Πανεπιστήμιο Δυτικής Αττικής
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΕΡΓΑΣΤΗΡΙΟ ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΕΡΓΑΣΙΑ 1.

ΗΜΕΡΟΜΗΝΙΑ ΠΑΡΑΔΟΣΗΣ:

Κυριακή 12 Μαΐου 2024-11.59μ.μ

ΟΜΑΔΑ ΕΡΓΑΣΤΗΡΙΟΥ:

Τμ.ΣΤ4-Α(Τετ 4-6)

Υπευθύνος Ομάδας:

ΙΟΡΔΑΝΑΚΗΣ ΜΙΧΑΛΗΣ

ΣΥΝΤΑΚΤΗΣ:

ΦΙΛΙΠΠΟΣ ΠΑΠΑΓΕΩΡΓΙΟΥ

ΑΜ:21390174

Τι θα δούμε στην Εργασία:

Η εργασία παρουσιάζει τον σχεδιασμό και την υλοποίηση ενός πολυνηματικού διακομιστή (server) και ενός συστήματος RPC (Remote Procedure Call) σε γλώσσα προγραμματισμού C. Ο σκοπός της εργασίας είναι η δημιουργία ενός συστήματος εξυπηρέτησης που μπορεί να εκτελεί τρεις διαφορετικούς υπολογισμούς με βάση τις εισόδους του χρήστη. Οι υπολογισμοί περιλαμβάνουν τον υπολογισμό της μέσης τιμής ενός διανύσματος, τον εντοπισμό των μέγιστων και ελάχιστων τιμών ενός διανύσματος, καθώς και το γινόμενο ενός αριθμού με ένα διάνυσμα.

Η εργασία παρέχει επίσης οδηγίες για τον τρόπο αλληλεπίδρασης του πελάτη με τον εξυπηρετητή μέσω sockets TCP/IP, καθώς και τον σχεδιασμό και την υλοποίηση του διεπικοινωνιακού πρωτοκόλλου RPC για την απομακρυσμένη εκτέλεση των υπολογισμών.

Τέλος, παρουσιάζεται ο κώδικας του πελάτη (client) που δίνει στο χρήστη τη δυνατότητα να επιλέξει τον επιθυμητό υπολογισμό και να λάβει τα αντίστοιχα αποτελέσματα από τον εξυπηρετητή.

Υ.Χ :

Επίσης για την δημιουργία και εκτέλεση της εργασίας χρησιμοποιήσα το

SEED Ubuntu 16.04 που χρησιμοποιήσαμε στο εργαστήριο ‘ΑΣΦΑΛΕΙΑ ΣΤΗΝ ΤΕΧΝΟΛΓΙΑ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ’ δίοτι το rpcgen δεν έτρεχε στο linux arch που είχα κατεβασμένο.

1.Όρισμος του πρωτοκόλλου RPC . Αύτο περιλαμβάνει την δημιουργία του idl.x αρχείου που ορίζονται οι δομές δεδομένων και οι συναρτησείς που θα χρησιμοποιηθούν άπο τον server και client.

```
struct intvector {
    int n;
    int vector<>;
};

struct intpair {
    int min;
    int max;
};


struct product{
    float a;
    intvector vector<>;
};

struct floatvector{
    float elements<>;
};


program INTERFACE_PROG{
    version INTERFACE_VERS{
        float calculate_average(intvector) = 1;
        intpair find_max_min(intvector) = 2;
        floatvector multiply_scalar(product) = 3;
    }=1;
}=0x23451111;
```

Δομές Δεδομένων

1. struct intvector**

- Αυτή η δομή αποθηκεύει ένα διάνυσμα ακεραίων.

- `int n;`: Αποθηκεύει τον αριθμό των στοιχείων στο διάνυσμα.

- `int vector<>;`: Ένας δυναμικός πίνακας που περιέχει τα στοιχεία του διανύσματος. Το `< >` σημαίνει ότι θα χρησιμοποιηθεί ένας μηχανισμός δυναμικής διάστασης στον RPC.

2. **struct intpair**

- Χρησιμοποιείται για την επιστροφή δύο τιμών (ελάχιστη και μέγιστη) από μια διαδικασία.

- `int min;` και `int max;`: Οι τιμές για το ελάχιστο και το μέγιστο του διανύσματος.

3. **struct product**

- Περιέχει ένα πραγματικό αριθμό και ένα διάνυσμα για υπολογισμούς προϊόντων.

- `float a;`: Ο συντελεστής πολλαπλασιασμού.

- `intvector vector<>;`: Το διάνυσμα που θα πολλαπλασιαστεί.

4. **struct floatvector**

- Αντιστοιχίζεται σε ένα διάνυσμα πραγματικών αριθμών.

- `float elements<>;`: Πίνακας πραγματικών αριθμών για αποθήκευση αποτελεσμάτων προϊόντων.

Διαδικασίες RPC

Ορίζονται τρεις διαδικασίες στο πρόγραμμα `INTERFACE_PROG` της έκδοσης `INTERFACE_VERS`:

- **float calculate_average(intvector) = 1;**

- Υπολογίζει τη μέση τιμή του δοθέντος διανύσματος.

- **intpair find_max_min(intvector) = 2;**

- Επιστρέφει

τις τιμές μέγιστου και ελάχιστου για το δοθέν διάνυσμα.

- **floatvector multiply_scalar(product) = 3;**

- Υπολογίζει το γινόμενο ενός πραγματικού αριθμού και ενός διανύσματος και επιστρέφει το αποτέλεσμα ως ένα διάνυσμα πραγματικών αριθμών.

Κάθε διαδικασία αναγνωρίζεται με έναν μοναδικό αριθμό (π.χ., `1`, `2`, `3`) για την κλήση μέσω του RPC. Η έκδοση `INTERFACE_VERS` ορίζεται ως `1` και ο αναγνωριστικός αριθμός του προγράμματος είναι `0x23451111`, προσδίδοντας μοναδικότητα στο πρωτόκολλο επικοινωνίας στο πλαίσιο του δικτύου.

Στην συνέχεια με την βοήθεια του rpc δημιουργηθηκαν τα template

Του server και του client που θα πρέπει να γραψουμε τον κωδικά για να δουλέψει η ασκήση client – server , οι κωδικές είναι ανεβασμένοι σε zip αρχείο . Παρακάτω θα εξηγησω την σκέψη μου για τους κωδικές και θα δούμε ενδεικτικά παραδείγματα.

2.SERVER

Ωραία, ας ξεκινήσουμε με την τεκμηρίωση του κώδικα του διακομιστή που παρέχετε. Ακολουθεί μια απλή ανάλυση των λειτουργιών του διακομιστή για το έγγραφό σας στο Word:

Αρχεία: idl_server.c

Αυτή η υλοποίηση server παρέχει τρεις βασικές υπηρεσίες για την επεξεργασία αριθμητικών δεδομένων που λαμβάνονται από πελάτες μέσω κλήσεων RPC. Κάθε συνάρτηση είναι υπεύθυνη για την εκτέλεση συγκεκριμένων μαθηματικών πράξεων σε πίνακες ακεραίων αριθμών.

1. Υπολογισμός μέσου όρου

Συνάρτηση:

float *calculate_average_1_svc(intvector *argp, struct svc_req *rqstp)

Περιγραφή:

Αυτή η συνάρτηση υπολογίζει το μέσο όρο ενός πίνακα ακεραίων αριθμών. Επαναλαμβάνει τον πίνακα για να αθροίσει τα στοιχεία του και στη συνέχεια διαιρεί με τον αριθμό των στοιχείων για να βρει τον μέσο όρο.

Παράμετροι:

- `intvector *argp`: Ένας δείκτης σε μια δομή `intvector` που περιέχει τον πίνακα και το μέγεθός του.

Επιστρέφει:

- Επιστρέφει έναν δείκτη σε ένα στατικό float που περιέχει τον υπολογισμένο μέσο όρο.

2. Βρείτε το μέγιστο και το ελάχιστο

Λειτουργία:

intpair *find_max_min_1_svc(intvector *argp, struct svc_req *rqstp)

Περιγραφή:

Προσδιορίζει τη μέγιστη και την ελάχιστη τιμή σε έναν πίνακα ακεραίων αριθμών. Διέρχεται επαναληπτικά από τον πίνακα, ενημερώνοντας τις μέγιστες και ελάχιστες τιμές καθώς προχωρά.

Παράμετροι:

intvector *argp: Ο πίνακας ακέραιων αριθμών.

struct svc_req *rqstp: Μια δομή αίτησης υπηρεσίας (δεν χρησιμοποιείται στον υπολογισμό).

Επιστρέφει:

Επιστρέφει έναν δείκτη σε μια στατική δομή intpair που περιέχει τις ελάχιστες και μέγιστες τιμές που βρέθηκαν.

3. Πολλαπλασιάστε το Scalar

Λειτουργία:

floatvector *multiply_scalar_1_svc(product *argp, struct svc_req *rqstp)

Περιγραφή:

Πολλαπλασιάζει κάθε στοιχείο ενός ακέραιου πίνακα με μια float τιμή κινητής υποδιαστολής. Η συνάρτηση δεσμεύει δυναμικά μνήμη για τον πίνακα που προκύπτει και διαχειρίζεται τη μνήμη από προηγούμενες κλήσεις.

Παράμετροι:

*argp: Μια δομή που περιέχει την κλίμακα (float a) και τον πίνακα ακέραιων αριθμών (intvector vector).

Επιστρέφει:

Επιστρέφει έναν δείκτη σε ένα διάνυσμα floatvector που περιέχει τον πινακα και το μεγεθος στον πολλ/μενο πινακα. Εάν η κατανομή μνήμης αποτύχει, επιστρέφει NULL.

Διαχείριση μνήμης:

Αποδεσμεύει τη μνήμη που είχε προηγουμένως κατανεμηθεί για την αποφυγή διαρροών και ανακατανέμει για την τρέχουσα κλήση.

3.CLIENT

Ας τεκμηριώσουμε τον κώδικα του client που αλληλεπιδρά με τον server RPC. Παρακάτω παρουσιάζεται μια ανάλυση των λειτουργιών που παρέχονται από την client-server:

Αρχείο: idl_client.c

Το πρόγραμμαεπιτρέπει στους χρήστες να αλληλεπιδρούν με έναν server RPC για την εκτέλεση υπολογισμών σε πίνακες ακεραίων. παρέχει επιλογές για τον υπολογισμό του μέσου όρου, την εύρεση της μέγιστης και της ελάχιστης τιμής και την εκτέλεση με πολλαπλασιασμού σε έναν πίνακα ακεραίων αριθμών.

Εγκατάσταση και αρχικοποίηση

Ο πελάτης εγκαθιστά μια σύνδεση με τον διακομιστή RPC χρησιμοποιώντας TCP. Επαληθεύει τη σύνδεση και εξέρχεται εάν η σύνδεση δεν μπορεί να δημιουργηθεί

Αλληλεπίδραση χρήστη

Ο πελάτης ζητά από το χρήστη να εισάγει έναν πίνακα ακεραίων αριθμών, παρέχοντας επιλογές για την εκτέλεση διαφόρων λειτουργιών σε αυτά τα δεδομένα:

Επιλογές μενού:

Ο χρήστης έχει ένα μενού για να επιλέξει τη λειτουργία που επιθυμεί να εκτελέσει:

1. Υπολογισμός του μέσου όρου:

Καλεί τη συνάρτηση calculate_average_1 RPC.

Εμφανίζει τον επιστρεφόμενο μέσο όρο εάν είναι επιτυχής.

2. Εύρεση Max και Min:

Καλεί τη συνάρτηση find_max_min_1 RPC.

Εμφανίζει τις μέγιστες και ελάχιστες τιμές αν είναι επιτυχής.

3. Πολλαπλασιασμός Scalar:

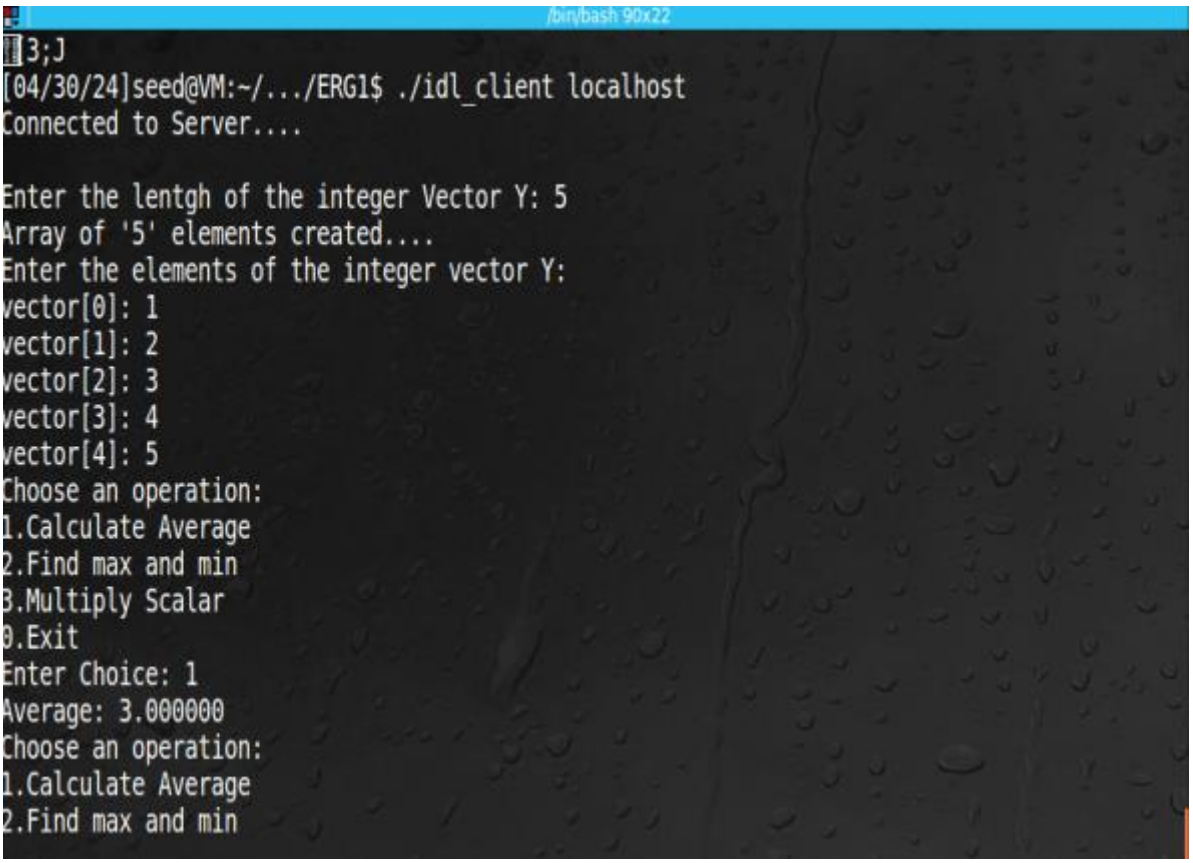
Ζητά από το χρήστη να δώσει μια float τιμή.

Καλεί τη συνάρτηση RPC multiply_scalar_1.

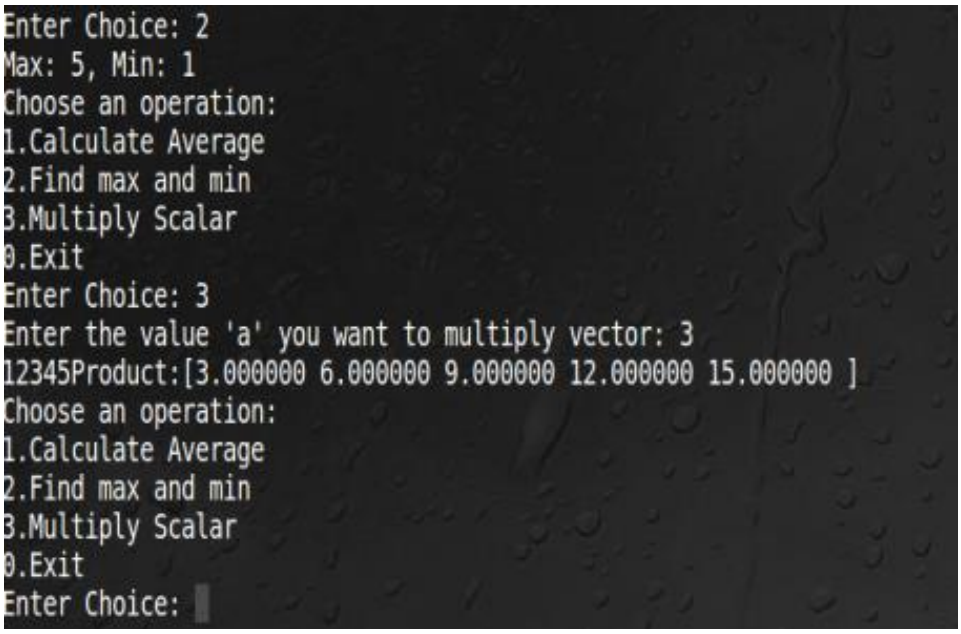
Εμφανίζει τον προκύπτοντα πίνακα αν είναι επιτυχής.

4.ΕΝΔΕΙΚΤΙΚΆ ΠΑΡΑΔΕΙΓΜΑΤΑ.

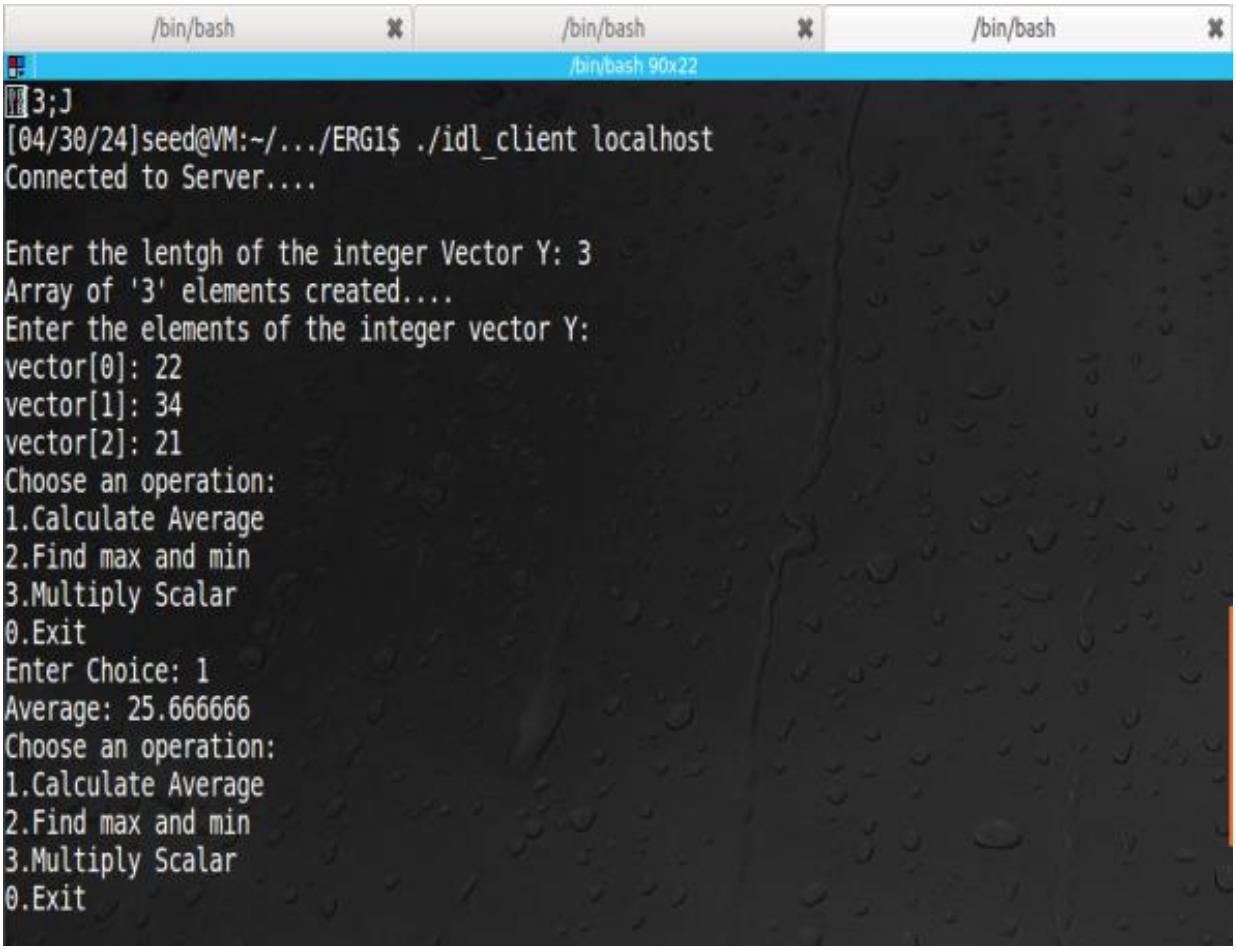
1.Είκονα τρεξιμό server.



2.Είκονα τρέξιμο client δημιουργία πίνακα και ευρεσή μέσο όρο.



3.Είκονα Ευρεσή max , min και Πολλαπλασιάζμος πίνακα με α float τιμε και επίστροφη καινουργιου πίνακα



4. Είκονα εκυπηρετήση και άλλου client παραλλήλα με τον προηγούμενο.