



ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

RPC and sockets – Εργασία 1

Αθήνα 2021-2022

Περιεχόμενα

ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ	1
RPC and sockets – Εργασία 1.....	1
Περιεχόμενα	2
Στόχος εργασίας	3
Compilation	4
Test runs	4

Στόχος εργασίας

Στην παρούσα εργασία μας ζητείτε να φτιάξουμε έναν C concurrent server ο οποίος θα δέχεται πολλούς clients και θα επιτελεί 3 διαφορετικούς υπολογισμούς:

- Μέση τιμή ενός διανύσματος Y
- Μέγιστη και ελάχιστη τιμή του Y
- Γινόμενο ενός αριθμού α με το Y

Τα παραπάνω υλοποιήθηκαν με την χρήση sockets, RPC και forks. Αρχικά έφτιαξα το .x αρχείο το οποίο περιέχει 4 διαφορετικά structs για κάθε έναν από τους υπολογισμούς και ένα για το input του χρήστη. Τα structs περιέχουν μέσα τον τύπο δεδομένων που θα πρέπει να επιστρέψουμε και τα υλοποίησα με βάση το [RPCL documentation](#).

Έπειτα με τις παρακάτω εντολές έγιναν generate τα RPC αρχεία μου:

```
rpcgen -C RPC.x
rpcgen -a -C RPC.x
make -f Makefile.RPC
mv Makefile.RPC Makefile
```

Σημαντική σημείωση 1: επειδή μου δημιουργήθηκαν πολλά errors λόγω του OS μου (Fedora OS) έπρεπε να κατεβάσω το tirc για να μου δουλέψει σωστά το RPC επομένως το Makefile μου έχει τα εξής flags:

```
CC = gcc
CFLAGS += -g -I/usr/include/tirpc
LDLIBS += -ltirpc
RPCGENFLAGS = -C
```

Όπου /usr/include/tirpc είναι το path το οποίο βρίσκεται το tirpc.

Έπειτα έφτιαξα το RPC_soc.c αρχείο το οποίο περιέχει το μενού επιλογών που θα εμφανίζεται στον χρήστη και στέλνει τα input του στο TCP server/RPC client μέσω socket.

Έπειτα συμπλήρωσα το `RPC_server.c` το οποίο περιέχει τις 3 πράξεις που θα υλοποιούνται και θα καλούνται από τον `RPC_client.c`. Το αρχείο υλοποιήθηκε με βάση τα παραδείγματα του `eclass`.

Τέλος, συμπλήρωσα το `RPC_client.c` το οποίο περιέχει μια `switch case` για κάθε επιλογή του χρήστη και λαμβάνει τα `input` του μέσω `socket`. Επίσης για την υλοποίηση σύνδεσης πολλαπλών χρηστών χρησιμοποιήθηκαν `forks`.

Σημαντική σημείωση 2: λόγω του OS μου ξανά έπρεπε να κάνω επανεκκίνηση το `service rpcbind` το οποίο έγινε με την εντολή `service rpcbind restart`.

Compilation

```
[kat in ~/Documents/RPC
$ make
gcc -g -I/usr/include/tirpc -c -o RPC_clnt.o RPC_clnt.c
gcc -g -I/usr/include/tirpc -c -o RPC_client.o RPC_client.c
RPC_client.c: In function 'main':
RPC_client.c:177:34: warning: implicit declaration of function 'waitpid' [-Wimplicit-function-declaration]
  177 |         procId = waitpid((pid_t) - 1, NULL, WNOHANG);
      |                        ~~~~~
gcc -g -I/usr/include/tirpc -c -o RPC_xdr.o RPC_xdr.c
gcc -g -I/usr/include/tirpc -o RPC_client RPC_clnt.o RPC_client.o RPC_xdr.o -ltirpc
gcc -g -I/usr/include/tirpc -o RPC_server RPC_svc.o RPC_server.o RPC_xdr.o -ltirpc
[kat in ~/Documents/RPC
$ gcc -o RPC_soc RPC_soc.c
RPC_soc.c: In function 'main':
RPC_soc.c:41:5: warning: implicit declaration of function 'inet_aton' [-Wimplicit-function-declaration]
   41 |     inet_aton(argv[1], &serv_addr.sin_addr);
      |     ~~~~~
RPC_soc.c:129:5: warning: implicit declaration of function 'close'; did you mean 'pclose' [-Wimplicit-function-declaration]
  129 |     close(sockfd);
      |     ~~~~~
      |     pclose
```

Test runs

The screenshot displays three terminal windows. The top-left window, titled 'RPC : RPC_server', shows the compilation of RPC-related files using 'gcc' and 'rpcgen'. It includes a warning about an implicit declaration of 'waitpid' and a list of compilation flags. The top-right window, titled 'RPC : bash', shows the execution of './RPC_soc localhost 43438', which opens a socket and connects to the server. It then presents a menu of actions: 1. Find average of array, 2. Find min and max of array, 3. Multiply array with number, and 4. Exit. The user has chosen option 1. The bottom-left window, titled 'RPC : RPC_client', shows the execution of './RPC_client localhost 43438', which connects to the server and sends a request. The bottom-right window, titled 'RPC : bash', shows the server's response, which includes a menu of actions: 1. Find average of array, 2. Find min and max of array, 3. Multiply array with number, and 4. Exit. The user has chosen option 4, and the server responds with 'Goodbye!'.

```
RPC_client.c: In function 'main':
RPC_client.c:177:34: warning: implicit declaration of function 'waitpid' [-Wimplicit-function-declaration]
   177 |         procId = waitpid((pid_t) - 1, NULL, WNOHANG);
       |                        ^~~~~~
rpcgen -C RPC.x
gcc -g -I/usr/include/tirpc -c -o RPC_xdr.o RPC_xdr.c
gcc -g -I/usr/include/tirpc -o RPC_client RPC_clnt.o RPC_client.o RPC_xdr.o -ltirpc
gcc -g -I/usr/include/tirpc -c -o RPC_svc.o RPC_svc.c
gcc -g -I/usr/include/tirpc -c -o RPC_server.o RPC_server.c
gcc -g -I/usr/include/tirpc -o RPC_server RPC_svc.o RPC_server.o RPC_xdr.o -ltirpc
[kat in ~/Documents/RPC [ 13:36 ]
$ service rpcbind stop
Redirecting to /bin/systemctl stop rpcbind.service
[kat in ~/Documents/RPC took 1s888ms [ 13:36 ]
$ service rpcbind start
Redirecting to /bin/systemctl start rpcbind.service
[kat in ~/Documents/RPC took 1s844ms [ 13:36 ]
$ ./RPC_server
Choose which action you want to perform:
1. Find average of array
2. Find min and max of array
3. Multiply array with number
4. Exit
Choose: 1
Enter the number a to multiply the array with:
5
Enter the size of array:

RPC : RPC_client
[kat in ~/Documents/RPC [ 13:36 ]
$ ./RPC_client
usage: ./RPC_client server_host
[kat in ~/Documents/RPC [ 13:37 ]
$ ./RPC_client localhost 43438
Connected.
Waitpid error.
: No child processes
Waitpid error.
: No child processes
Goodbye!
[kat in ~/Documents/RPC took 49s908ms [ 13:38 ]
```

Στο πάνω δεξιά παράθυρο έκανα start το rpcbind και τον RPC server. Στο κάτω δεξιά έκανα start το RPC client/TCP server. Στα άλλα δύο είναι 2 clients οι οποίοι είναι συνδεδεμένοι την ίδια στιγμή στον TCP server. Πιο συγκεκριμένα ο πάνω αριστερά ζητά να γίνει εύρεση μέσης τιμής. Έχουμε το παρακάτω:

```
kat in ~/Documents/RPC
$ ./RPC_soc localhost 43438
Opening socket...
Connected..
Choose which action you want to perform:

1. Find average of array
2. Find min and max of array
3. Multiply array with number
4. Exit

Choose: 1

Enter the number a to multiply the array with:
5

Enter the size of array:
5

Fill the array
Y[0]:1
Y[1]:2
Y[2]:3
Y[3]:4
Y[4]:5
Average is: 3.000
```

Ο κάτω δεξιά από την άλλη ζητά την εύρεση μέγιστου και ελάχιστου και μετά να γίνει ο πολλαπλασιασμός $a*Y$.

```

[kat in ~/Documents/RPC
$ ./RPC_soc localhost 43438
Opening socket...
Connected..
Choose which action you want to perform:
1. Find average of array
2. Find min and max of array
3. Multiply array with number
4. Exit
Choose: 2
Enter the number a to multiply the array with:
5
Enter the size of array:
5
Fill the array
Y[0]:1
Y[1]:2
Y[2]:3
Y[3]:4
Y[4]:5
Min is: 1
Max is: 5
Max is: 5
Choose which action you want to perform:
1. Find average of array
2. Find min and max of array
3. Multiply array with number
4. Exit
Choose: 3
Enter the number a to multiply the array with:
5
Enter the size of array:
5
Fill the array
Y[0]:1
Y[1]:2
Y[2]:3
Y[3]:4
Y[4]:5
The multiplied array is:
X[1]:5.000
X[2]:10.000
X[3]:15.000
X[4]:20.000
X[5]:25.000

```

Και στο τέλος επιλέγει να φύγει.

```

Choose which action you want to perform:
1. Find average of array
2. Find min and max of array
3. Multiply array with number
4. Exit
Choose: 4
Goodbye!

```