**Assignment 3:**

EXERCISE 1 - RPC execution

(a)

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | c2 → s1(k1,5) | s1 (k1,5) | s1 → c2(k1,5) |
| 2. | c2 → s1(k1,3) | s1 (k1,8) | s1 → c2(k1,8) |
| 3. | c4 → s2(k2,8) | s2 (k2,8) | s2 → c4(k2,8) |
| 4. | c1 → s2(k1,1) | s2 (k1,1) | s2 → c1(k1,1) |
| 5. | c3 → s1(k2,4) | s1 (k2,4) | s1 → c3(k2,4) |
| 6. | c3 → s1(k2,5) | s1 (k2,9) | s1 → c3(k2,9) |
| 7. | c1 → s2(k1,1) | s2 (k1,2) | s2 → c1(k1,2) |
| 8. | c3 → s1(k10,3) | s1 (k10,3) | s1 → c3(k10,3) |
| 9. | c4 → s1(k2,1) | s1 (k2,10) | s1 → c4(k2,10) |
| 10. | c1 → s1(k2,9) | s1 (k2,19) | s1 → c1(k2,19) |

EXERCISE 2 - RPC failure semantics

(a)

|  |  |  |  |
| --- | --- | --- | --- |
| 1. | c3 → s1(k2,4) | s1 (k2,4) | s1 → c3(k2,4) |
| 2. | c2 → s2(k3,2) | s2 (k3,2) | s2 → c2(k3,2) |
| 3. | c1 → s1(k1,3) | s1 (k1,3) | s1 → c1(k1,3) |
| 4. | c2 !→ s1(k1,3) | failed |  |
| 5. | c4 !→ s2(k1,2) | failed |  |
| 6. | c3 → s2(k3,8)  c3 → s2(k3,8) | s2 (k3,10)  s2 (k3,18) | s2 → c3(k3,10)  s2 → c3(k3,18) |
| 7. | c4 → s2(k1,3) | s2 (k1,3), !s | s\*, s2 → c4(k1,3) |
| 8. | c4 !→ s1(k2,8)  c4 → s1(k2,8) | s1 (k2,12) | s1 → c4(k2,12) |
| 9. | c3 → s1(k3,14)  c3 → s1(k3,14) | s1 (k3,14)  s1 (k3,28) | s1 → c3(k3,14)  s1 → c3(k3,28) |
| 10. | c4 → s1(k2,2)  c4 → s1(k2,2) | s1 (k2,14), !s  s1 (k2,16) | s\*  s1 → c4(k2,16) |
| 11. | c2 → s1(k1,3) | !s | s\* |
| 12. | c3 → s2(k3,7) | s2 (k3,25) | s2 → c3(k3,25) |
| 13. | c1 → s2(k1,10)  c1 → s2(k1,10) | s2 (k1,13)  s2 (k1,23), | s2 → c1(k1,13)  s2 → c1(k1,23) |
| 14. | c1 !→ s2(k2,5)  c1 → s2(k2,5) | s2 (k2,5) | s2 → c1(k2,5) |
| 15. | c2 → s2(k1,1) | s2 (k1,24), !s | s\* |
| 16. | c1 → s1(k1,5) | s1 (k1,8) | s1 → c3(k1,8) |

EXERCISE 3 - RPC service definition

(a)

**Define data model:**

message Costumer{

required int32 costumer\_id = 1;

required string name = 2;

required Address address = 3;

}

message Address{

required string city = 1;

required int32 zipcode = 2;

required string street = 3;

}

message Item{

required int32 item\_id = 1;

required string name = 2;

required Category category = 3;

required float price = 4;

}

message Category{

required int32 category\_id = 1;

required string name = 2;

}

message Order{

required int32 order\_id = 1;

required date date = 2;

required float price = 3;

}

(b)

**Define service:**

service AdministrationService{

rpc CreateCustomer (int32, string, Address) returns (Customer);

rpc CreateCategory (int32, string) returns (Category);

rpc DeleteCustomer (int32) return ();

rpc DeleteCategory (int32) return ();

rpc SearchCustomer(string) return (Customer[]);

rpc SearchCategory(string) return (Category[]);

}