους 16 byte που υλοποιεί μία επαφή, διαθέτει μία μεταβλητή τύπου MACaddress για την καταχώρηση της διεύθυνσης MAC της επαφής και μία μεταβλητή τύπου timeval για τη καταχώρηση της χρονικής στιγμής που συναντήθηκε η επαφή.

Επιπλέον, έχουμε την αρχικοποίηση των παρακάτω global μεταβλητών:

- contacts[120]: Ένας στατικός πίνακας τύπου “struct contact”, όπου αποθηκεύονται οι επαφές που επιστρέφονται από τη BTnearMe() κατά τη διάρκεια εκτέλεσης του προγράμματος. Ο πίνακας έχει αρχικοποιηθεί στο

μέγιστο (και τελικό) του μέγεθος των 120 θέσεων, καθώς ξέρουμε ότι κάθε αναζήτηση που γίνεται ανά 0.1 δευτερόλεπτα επιστρέφει μία MAC διεύθυνση - επαφή, και θέλουμε να αποθηκεύουμε τις επαφές που δεν είναι κοντινές για 12 δευτερόλεπτα και μετά να διαγράφονται, επομένως σε οποιαδήποτε χρονική στιγμή θα έχουμε μέγιστα 12/0.1 = 120 επαφές που πρέπει να 'θυμόμαστε'.

- contacts_writeIndex: Ακέραια μεταβλητή που δηλώνει σε ποιο σημείο του πίνακα contacts έχει φτάσει η εγγραφή επαφών (μόνο για την αρχή του προγράμματος, μετά θα είναι συνεχώς γεμάτος και η τιμή της μεταβλητής αυτής θα μένει σταθερή στο 119).
- threadBTnearMe_sleepTime, threadTestCOVID_sleepTime,: Μεταβλητές μέτρησης χρόνου τύπου "struct timespec", που είναι απαραίτητες για την λειτουργία της nanosleep() (βλ. παρακάτω). Έχουν δύο πεδία, το tv_sec και το tv_nsec, που είναι ο χρόνος σε δευτερόλεπτα και τα νάνο-δευτερόλεπτα επιπλέον από το τελευταίο δευτερόλεπτο.
- MACsArray: Μεταβλητή δείκτης σε πίνακα τύπου "struct MACaddress", στον οποίο αποθηκεύονται οι τυχαίως παραγόμενες μεταβλητές.
- contacts_mutex: Ένα mutex για τη πρόσβαση στον πίνακα contacts, αναγκαίο καθώς στον πίνακα αυτό χρειάζονται πρόσβαση δύο νήματα που θα τρέχουν παράλληλα

Τέλος, ορίζονται οι παρακάτω συναρτήσεις:

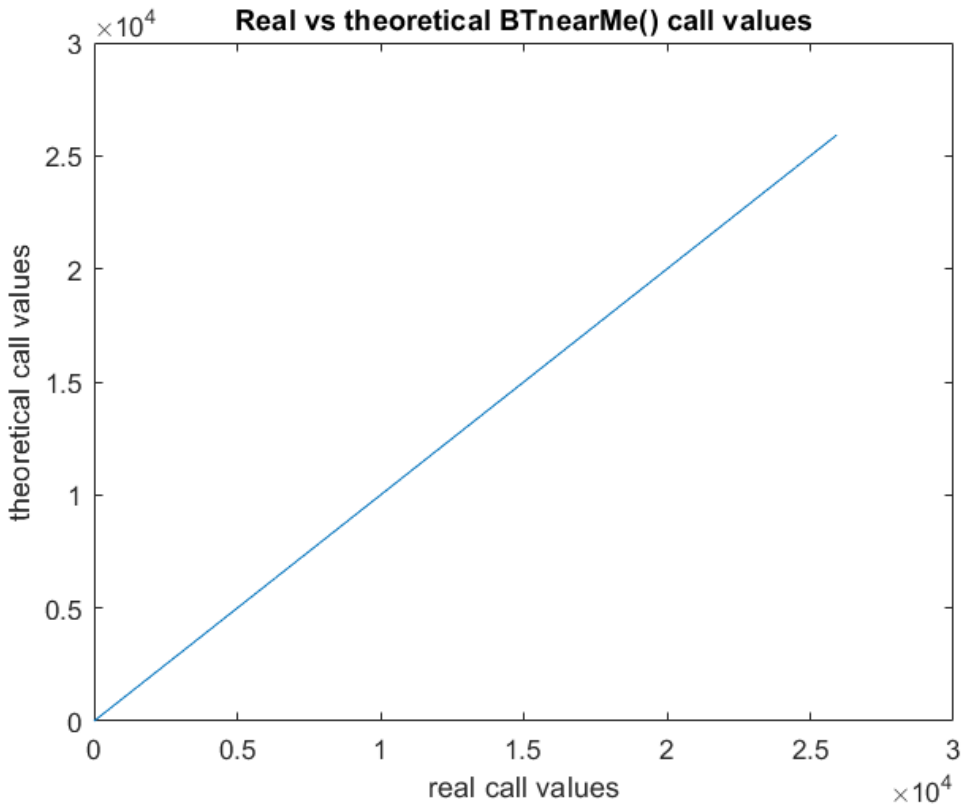
- void initializeMACs(): Γεμίζει τον πίνακα MACsArray με MAC_ARRAY_SIZE τυχαίες διευθύνσεις MAC, που παράγει εκείνη την ώρα
- struct MACaddress BTnearMe(): Καταγράφει την ώρα που καλέστηκε στο αρχείο "BTcalls.bin" και επιστρέφει μία τυχαία MAC διεύθυνση από τον MACsArray
- void addToContacts(struct contact newContact): Προσθέτει την επαφή newContact στον πίνακα contacts, σύμφωνα με το contacts_writeIndex, εάν αυτό είναι μικρότερο ή ίσο του 119 (δηλαδή ο πίνακας contacts δεν έχει γεμίσει ακόμη επειδή είμαστε στα πρώτα δευτερόλεπτα της εκτέλεσης του προγράμματος). Εάν το contacts_writeIndex είναι ίσο με 120, ο πίνακας contacts είναι γεμάτος, άρα έχουν περάσει 12 δευτερόλεπτα και πλέον για κάθε επαφή που προσθέτουμε στον πίνακα, πρέπει να διαγράφουμε και την παλαιότερη (εφόσον θα έχουν περάσει 12 δευτερόλεπτα από τη στιγμή που τη συναντήσαμε και δεν χρειάζεται να τη θυμόμαστε άλλο). Αυτό γίνεται κάνοντας αρχικά μία 'ολίσθηση' μία θέση πίσω όλες τις επαφές του πίνακα (η $2^{η}$ επαφή θα γίνει η $1^{η}$ διαγράφοντας την προηγούμενη πρώτη/παλαιότερη, η $25^{η}$ θα γίνει $24^{η}$ και η $120^{η}$ θα γραφτεί δύο φορές, στην τελευταία και προτελευταία θέση του πίνακα). Έπειτα, γράφουμε στο τέλος του πίνακα (θέση 119) την newContact.
- void addToTimespec(struct timespec* t_spec, struct timespec addend): Προσθέτει στο πρώτο struct timespec όρισμα το δεύτερο.
- void *threadBTnearMe(): Η συνάρτηση που καλείται σα pthread, αναλαμβάνει τις κλήσεις της BTnearMe() και διεκπεραιώνει όποιες επιπλέον

διεργασίες συνεπάγεται αυτό. Τρέχει διαρκώς μέχρι να τερματίσει τον εαυτό της, όταν περάσουν RUNTIME_USECS δευτερόλεπτα χρησιμοποιώντας την clock_nanosleep() για την διαλλειματική λειτουργία της.

- bool testCOVID(): Επιστρέφει 0 (αρνητικό) ή 1 (θετικό) αποτέλεσμα για το τεστ της ασθένειας, με COVID_CHANCE % πιθανότητα να επιστρέψει θετικό αποτέλεσμα και (1 - COVID_CHANCE)% πιθανότητα να επιστρέφει αρνητικό αποτέλεσμα.
- void uploadContacts(struct MACaddress* closeContacts, int arraySize): Τυπώνει στο αρχείο "closeContacts.txt" τα περιεχόμενα του πίνακα closeContacts, μεγέθους arraySize.
- bool equalMACs(struct MACaddress mac1, struct MACaddress mac2): Επιστρέφει 1 αν οι δύο διευθύνσεις MAC mac1 και mac2 είναι ίδιες, αλλιώς επιστρέφει 0.
- void printContactsArray(): Τυπώνει το περιεχόμενο του πίνακα contacts στο αρχείο "allContacts.txt", για περεταίρω επιβεβαίωση της καλής λειτουργίας του προγράμματος κατόπιν της εκτέλεσης του
- void *threadTestCOVID(): Η συνάρτηση που καλείται σα pthread, αναλαμβάνει τις κλήσεις της testCOVID() και διεκπεραιώνει όποιες επιπλέον διεργασίες συνεπάγεται αυτό (κυριότερα, σε περίπτωση θετικού αποτελέσματος εξάγει από τον contacts τις κοντινές επαφές και καλεί τις αντίστοιχες συναρτήσεις για την παραγωγή των αρχείων). Τρέχει διαρκώς μέχρι να τερματίσει τον εαυτό της, όταν περάσουν RUNTIME_NSECS δευτερόλεπτα, χρησιμοποιώντας την clock_nanosleep() για την διαλλειματική λειτουργία της.
- int programFunction(): Τέλος, η συνάρτηση που καλείται από τη main για την εκτέλεση της λειτουργίας όλου του προγράμματος. Κυριότερα, καλεί την initializeMACs() για την υλοιποίηση των τυχαίως παραγόμενων MAC διευθύνσεων, αρχικοποιεί το contacts_mutex, δημιουργεί τα δύο pthreads με τις συναρτήσεις *threadBTnearMe() και *threadTestCOVID() και περιμένει τη λήξη τους για να καταστρέψει το contacts_mutex και να ολοκληρώσει το πρόγραμμα.

Β. Διάγραμμα διαφοράς χρόνων κλήσεων της BTnearMe()

Παρακάτω παρατίθεται το διάγραμμα με τους πραγματικούς χρόνους κλήσεων της BTnearMe() στον Χ άξονα, σε σχέση με τους θεωρητικούς χρόνους που θα έπρεπε να κληθεί η BTnearMe() στον Υ άξονα.



Όπως φαίνεται και από το διάγραμμα, υπάρχει πρακτικά πλήρης ταύτιση των χρόνων. Η απειροελάχιστη απόκλιση οφείλεται κατά κύριο (και ενδεχομένως μόνο) λόγο στην ανακρίβεια της clock_nanosleep(), η οποία πειραματικά μετρήθηκε στο +0.001%, για χρόνο ύπνου των 0.1 δευτερολέπτων.

Γ. Σχόλια και παρατηρήσεις

Όσον αφορά τη χρήση της CPU κατά την εκτέλεση του προγράμματος, χρησιμοποιώντας την εντολή vmstat αμέσως μετά το τέλος της εκτέλεσης του προγράμματος (και έχοντας κάνει boot αμέσως πριν την αρχή της εκτέλεσης), βρέθηκε πως το πρόγραμμα κατά την εκτέλεση του χρησιμοποίησε κατά μέσο όρο 0% της CPU. Επειδή η vmstat στρογγυλοποιεί στην κοντινότερη μονάδα, αυτό σημαίνει ότι η χρήση της CPU ήταν στο διάστημα (0,0.5%). Για να το επιβεβαιώσω, έτρεξε το πρόγραμμα και στο official image και εκεί χρησιμοποίησα με τον ίδιο τρόπο την iostat εντολή του πακέτου sysstat, η οποία έχει ακρίβεια δύο δεκαδικών ψηφίων. Πράγματι, έδειξε πως η μέση χρήση της CPU από διεργασίες χρήστη ήταν στο 0.31% κατά την εκτέλεση του προγράμματος.

Επομένως η CPU ήταν αδρανής για το 99.69% του χρόνου εκτέλεσης του προγράμματος, κατά μέσο όρο.

Το αρχείο covidTrace.h, μαζί με τα παραγόμενα αρχεία .bin και .txt, καθώς και το MATLAB script για την παραγωγή του διαγράμματος, θα τα βρείτε στο παρακάτω link:

https://drive.google.com/drive/folders/1em9RJcZslFmHD-q-Ly9PQJ1Vbk20Q9Yp?usp=sharing