UNIVERSIDAD CARLOS III DE MADRID
***BACHELOR IN INFORMATICS ENGINEERING*. DISTRIBUTED SYSTEMS**
Midterm. March 20th, 2013.

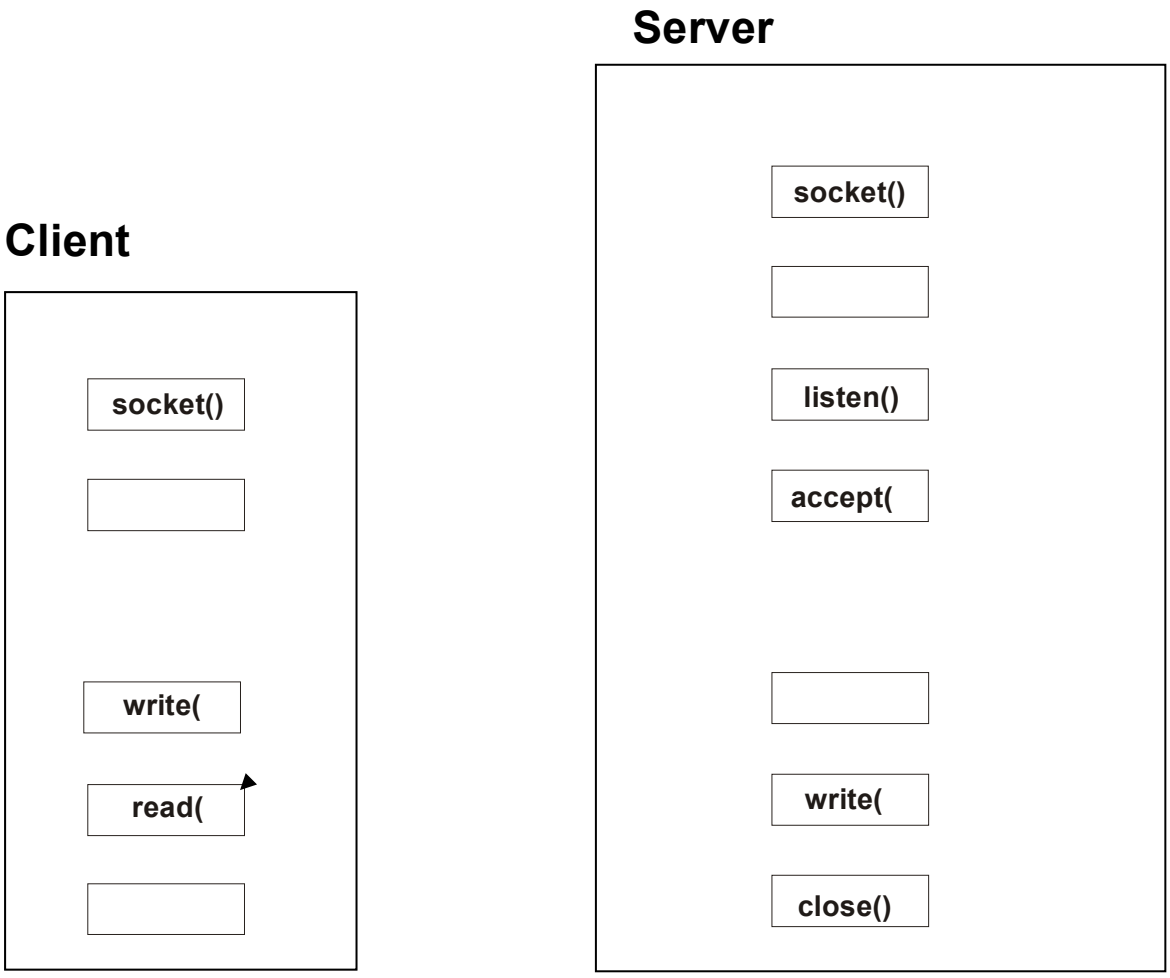
To conduct this examination will be available **one hour and 20 minutes**. Using books, notes or calculators of any kind is not allowed. Respond to the exercises in the provided space.

Name: _____ **Group:** _____

Exercise 1 (2 points). Given the following code, indicate if it is executed properly. What would be the most likely output of this code?

```
#include <stdio.h>
#include <pthread.h>
#define NUM_THREADS 10

int function(int *value) {
        int k;

        sleep(100);                    /* sleep 100 seconds */

        k = *value;
        printf("Value = %d: ", k);
        pthread_exit(NULL);
}

int main () {
        pthread_t arrayThreads[NUM_THREADS]; /* Threads array */
        int i;

        /* THREADS CREATION */
        for(i=0; i<NUM_THREADS; i++){
                if (pthread_create(&arrayThreads[i],  NULL, (void *)function,    &i)==-1)
                        printf("Error creating pthread\n");
        }

        /* WAIT FOR THREAD TERMINATION */
        for(i=0; i<NUM_THREADS; i++)
                pthread_join (arrayThreads[i], NULL);
        exit(0);
}
```

Exercise 2. (2 points) Given the following scheme of communication between a client and a server using TCP sockets.



State:

a) Indicate on the above picture the missing services or calls.

b) The above structure represents a sequential server. Convert the previous scheme in which the server works concurrently.

c) If the client runs on the IP address 163.117.148.200 and the server on IP address 163.117.148.10 and port 80, when a port is assigned to the client, what port is assigned?

Exercise 3. (1.5 points) Regarding to remote procedure calls, answer the following question:

a) Given a client, a server, and a binder, indicate the messages exchange required to register an RPC service and the message exchange required to locate a RPC service.

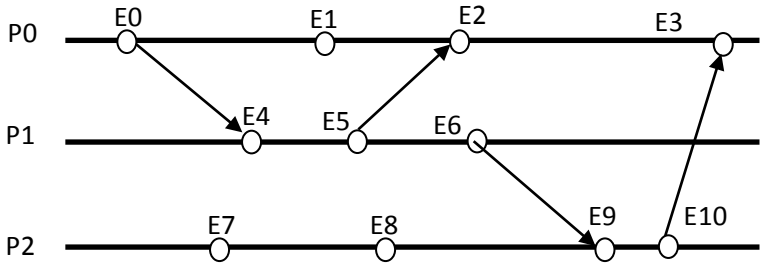
b) The following output corresponds to the execution of the coomand

```
rpcinfo -p guernika.lab.inf.uc3m.es:
```

```
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100003    4   tcp   2049  nfs
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    4   udp   2049  nfs
    200030    2   tcp   9000  calculator
    200030    2   udp   9000  calculator
```

What is the function of the portmapper service? If a client tries to access `guernika.lab.inf.uc3m.es`, port 9000, service "calculator" and version "1". Did properly execute RPC call to the service? Justify your answer.

Exercise 4. (1.5 points) Consider the processes P1, P2 and P3 running in a distributed system. These processes generate events marked in the figure below.



State:

a) Name three pairs of events where there is potential Lamport causal relationship.

b) Name three pairs of events that are concurrent Why are concurrent?

c) Using Lamport logical clocks indicate the time stamps for the events of the previous processes.

d) Indicate what the problem posed by Lamport logical clocks. Give an example using the above diagram.

Exercise 5 (3 points): It aims to implement a distributed barrier with the following features. A set of N client processes that do not share memory with each other, want to use a synchronization barrier offered by a server process. Clients iterate F stages each with M iterations (as in the proposed exercise 1). In each iteration, each process sleeps a random number of seconds. At the end of each phase, each client is blocked waiting for completion at the end of phase all customers. For this, each client sends a request indicating the server you want to run the synchronization barrier. Meanwhile, the server implements the barrier, which simply unlocks all blocked customers upon receipt of request from customers latter process.

State:

a) Design the previous client-server application using POSIX message queues, indicating and specifying necessary design issues.

b) According to the design developed, implemented in C, the code of the client and the server. No need to make a fully detailed implementation of the code.