



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

## Άσκηση #1

Ζωβοΐλης Δημήτριος-Μάριος

ΑΜ:19390064

Τμήμα: ΣΤ5-Α (Πέμπτη 12-2)

08/05/22

# ΠΕΡΙΕΧΟΜΕΝΑ

## Πίνακας περιεχομένων

ΠΕΡΙΕΧΟΜΕΝΑ.....	2
1. Εισαγωγή-περιγραφή της εργασίας.....	3
2. Τεκμηρίωση – Επίλυση της εργασίας.....	3

## 1. Εισαγωγή-περιγραφή της εργασίας

Σας ζητείται να φτιάξετε σε C έναν concurrent1 server (διεργασία εξυπηρετητή) ο οποίος ως έργο εξυπηρέτησης θα επιτελεί τους ακόλουθους υπολογισμούς (λαμβάνοντας ως εισόδους έναν πραγματικό αριθμό  $a$  και ένα διάνυσμα ακεραίων της μορφής  $Y (y_1, y_2, \dots, y_n)$  μήκους  $n$  όπου το  $n$  θα το ορίζει ο χρήστης, και τις οποίες θα μπορούν να στέλνουν επαναληπτικά ένας ή περισσότεροι clients2 / διεργασίες πελατών):

1. Τη μέση τιμή του διανύσματος  $Y$  (επιστροφή: ένας πραγματικός αριθμός)
2. Τη μέγιστη και την ελάχιστη τιμή του  $Y$  (επιστροφή: ένας πίνακας μήκους 2 ακεραίων)
3. Το γινόμενο  $a*Y$  (επιστροφή: ένα διάνυσμα πραγματικών αριθμών μήκους  $n$ )

Η επικοινωνία θα πρέπει να γίνεται μέσω TCP AF\_INET (Internet Domain) sockets. Η κάθε διεργασία socket-Client θα διαβάζει από το πληκτρολόγιο (επαναληπτικά, μέχρι να δηλώσει ο χρήστης ότι δεν επιθυμεί να συνεχίσει) (α) την επιλογή του υπολογισμού που επιθυμεί ο χρήστης να γίνει (1,2,3) και (β) τα αντίστοιχα-απαραίτητα κατά περίπτωση δεδομένα ( $n, Y, a$ ), θα τα διοχετεύει στη διεργασία socket-Server και θα περιμένει να λάβει από αυτήν το αποτέλεσμα για να το τυπώσει στην οθόνη.

Η διεργασία socket-Server θα δέχεται τα δεδομένα προς επεξεργασία από τις διεργασίες socket-Clients, και θα παράγει το αντίστοιχο αποτέλεσμα ΟΧΙ μέσω δικιάς του (τοπικής) συνάρτησης-υπολογισμού ΑΛΛΑ μέσω κατάλληλου Remote Procedure Call που θα υλοποιήσετε με τη βοήθεια του ONC RPC implementation. Θα πρέπει δηλαδή η διεργασία socket-Server (λειτουργώντας παράλληλα και ως RPC-Client) να καλεί (ανάλογα με την τιμή υπολογισμού που έστειλε ο χρήστης - 1,2,3) την αντίστοιχη ρουτίνα από έναν RPC-Server και να περιμένει το αντίστοιχο αποτέλεσμα από αυτόν (προκειμένου να το διοχετεύσει στη συνέχεια στον αντίστοιχο socket-Client).

## 2. Τεκμηρίωση – Επίλυση της εργασίας

Στο αρχείο [server\\_funcs.x](#) (το interface file) ορίζονται μέσα σε αυτό τρεις ξεχωριστές συναρτήσεις διαδικασίες (μία για κάθε έναν από τους τρεις υπολογισμούς που ζητούνται παραπάνω).

Τα αρχεία [server\\_funcs\\_server.c](#) και [server\\_funcs\\_client.c](#) αποτελούν την υλοποίηση του RPC server και RPC client αντίστοιχα. Να σημειωθεί πως το αρχείο [server\\_funcs\\_client.c](#) αποτελεί και την πλευρά του server στο κομμάτι server-client των sockets. Τέλος, το αρχείο [client.c](#) αποτελεί την πλευρά του client στο κομμάτι server-client των sockets.

Για να τρέξει το πρόγραμμα μας αρκεί να κάνουμε compile χρησιμοποιώντας την εντολή make στο αρχείο Makefile.server\_funcs και να κάνουμε compile το αρχείο [client.c](#) με τον τρόπο που φαίνεται και στις παρακάτω εικόνες.

```
Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x
Ice19390064@ubuntu:~/RPC$ make -f Makefile.server_funcs
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_clnt.o server_funcs_clnt.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_client.o server_funcs_client.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_xdr.o server_funcs_xdr.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -o server_funcs_client server_funcs_clnt.o server_funcs_client.o server_funcs_xdr.o -lnsl
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_svc.o server_funcs_svc.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_server.o server_funcs_server.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -o server_funcs_server server_funcs_svc.o server_funcs_server.o server_funcs_xdr.o -lnsl
```

```
Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x
Ice19390064@ubuntu:~/RPC$ gcc -o client client.c -pthread
```

Έπειτα, για να τρέξουμε το πρόγραμμα πρέπει να:

1. Να σηκώσουμε τον RPC server,
2. Να σηκώσουμε τον RPC client – Server,
3. και τέλος να τρέξουμε έναν ή περισσότερους client.

1.

```
Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x
Ice19390064@ubuntu:~/RPC$ make -f Makefile.server_funcs
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_clnt.o server_funcs_clnt.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_client.o server_funcs_client.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_xdr.o server_funcs_xdr.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -o server_funcs_client server_funcs_clnt.o server_funcs_client.o server_funcs_xdr.o -lnsl
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_svc.o server_funcs_svc.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -c -o server_funcs_server.o server_funcs_server.c
gcc -g -DRPC_SVC_FG -lpthread -pthread -o server_funcs_server server_funcs_svc.o server_funcs_server.o server_funcs_xdr.o -lnsl
Ice19390064@ubuntu:~/RPC$ ./server_funcs_server
```

2.

```
Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x
Ice19390064@ubuntu:~/RPC$ ./server_funcs_client localhost 51717
Waiting for a connection...
Waiting for a connection...
```

3.

```
Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x Ice19390064@ubuntu: ~/RPC x
Ice19390064@ubuntu:~/RPC$ gcc -o client client.c -pthread
Ice19390064@ubuntu:~/RPC$ ./client localhost 51717
Trying to connect...
Connected.
Dia8esimes epiloges:
  1: Ypologismos mesou orou dianysmatos.
  2: Ypologismos megistis-elaxisths timhs dianysmatos.
  3: Ypologismos ginomenoy dianysmatos me ari8mo.
  4: Ejodos.
Dwse thn epilogh sou:
>
```

Τέλος, ακολουθούν εικόνες από τρεξίματα του προγράμματός μας.

```
Ice19390064@ubuntu: ~/RPC
Ice19390064@ubuntu:~/RPC$ gcc -o client client.c -pthread
Ice19390064@ubuntu:~/RPC$ ./client localhost 51717
Trying to connect...
Connected.
Dia8esimes epiloges:
    1: Ypologismos mesou orou dianysmatos.
    2: Ypologismos megistis-elaxisths timhs dianysmatos.
    3: Ypologismos ginomenoy dianysmatos me ari8mo.
    4: Ejodos.
Dwse thn epilogh sou:
> 3
Dwse to mhkos toy dianysmatos:
> 4
Dwse to dianysma, xwrismeno me kena:
> 1 2 4 10
Dwse ton pragmatiko ari8mo a:
> 0.5
Neos pinakas:
0.500000 1.000000 2.000000 5.000000
Dia8esimes epiloges:
    1: Ypologismos mesou orou dianysmatos.
    2: Ypologismos megistis-elaxisths timhs dianysmatos.
    3: Ypologismos ginomenoy dianysmatos me ari8mo.
    4: Ejodos.
Dwse thn epilogh sou:
> 3
Dwse to mhkos toy dianysmatos:
> 2
Dwse to dianysma, xwrismeno me kena:
> 4 8
Dwse ton pragmatiko ari8mo a:
> 2.5
Neos pinakas:
10.000000 20.000000
Dia8esimes epiloges:
    1: Ypologismos mesou orou dianysmatos.
    2: Ypologismos megistis-elaxisths timhs dianysmatos.
    3: Ypologismos ginomenoy dianysmatos me ari8mo.
    4: Ejodos.
Dwse thn epilogh sou:
```

```
lce19390064@ubuntu:~/RPC$ ./client localhost 51717
Trying to connect...
Connected.
Dia8esimes epiloges:
    1: Ypologismos mesou orou dianysmatos.
    2: Ypologismos megistis-elaxisths timhs dianysmatos.
    3: Ypologismos ginomenoy dianysmatos me ari8mo.
    4: Ejodos.
Dwse thn epilogh sou:
> 1
Dwse to mhkos toy dianysmatos:
> 6
Dwse to dianysma, xwrismeno me kena:
> 1 2 8 9 4 6
Mesh timh:
5.000000
Dia8esimes epiloges:
    1: Ypologismos mesou orou dianysmatos.
    2: Ypologismos megistis-elaxisths timhs dianysmatos.
    3: Ypologismos ginomenoy dianysmatos me ari8mo.
    4: Ejodos.
Dwse thn epilogh sou:
> 2
Dwse to mhkos toy dianysmatos:
> 5
Dwse to dianysma, xwrismeno me kena:
> 23 54 87 103 76
Max Min: 103 23
Dia8esimes epiloges:
    1: Ypologismos mesou orou dianysmatos.
    2: Ypologismos megistis-elaxisths timhs dianysmatos.
    3: Ypologismos ginomenoy dianysmatos me ari8mo.
    4: Ejodos.
Dwse thn epilogh sou:
> 
```