

Εργασία στα Ενσωματωμένα Συστήματα Πραγματικού Χρόνου

ΕΡΓΑΣΙΑ ΕΞΑΜΗΝΟΥ
ΝΙΚΟΛΑΟΣ ΓΙΑΝΝΟΠΟΥΛΟΣ
ΑΕΜ: 9629
NGIANNOP@ECE.AUTH.GR

Εισαγωγή

Στην παρούσα εργασία θα υλοποιήσουμε μια δομή που λειτουργεί σε πραγματικό χρόνο για το ενσωματωμένο σύστημα Raspberry Pi Zero που αφορά συναλλαγές μεταξύ μέτοχων που παρέχονται δωρεάν από το finnhub.io. Όλοι η υλοποίηση της εργασίας βασίστηκε στην βιβλιοθήκη [libwebsockets](https://github.com/strawlab/libwebsockets) που παρέχει υπηρεσίες για την λήψη των μηνυμάτων από την ισότοπο finnhub.io.

Στρατηγική

Η στρατηγική που ακολούθησα για την υλοποίησή της εργασίας μια δομή ουράς (όπως στο πρώτο παραδοτέο) για να τοποθετώ της συναλλαγές που συμβαίνουν την στιγμή που θα συμβούν και να γράφονται στα ανάλογα αρχεία (SYMBOL_NAME/SYMBOL_NAME_trading_info.txt) ώστε να μην χαθεί καμία από αυτές. Καθώς έχουμε στην διάθεση μας τόσα νήματα όσα είναι και οι θέσεις της ουράς για να είμαστε βέβαιοι ότι δεν θα υπάρξει απώλεια δεδομένων.

Για την υλοποίηση που αφορά το κομμάτι του Candlesticks καθώς και για τον μέσω ορώ τιμών αλλά και όγκου χρησιμοποίησα ένα array σταθερού μεγέθους με μια δομή τύπου tradingInfo οπού αυτό το struct περιχέει

1. Float price
2. Char symbol[40]
3. Unsigned long int timestrap
4. Float volume

αρά και όλη υλοποίηση του array αυτό αποτελείται από BUFFER_SIZE = 10.000 θέσεις τύπο struct tradingInfo η συνολική δέσμευση μνήμης είναι περίπου ίσο με 10,485,760 bits => 1.31072 MB οπού η μνήμη του Raspberry Pi που είναι διαθέσιμη είναι 512MB και το array αυτό για να είναι πιο γρήγορο χωρίς να υπάρχει κάποιο deadlock γίνεται με την λογική της κυκλικής ουράς.

Επιπλέον καθώς υπάρξει συναλλαγή τότε θα δημιουργεί ένα thread για την συγκεκριμένη συναλλαγή και για τον υπολογισμό του Candlesticks και των μέσο ορών.

Έτσι με την συνολική εικόνα τις στατικής φαίνεται να μην δαπανά πόρους καθώς δεν υπάρχουν συναλλαγές και επιπλέον είναι σίγουρο ότι δεν θα χαθεί καμία πληροφορία μιας που το μέγεθος του array είναι αρκετά μεγάλο ώστε έτσι δεν θα υπάρξει κάποιο overwrite σε πρόσφατα αποθηκευμένα δεδομένα.

Τα μηνύματα που λαμβάνουμε έρχονται με μια προκαθορισμένη σειρά

1. Price
2. Symbol
3. Timestrap
4. Volume

Έτσι κάθε φορά που ο parser καλείται η συνάρτηση cb αποτελείται από ένα switch/case όπου αφού λάβουμε όλη την πληροφορία τότε προσθέτουμε στην ουρά τα δεδομένα ώστε να δημιουργηθεί η εγγραφή τι στιγμή που μας έφτασε.

Παρουσίαση/Ανάλυση

```
void *saveCandleSticksAndAvarage()
{
    static int candleStickArrayFull = 0;

    static unsigned int tail = 0;

    static unsigned int candleSticksIndex = 0;

    struct tradingInfo data;

    struct candleStick candleSticks[4 * 15];

    struct tradingInfo finalTradingInfos[4];
```

...

Στην ρουτίνα αυτή το candleSticks[4*15] έχουμε ένα μονοδιάστατο array για κάθε σύμβολο να περιέχει 15 θέσεις ώστε να

μπορεί να υπολογιστή το candlestick και ο κινούμενο μέσος ορός τιμών και όγκος

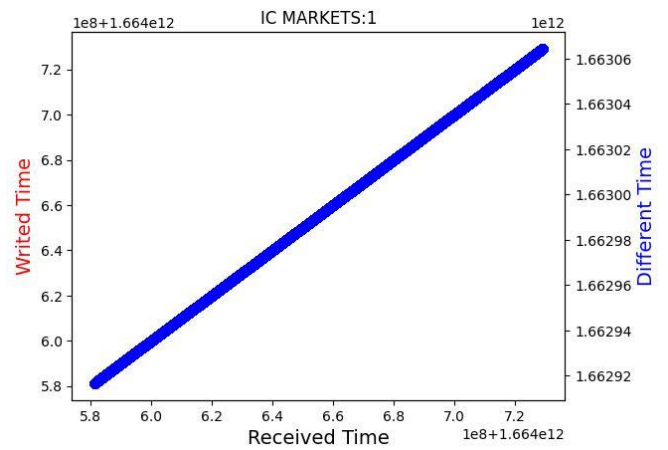
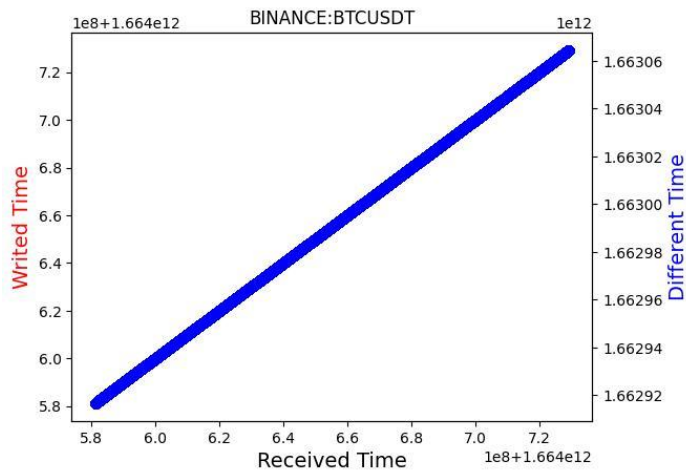
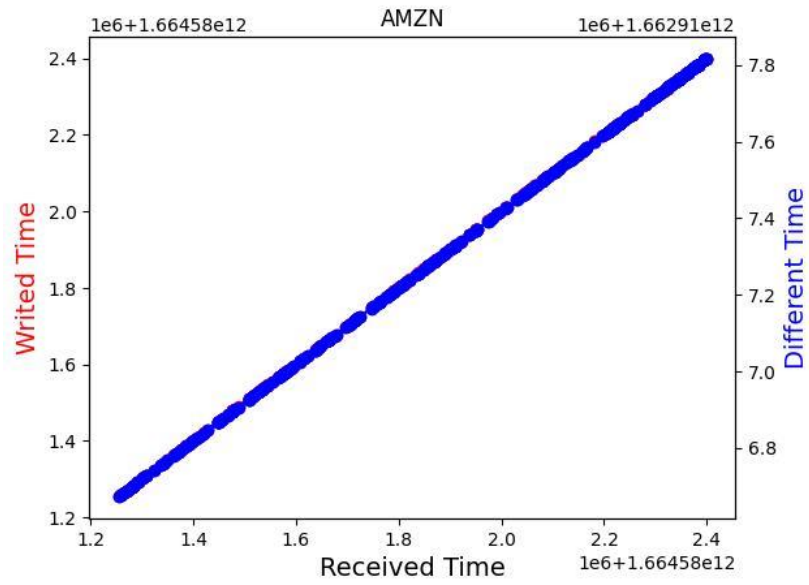
Για τον έλεγχο της συνδέσεις και για την συνεχή εκτέλεσης του προγράμματος χρησιμοποιώ ένα while το οποίο κάθε φορά αφού τερματίσει η σύνδεση ξανά ξεκινάει μια νέα καινούργια

```
while(1){
    ...      // this is a sample not the completed code!
    ...
    urlPath[0] = '/';
    strncpy(urlPath + 1, urlTempPath, sizeof(urlPath) - 2);
    urlPath[sizeof(urlPath) - 1] = '\0';
    clientConnectionInfo.port = 443;
    clientConnectionInfo.path = urlPath;
    clientConnectionInfo.ssl_connection = LCCSCF_USE_SSL |
LCCSCF_ALLOW_SELF_SIGNED | LCCSCF_SKIP_SERVER_CERT_HOSTNAME_CHECK;

    clientConnectionInfo.host = clientConnectionInfo.address;
    clientConnectionInfo.origin = clientConnectionInfo.address;
    clientConnectionInfo.ietf_version_or_minus_one = -1;
    clientConnectionInfo.protocol = protocols[0].name;
    printf("Testing %s\n\n", clientConnectionInfo.address);
    printf("Connecting to %s://%s:%d%s \n\n", urlProtocol,
        clientConnectionInfo.address, clientConnectionInfo.port, urlPath);
    wsi = lws_client_connect_via_info(&clientConnectionInfo);
    while (wsi == NULL)
    {
        printf(KRED "[Main] wsi create error.\n" RESET);
        printf("Try to connect..\n");
        wsi = lws_client_connect_via_info(&clientConnectionInfo);
    }
}
```

Επιπλέον στο δεύτερο εσωτερικό while ελέγχει την σύνδεση και στην περίπτωση που το δίκτυο έχει πέσει τότε περιμένει εκεί μέχρι να επανέλθει και να συνεχίσει.

Στις παρακάτω εικόνες φαίνεται η διαφορά μεταξύ Εγγραφής & παραλαβή της πληροφορίας και η διαφορά τους.



[Google Drive](#)