Solver: Dang Xuan Vu

1) a) Model Description:

Client-Server:

Servers manage the resources and provide the service for clients, whereas **clients** use the service provided by the servers to access their resources. Their interactions are on **request-response** basis: any client makes a **request** to the server, and the server sends a **response** in reply.

Peer-to-peer:

Resources are managed and shared across **peers**, which act as **both server** (serving the resources) and **client** (requesting the resources).

Difference:

- In client-server model, the roles are separated by clients who access the resources, and servers who provide the resources, whereas in peer-to-peer model the resources are hosted on each of the participating peer.
- In client-server model, requests can only be sent from clients to servers, and responses can only be sent from servers to clients, whereas in peer-to-peer model the request-response can originated from any peers.

Examples:

- Client-Server: Google (search engine)
- Peer-to-peer: Napster (file sharing)

b) i) Server's interface

```
import java.rmi.*;
public interface Server extends Remote {
     void updateStock(String name, int number) throws
     RemoteException;
     void register(Client client) throws
     RemoteException;
}
```

ii) Client's interface

```
import java.rmi.*;
public interface Client extends Remote {
     void alert(String name) throws RemoteException;
}
```

iii) Pass by value: String name, int number
Pass by reference: Client client

2) a) States of callback promise:

- Valid: the cached copy of the file is fresh
- Cancelled: the cached copy of the file is stale

Situations:

- Client opens a file that is cached and its callback promise is cancelled: **cancelled → valid**
- Upon receiving an update of a file, server broadcasts callback to the clients holding that file: **valid → cancelled**

b) In case of an unreliable communication channel, a possible solution to avoid callback lost is to introduce an acknowledgement for the callback. Upon receipt of the callback, the client sends an acknowledgement to the server. If the acknowledgement is not received by the server within a time frame, the server considers the callback lost and resend the callback.

Advantages: Simple yet reliable solution
Disadvantages: Incur more network traffic and more load on servers

c) i) Node **N38** is responsible for file **K36**
Node **N3** is responsible for file **K120**

ii) N53 → N118 → N32 → N38

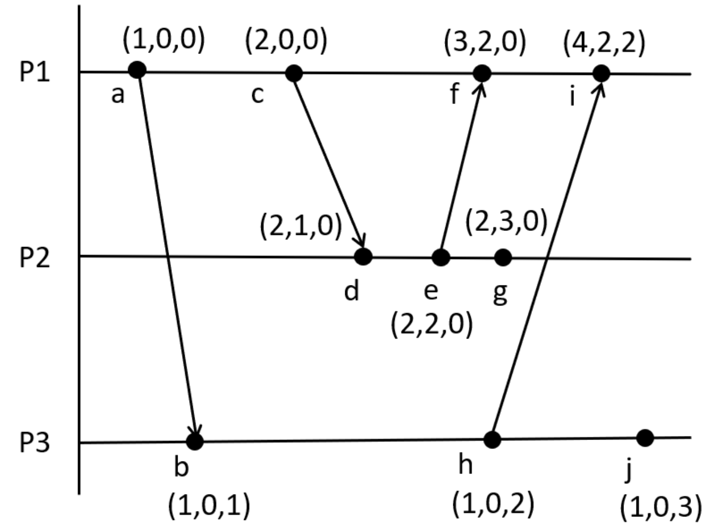
How the tracing goes:
(**53** + 64) mod 128 = 117 < **118**
(**118** + 32) mod 128 = 22 < **32**
(**32** + 4) mod 128 = 36 (K36 is the file) < **38**

iii) Node **N38** need to update it **7th entry** from N118 to **N105** (**38** + 64 = 102 < **105**)

3) a) a, c, d, e

b) g, b, h, j

c)



a – (1,0,0) c – (2,0,0) f – (3,2,0) i – (4,2,2)

d – (2,1,0)	e – (2,2,0)	g – (2,3,0)
b – (1,0,1)	h – (1,0,2)	j – (1,0,3)

d) p_1 should set its time to:

$$t_{p1} = \frac{1}{2}\left(t_f + t_e + t_d - t_c\right) = 80$$

The accuracy of the setting:

$$\Delta t = \pm\frac{1}{2}\left(t_f + t_d - t_e - t_c\right) = \pm 20$$

e) FIFO message delivery order ensures that on any communication channel, if message m is sent before message m', then message m must be received before message m'.
Possible states:

$$(p_1, p_2, p_3) = (S_a, S_2, S_b), (c_{3\to2}, c_{2\to1}, c_{3\to1}) = (\emptyset, \emptyset, \emptyset)$$
$$(p_1, p_2, p_3) = (S_a, S_2, S_h), (c_{3\to2}, c_{2\to1}, c_{3\to1}) = (\emptyset, \emptyset, m)$$
$$(p_1, p_2, p_3) = (S_a, S_2, S_i), (c_{3\to2}, c_{2\to1}, c_{3\to1}) = (\emptyset, \emptyset, m)$$

Where *m* denotes the message $h \to i$

4) a) i) A: state := WANTED
B: wait for responses from all the other processes
C: state == RELEASED
D: reply to all processes stored in the queue

ii) There is a possibility that all processes have non-empty queues at the same time, that is for N processes in the system, 1 is holding the CS, N-3 have established their interest in the CS, and last 2 broadcasts their interest at the same time. The last two processes will receive each other requests, which will then be queued since their states are both WANTED.

b) i) $ab \in \{41,43,31\}$

ii) $ab \in \{21,22,23,24,32,33,34,42\}$

iii) $RW \in \{15,25,35,45,55,24,34,44,54,33,43,53\}$

--End of Answers--