UNIVERSIDAD CARLOS III DE MADRID
AREA DE ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES
BACHERLOR IN INFORMATICS ENGIEERING. DISTRIBUTED SYSTEMS
FINAL EXAM. May 20 2013.

Student: _____ **Group:** _____

Question 1 (2 points): Answer the following questions:

a) Given a distributed system consisting of *N* nodes. In this system a service request takes *t* units of time. Consider *R* requests the same service running on *R* * *t* units of time. What is the scalability of a distributed system. Is this system scalable? Justify your answer.
b) What is the RMIRegistry component in Java RMI?
c) List the advantages and disadvantages of using cache blocks in the clients of a distributed file service with respect to solutions who does not use cache blocks..
d) What is volunteer computing? Cite some projects that implement this paradigm.

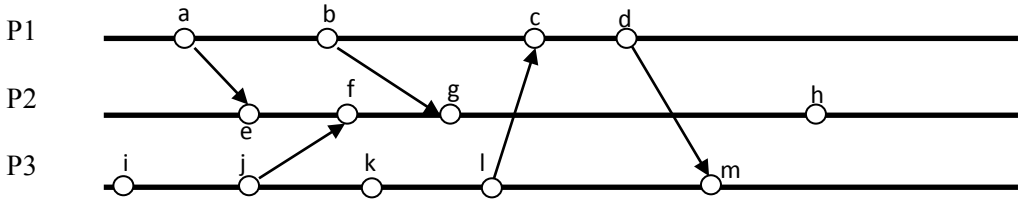
Question 2 (1 point): Given the following SOAP request message:

```
<SOAP-ENV:Envelope  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
     <m:GetPrice xmlns:m="http://example.com/stockquote.xsd">
          <item>book</item>
     </m: GetPrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

State:

a) What IDL is used for SOAP-based web services?
b) What function is encapsulated in this web service?
c) Which transport protocol can be used to encapsulate this message?
d) Identify the main fields of the SOAP message.

Question 3 (1 punto): Consider the processes P1, P2 and P3 that are running in a distributed system. These processes generate the events marked in the figure.



State:

a) Indicate three pairs of sorted events that use the Lamport's potential causality.
b) Indicate three pairs of events that are concurrent. Why are concurrent?
c) Using Lamport logical clocks, indicate the timestamps to the events of the previous processes.
d) Using vector clocks, set the time stamps for the events of the previous processes.

Question 4 (2 points): A distributed system uses the following interface for message passing:

- `send(i,msg)` – send the message `msg` to the process `i`
- `receive(i,&msg)` – recieve the message `msg` from the process `i`

The system consists of *N* distributed processes where i process IDs ranging from *0* to *N-1*, and wherein each process executes the following code.

```
1:    my_function(int i){
2:        if (i==0)
3:              send(i+1,msg);

4:         while(1){
5:              if (i==0)  j=N;
6:              else  j=i;
7:              receive(j-1,&msg)
8:              // Do something
9:              if (i==N-1) j=-1;
10:             else j=i;
11:             send(j+1,msg);
12:        }
13:   }
```

Answer the following questions:

a) How many times at most can be concurrently executed the receive operation on line 7?
b) How many times at most the code can be concurrently executed on line 8?
c) How are logically ordered processes running on the distributed system?
d) How useful is the algorithm? Justify.

Question 5 (4 points): We ask to implement a client-server application for searching files which contain medical images. The service includes the following operations:

- Search an image in the image server. The client asks the server if an image exists on the server. The client sends the name of the file and the server will send a code indicating whether or not that image exists.

- *Get* an image from the image server. The client can retrieve an image file from the server. The client sends to the server the name of the image and the server sends the image to the client.

- *Store* an image on the image server. An image is given by name, date of creation, and the image file. The contents of this file will be sent to the server, which save it into a file.

Note: the image file can be of any size.

State:

a) Design a client-server application, indicating and specifying all aspects necessary for your design. As part of the design, describe in detail the service protocol.

b) According to the previous design, implement in the C programming language, the server code.