Examination - Distributed Systems - HT25
- with answers

Welcome to the exam in DIT357. You can achieve up to **100 points**, whereof 35 points are for Part A: Computer Communication and 65 points are for Part B: Distributed Systems.

Allowed and not-allowed material

- A cheat-sheet of DIN-A4 format *handwritten* on both sides. You can take the cheat-sheet home again after the exam.
- No books, no other notes, no calculators, no electronic devices, and especially no form of artificial intelligence (AI/LLMs) are allowed during the exam.
- **Your phones and other electronic devices must be hard switched off (not just in silent or airplane mode, etc.). If we find an activated device with you, it will result in an immediate termination of the exam for you and 0 points as a result.**

Instructions

- The exam paper consists of 8 problems, whereof Problems 1 and 4 consist of multiple-choice questions.
- Please DO NOT use a red pen.
- Please answer in English, Swedish, or German.
- Write clearly your name on *each page* of the answer sheets.
- Indicate clearly on each page of the answer sheet which Problem you are solving.
- Write in a **clear manner** and **motivate** (explain, justify) your answers. This includes showing steps in mathematical calculations.
- A **rule-of-thumb for the extent of detail to provide** is to include enough information/explanation so that a person, whose knowledge on distributed systems is at the level of our introductory lecture, can understand.
- If you make any assumptions in answering any item, do not forget to clearly state what you assume.
- If your answer is not clear to us or we cannot read your handwriting, it will be considered wrong.
- Any attempt to cheat during the exam will result in an immediate termination of the exam and 0 points as a result.

Information about the multiple-choice problems

- Each multiple-choice question has four answers. Only one of the suggested answers is correct.
- If you choose the correct answer, you will receive +1 point.
- You will have an additional fifth option: “I do not know the answer”.
- If you choose the “I do not know the answer” option, you will receive +0.25 points.
- You can only choose one of the answers.

- There is no specific "Answer Sheet" for the multiple choice. Write the answer on your normal answer sheet.
- Write clearly on your answer sheet the Question number and answer number. Example: 1a): Answer 1
- If you choose several answers for a question, or we cannot determine which answer you chose, you will receive 0 points, or 0.25 points in case "I do not know the answer" was one of your chosen answers.

Grading scheme

The following grading scheme will be applied:

- 0-49 points: Not passed (U)
- 50-69 points: Pass (3)
- 70-89 points: Pass (4)
- 90-100 points: Pass (5)
- We will round the final result to the next full point *in the student's favour:* 49.25 points -> 50 points.

Bonus Points

- The Bonus Points from the midterm exam count only towards Part A of this exam.
- **You cannot achieve more than 35 points in Part A of the exam, *including the bonus point from the midterm exam*.** For example, if you have 15 bonus points, and achieve 25 points in Part A, you will receive 35 points in Part A.
- You can pass with bonus points, as long as you do not exceed 35 points in Part A. For example: If you have 15 bonus points, and you get 20 points in Part A and 15 points in Part B of this exam, you will pass with 50 points.

Part A: Computer Communication

You can achieve up to 35 points in Part A.

Problem 1: Multiple Choice (10 points)

For each of the following questions only one of the answers is correct. For each correctly answered question you will receive +1 point. If you are not sure about the correct answer, you can use the option "I do not know the answer", which will give you +0.25 points.

1a) Which of the following best defines a computer network? (1 point)

1) A system where computers operate independently without communication.
2) A collection of devices, routers, and links that enable data exchange between hosts.
3) A group of personal computers directly connected via Ethernet.
4) A software-only system for executing parallel computations.
5) I do not know the answer (+0.25 points).

1b) Why are sequence numbers used in reliable data transfer? (1 point)

1) To indicate which application is using the transport layer.
2) To compute checksums.
3) To detect duplicates and ensure correct packet ordering.
4) To enforce congestion control.
5) I do not know the answer (+0.25 points).

1c) Which transport protocol provides reliable, in-order delivery of data? (1 point)

1) UDP
2) IP
3) TCP
4) ICMP
5) I do not know the answer (+0.25 points).

1d) Which mechanism allows TCP to avoid overwhelming the receiver? (1 point)

1) Flow control
2) Error correction
3) Congestion avoidance
4) Window scaling
5) I do not know the answer (+0.25 points).

1e) Why is stop-and-wait inefficient? (1 point)

1) Because it wastes CPU cycles during transmission.
2) Because receivers cannot process packets fast enough.
3) Because packets are transmitted in bursts only.
4) Because the sender spends most time waiting for acknowledgements.
5) I do not know the answer (+0.25 points).

1f) What is the main advantage of hierarchical IP addressing? (1 point)

1) It allows scalable routing by prefixes matching.
2) It simplifies human memorisation of addresses.
3) It prevents packet fragmentation.
4) It eliminates the need for NAT.
5) I do not know the answer (+0.25 points).

1g) What is the function of NAT in home networks? (1 point)

1) To assign a unique public IP to each device.
2) To enforce IPv6-only communication.
3) To guarantee QoS for all packets.
4) To allow many private hosts to share one public IP.
5) I do not know the answer (+0.25 points).

1h) Consider the network `192.168.10.0/24`. You need to create 6 subnets, each with at least 25 usable host addresses. Which subnet mask should be used for the subnets? (1 point)

1) 255.255.255.128
2) 255.255.255.192
3) 255.255.255.224
4) 255.255.255.240
5) I do not know the answer (+0.25 points).

1i) You are given the subnet `192.168.45.64/27`. You need to broadcast a message to all connected clients. What address do you use? (1 point)

1) 192.168.45.65
2) 192.168.45.94
3) 192.168.45.95
4) 192.168.45.127
5) I do not know the answer (+0.25 points).

1j) Which of the following statements about IPv6 is correct? (1 point)

1) IPv6 uses 32-bit addresses, allowing approximately 4 billion unique addresses.
2) IPv6 eliminates the need for routing tables by using broadcast addresses.
3) IPv6 is only used in mobile-phone networks and not supported on wired infrastructure.
4) IPv6 addresses are 128 bits long and written in hexadecimal colon notation.
5) I do not know the answer (+0.25 points).

Answers:

1a) 2 A collection of devices, routers, and links that enable data exchange between hosts.

1b) 3 To detect duplicates and ensure correct packet ordering.

1c) 3 TCP

1d) 1 Flow control

1e) 4 Because the sender spends most time waiting for acknowledgements.

1f) 1 It allows scalable routing by prefixes matching.

1g) 4 To allow many private hosts to share one public IP.

1h) 3 255.255.255.224

1i) 3 192.168.45.95

1j) 4 IPv6 addresses are 128 bits long and written in hexadecimal colon notation.

Problem 2: A drone live video streaming protocol (15 points)

Design and describe an application-level protocol to be used between a mobile app and a camera-equipped drone for live video streaming. The protocol should allow:

- The user to initiate and terminate a video stream;
- The drone to start sending a continuous stream of video data packets;
- The app to receive and display the video *with minimal latency.*

Your tasks are:

a) Specify the protocol by:

  - Listing the messages exchanged and their purposes. (2 points)
  - Outlining actions by the mobile app or the drone on sending or receiving each message. (2 points)
  - Sketching the normal operation of your protocol (no errors) using a simple arrow diagram. (2 points)
  - Sketching a situation in which packet loss or reordering occurs using a simple arrow diagram and explaining how the app should react in such a situation. (2 points)

b) Which common transport protocol could be used? Explain why it is sufficient, and eventually even beneficial, to use a connectionless transport service. (2 point)

The transport protocol of your choice specifies the 1s-complement for checksums. Assume you have the following three 16-bit words from your drone:

- `1001 0110 0101 1011;`
- `0110 1110 0010 0001;`
- `0000 1111 1100 0100.`

c) Compute the checksum for this data. Show all steps of your addition. (4 points)

d) How can the receiver use this checksum to detect transmission errors? (1 point)

Answer:

Part a) Protocol specification:

Messages from Mobile App to Drone

Msg Name	Purpose
HELLO <userid>	Initiate contact with the drone
START_STREAM	Request to begin live video streaming
STOP_STREAM	Request to terminate the stream
BYE	End the session

Messages from Drone to Mobile App

Msg Name	Purpose
OK	Confirm last operation (e.g., stream started/stopped)
VIDEO_PKT <seq, timestamp, payload>	One packet of video stream data (connectionless, continuous)
BYE	End session and shut down the streaming interface

Correct Operation: Continuous Video Stream

Client (App)		Server (Drone)
HELLO <userid>	→	Acknowledge user
START_STREAM	→	Begin sending video packets
	←	OK
	←	VIDEO_PKT(seq=101)
	←	VIDEO_PKT(seq=102)
	←	VIDEO_PKT(seq=103)
...		...
STOP_STREAM	→	Stop camera
	←	OK
BYE	→	End session
	←	BYE

Error-Tolerant Operation: Packet Loss and Reordering

Client (App)		Server (Drone)	
...		...	
	←	VIDEO_PKT(seq=201)	
	←	VIDEO_PKT(seq=203)	*(seq=202 lost)*
	←	VIDEO_PKT(seq=204)	
	←	VIDEO_PKT(seq=206)	
	←	VIDEO_PKT(seq=205)	*(arrived out of order)*
...		...	

The app should simply drop the missing data packets as they become irrelevant in a live video stream.

Part b) The app and drone do not require a persistent connection. Packet loss, duplication, or reordering can be tolerated by the application logic. Low latency is prioritised over reliability, making retransmissions unnecessary. If a data packet in a live video stream is lost, it makes no sense anyway to retransmit it.

Part c) *Step 1: Add the first two words*:

$$\begin{array}{l} \texttt{1001 0110 0101 1011} \\ +\texttt{0110 1110 0010 0001} \\ \texttt{---- ---- ---- ----} \\ =^{1}\texttt{0000 0100 0111 1100} \end{array}$$

Step 2: Add carry around:

$$\begin{array}{l} \texttt{0000 0100 0111 1100} \\ +\texttt{0000 0000 0000 0001} \\ \texttt{---- ---- ---- ----} \\ =\texttt{0000 0100 0111 1101} \end{array}$$

Step 3: Add third word:

$$\begin{array}{l} \texttt{0000 0100 0111 1101} \\ +\texttt{0000 1111 1100 0100} \\ \texttt{---- ---- ---- ----} \\ =\texttt{0001 0100 0100 0001} \end{array}$$

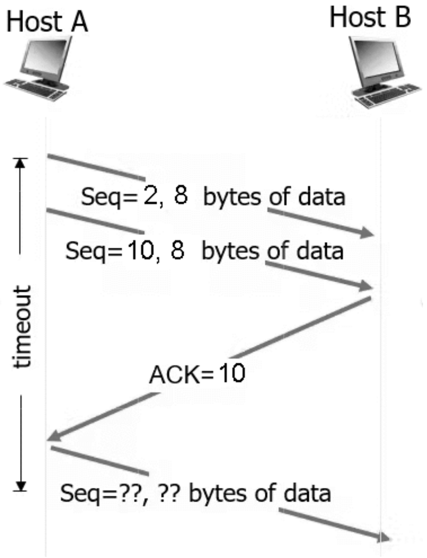
Step 4: Take the 1s complement (bitwise inversion)

$$\begin{array}{l} !\texttt{0001 0100 0100 0001} \\ \texttt{---- ---- ---- ----} \\ =\texttt{1110 1011 1011 1110} \end{array}$$

Part d) The receiver can add the received words using the 1's complement and compare the result with the checksum. (Alternatively: The receiver adds all received wordd, including the received checksum, using 1's complement addition. If the result is all 1s (1111 1111 1111 1111), the packet is assumed correct.)

Problem 3: Reliable Data Transmission (10 points)

In TCP, sequence numbers are used by the sender and receiver for implementing a reliable data-transfer service. An example is shown in the Figure below:

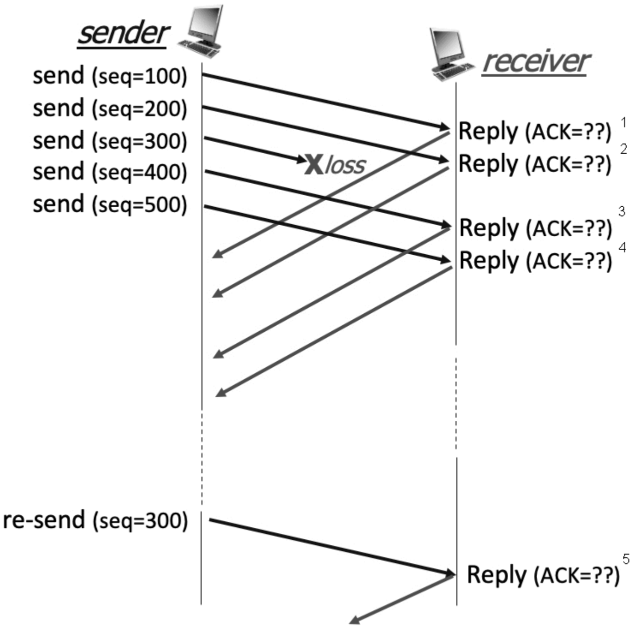


a) What is the next sequence number Seq sent from Host A to Host B (marked with Seq=??)? (1 point)

b) Assume the second transmission from Host A to Host B (Seq=10, 30 bytes of data) would have been successfully received by Host B. What would the ACK value shown in the Figure above be in this case? (1 point)

Problem 3 continued

Another example of sequence numbers is shown in the Figure below:



Assume now that each segment contains 100 bytes.

c) What are each of the ACK values for the first four ACKs (1-4)? (4 points)

d) What is the ACK value of the last ACK (5) sent by the receiver after receiving the re-sent segment? (1 point)

e) In TCP, the sender can perform a *fast re-transmit.* What is the necessary condition that triggers a fast re-transmit? (1 point)

f) Why did the sender send 5 segments at once, and not one segment at a time? What is the advantage of this behaviour? (2 points)

Answers:

a) Seq=10 (1 point)

b) ACK=18 (1 point) (or ACK=40 as there was a typo in the exam)

c) 1: 200 2: 300 3: 300 4: 300 (4 points)

d) 5: 600 (1 point)

e) Three identical ACK values (1 point)

f) Because the sender performs pipelining to increase the channel utilisation. (2 points)

Part B: Distributed Systems

You can achieve up to 65 points in Part B.

Problem 4: Multiple Choice (15 points)

For each of the following questions only one of the answers is correct. For each correctly answered question you will receive +1 point. If you are not sure about the correct answer, you can use the option "I do not know the answer", which will give you +0.25 points.

4a) Which statement about distribution transparency is correct? (1 point)

1) It guarantees that all distributed operations are executed instantaneously.
2) It removes all network delays.
3) It prevents failures from being visible to the user.
4) It hides differences in representation, access, and location of resources from users.
5) I do not know the answer (+0.25 points).

4b) Why are architectural styles useful when designing distributed systems? (1 point)

1) They prescribe the hardware specifications for each server.
2) They provide reusable design patterns that describe components, connectors, and interaction rules.
3) They are required by the TCP/IP protocol.
4) They eliminate the need for middleware.
5) I do not know the answer (+0.25 points).

4c) What is the key characteristic of a RESTful architecture? (1 point)

1) All resources are accessed through a uniform interface using standard operations like GET, POST, PUT, DELETE.
2) Each service defines its own proprietary operations.
3) The client maintains all session state.
4) It allows for referentially decoupled communication between peers.
5) I do not know the answer (+0.25 points).

4d) Which of the following best describes a process in a distributed system? (1 point)

1) A collection of unrelated threads running on a single core.
2) A hardware unit capable of executing one instruction per clock cycle.
3) A file containing executable code but not yet running.
4) A program in execution, with its own memory space and one or more threads.
5) I do not know the answer (+0.25 points).

4e) What is the main advantage of using threads instead of multiple processes in distributed applications? (1 point)

1) Threads provide hardware-level isolation between users.
2) Threads always execute faster than processes on single-core CPUs.
3) Threads prevent race conditions by design.
4) Thread creation and switching are cheaper since threads share the same address space.
5) I do not know the answer (+0.25 points).

4f) What is clock skew? (1 point)

1) The delay between two consecutive instructions in a thread.
2) The time offset introduced by the NTP handshake.
3) The time required for a message to travel through the network.
4) The difference in clock values between different computers caused by drift.
5) I do not know the answer (+0.25 points).

4g) A process P1 sends a message m at logical time 5. Process P2 receives m at its local logical time 3. According to Lamport's rules, what should the clock of P2 be set to before forwarding m to the application layer? (1 point)

1) C2 = 5 - 3 = 2
2) C2 = 3 + 1 = 4
3) C2 = max(3, 5) = 5
4) C2 = max(3, 5) + 1 = 6
5) I do not know the answer (+0.25 points).

4h) Which of the following best describes an identifier, a name, and an address? (1 point)

1) An identifier describes where an entity is located, a name identifies the entity, and an address is a human-readable alias.
2) An identifier uniquely denotes an entity, a name refers to it, and an address specifies its access point.
3) An identifier is always a network interface, while an address is a user name.
4) There is no practical distinction between them in distributed systems.
5) I do not know the answer (+0.25 points).

4i) What is the purpose of recursive DNS resolution? (1 point)

1) The resolver delegates the entire resolution process to another DNS server, which performs all subsequent lookups.
2) The resolver contacts each DNS server directly until the name is resolved.
3) The resolver stores all queried results permanently.
4) The resolver bypasses caching to ensure freshness.
5) I do not know the answer (+0.25 points).

4j) What is a consistency model in distributed systems? (1 point)

1) A physical replica of the database schema.

2) A contract between a data store and its clients that defines how read and write operations behave under concurrency.
3) A technique to improve load balancing among servers.
4) A caching mechanism to increase throughput.
5) I do not know the answer (+0.25 points).

4k) Why does the token-ring mutual exclusion algorithm prevent starvation, if the token is not lost? (1 point)

1) Because the coordinator ensures fairness.
2) Because the token circulates in a fixed order and eventually reaches every process.
3) Because processes can request the token multiple times in a row.
4) Because crashed processes always hold the token indefinitely.
5) I do not know the answer (+0.25 points).

4l) Assume you use a decentralised mutual exclusion algorithm with 8 replicas. A process needs at least 6 votes to access a critical shared resource. How many replicas would need to reset and forget that they already handed out a vote such that a second process could accidentally gain access to that shared resource too? (1 point)

1) 2.
2) 4.
3) 6.
4) 8.
5) I do not know the answer (+0.25 points).

4m) Which of the following describes best the relation between a fault, an error, and a failure? (1 point)

1) All three terms describe the same phenomenon.
2) Faults are user mistakes, errors are hardware defects, failures are external events.
3) A failure causes an error, which may lead to a fault.
4) A fault causes an error; an error may lead to a failure if it affects service delivery.
5) I do not know the answer (+0.25 points).

4n) Why is *open design* a desirable security principle in a distributed system? (1 point)

1) Because it prevents access to and from the dark web.
2) Because open systems require no authentication.
3) Because security mechanisms are publicly reviewable rather than secret.
4) Because in an open design there are no vulnerabilities.
5) I do not know the answer (+0.25 points).

4o) A malicious actor intercepts an online banking connection and replaces the bank's public key with his own, thereby decrypting and re-encrypting traffic unnoticed. Which security mechanism would have prevented this attack? (1 point)

1) Use of a digital certificate issued by a trusted certificate authority (CA).
2) Using a symmetric cipher such as AES instead of RSA.

3) Employing time redundancy to resend each packet twice.
4) Using a lava-lamp wall to create truly random public keys.
5) I do not know the answer (+0.25 points).

Answers:

4a) 4 It hides differences in representation, access, and location of resources from users.

4b) 2 They provide reusable design patterns that describe components, connectors, and interaction rules.

4c) 1 All resources are accessed through a uniform interface using standard operations like GET, POST, PUT, DELETE.

4d) 4 A program in execution, with its own memory space and one or more threads.

4e) 4 Thread creation and switching are cheaper since threads share the same address space.

4f) 4 The difference in clock values between different computers caused by drift.

4g) 4 C2 = max(3, 5) + 1 = 6

4h) 2 An identifier uniquely denotes an entity, a name refers to it, and an address specifies its access point.

4i) 1 The resolver delegates the entire resolution process to another DNS server, which performs all subsequent lookups.

4j) 2 A contract between a data store and its clients that defines how read and write operations behave under concurrency.

4k) 2 Because the token circulates in a fixed order and eventually reaches every process.

4l) 2 4.

4m) 4 A fault causes an error; an error may lead to a failure if it affects service delivery.

4n) 3 Because security mechanisms are publicly reviewable rather than secret.

4o) 1 Use of a digital certificate issued by a trusted certificate authority (CA).

Problem 5: Formal analysis of a centralised queue-based service (15 points)

You run a start-up for a really fancy cool app. One centralised authentication process on a server handles login requests from your users. Each request keeps the process busy for an average of 100 milliseconds. The server receives on average 5 login requests per second.

a) How many requests can the server process per second (processing capacity)? (1 point)

b) Compute the system utilisation. (2 points)

c) Compute the average number of requests in the system. (2 points)

d) Compute the average response time of the system. (2 points)

Your app went viral and is becoming extremely popular. You start worrying that the app might become unresponsive due to the many users that need authentication.

e) How many users can be served by the authentication server before the system becomes unresponsive? (2 points)

Users demand the ability to login using their social media accounts. You figured out that this would increase the average processing time to 200 ms per request. Continue assuming that the server receives on average 5 login requests per second.

f) What is the new average response time of your system? (2 points)

You (and your customers) are obviously not happy with the response time and you want to increase the speed of your system such that you get back to 100 ms processing time. Your idea is to replicate the process over several new processor cores. So instead of a 1-core server you order a new server with x CPU cores.

g) You know that 40% of your code cannot be parallelised. If you want to decrease processing time from 200 ms to 100 ms, how many CPU cores should you order? (4 points).

Answers: Given are the service time $S = 100$ ms $= 0.1$ s. Thus, the service rate $\mu = S^{-1} = \frac{1}{0.1} = 10$ req./s. The arrival rate is $\lambda = 5$ req./s.

a) $\mu = \frac{1}{S} = \frac{1}{0.1} = 10$ requests per second.

b) Utilisation: $U = \frac{\lambda}{\mu} = \frac{5}{10} = 0.5$.

c) Average number of requests in the system: $\bar{N} = \frac{U}{1-U} = \frac{0.5}{0.5} = 1$.

d) Average response time: $R = \frac{S}{1-U} = \frac{0.1}{0.5} = 0.2$ seconds.

e) The system becomes unresponsive when $U = 1 \Rightarrow \lambda = \mu = 10$ requests per second.

f) Average response time: $R = \frac{S}{1-U}$. With new $U = \lambda/\mu = 5/5 = 1$, the average response time will be very very large (infinite).

g) We use Amdahl's Law to calculate the necessary number of cores:
$\mathit{speedup} \leq \frac{1}{S+\frac{1-S}{N}} \Rightarrow 2 \leq \frac{1}{0.4+\frac{0.6}{N}} = \frac{N}{0.4N+0.6}$
$\Rightarrow 0.5N \geq 0.4N + 0.6 \Rightarrow 0.1N \geq 0.6 \Rightarrow N \geq 6$
You should order 6 or more CPU cores.

Problem 6: Design Trade-offs in Service Communication Architectures (10 points)

The app your super cool start-up company is developing is part of a distributed e-commerce system with several interdependent services: `UserAuth`, `Payment`, `ProductCatalog`, and `Shipping`. You and your development team are considering whether to use direct service-to-service calls using wrappers.

a) Draw the architecture *without* using a broker. Assume that each service must communicate with every other service directly (i.e., full interconnection). Label the services. (2 points)

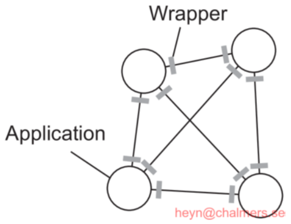
b) Now draw the architecture using a central broker instead. Explain in one sentence how this reduces referential coupling in the system. (2 points)

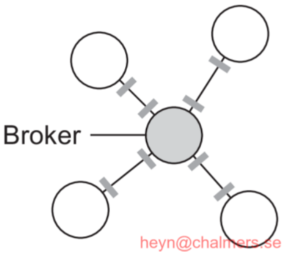
c) How many wrappers can be saved by using a central broker? What is a major disadvantage of using a central broker? (2 points)

Assume now your team chose a broker design. The broker can handle 2000 messages per second. Each service sends 5 messages per second to each of the other services.

d) What is the maximum number of services the broker can support before becoming overloaded? (4 points)

Answers:

a) 

b) 

Each process only needs to know how to communicate with the broker, not with each other process which reduces referential coupling between the processes.

c) Without broker: $N \cdot (N-1) = 4 \cdot 3 = 12$. With broker: $2N = 8$. We can save 4 unique connections. However, a broker introduces a single-point of failure for the entire system.

d) Each service sends 5 messages per seconds to three other services: $\text{Msg/service} = 5 \cdot (N-1)$. Therefore, the total load on the broker from all services: $\text{Load} = N \cdot 5 \cdot (N-1)$. We want to ensure that $\text{Load} \leq 2000$. Then:
$5N(N-1) \leq 2000 \Rightarrow N(N-1) \leq 400$.
Now solve $N^2 - N - 400 = 0 \Rightarrow N_{1,2} = 0.5 \pm \sqrt{0.25 + 400} \approx 20.5$
The maximum number of services is 20.

Process P_1	**Process** P_2	**Process** P_3
$x \leftarrow 1$;	$y \leftarrow 1$;	`print(x, y)`

Problem 7: Sequences and consistency (15 points)

Assume we have three processes and the following executions in each process:

a) How many possible combination of execution exist? (2 point)

b) Create a table in which you show all executions combinations and the resulting prints. (6 points)

c) How many different print output sequences can occur? (2 points)

d) Which of the execution combinations are sequentially consistent? (1 point)

e) Assume that printing a value can only occur after all writing processes are done. Mark in your table which execution combinations violate this assumption. How many execution combinations remain? (2 points)

f) How could you ensure that a print does not occur before all writing operations are finished? (2 points)

Answers:

a) $n_{tot} = 3! = 3 \cdot 2 \cdot 1 = 6$

b)

Ex. 1	**Ex. 2**	**Ex. 3**	**Ex. 4**	**Ex. 5**	**Ex. 6**
P_1: $x \leftarrow 1$;	P_2: $y \leftarrow 1$;	P_3: `print(x,y);`	P_1: $x \leftarrow 1$;	P3: `print(x,y);`	P_2: $y \leftarrow 1$;
P_2: $y \leftarrow 1$;	P_1: $x \leftarrow 1$;	P_1: $x \leftarrow 1$;	P3: `print(x,y);`	P_2: $y \leftarrow 1$;	P3: `print(x,y);`
P3: `print(x,y);`	P3: `print(x,y);`	P_2: $y \leftarrow 1$;	P_2: $y \leftarrow 1$;	P_1: $x \leftarrow 1$;	P_1: $x \leftarrow 1$;
Prints: 11	*Prints:* 11	*Prints:* 00	*Prints:* 10	*Prints:* 00	*Prints:* 01

c) Four different print outputs can occur, depending on the execution sequence: `00`, `01`, `10`, and `11`.

d) All combinations are sequentially consist.

e) Only two execution sequences are causally consistent.

f) We could use a mutual exclusion mechanism to lock variables during a writing operation. Furthermore, we must ensure that processes which write execute before processes that read.

Problem 8: Public-private key pairs (10 points)

Suppose Bob wants to send an e-mail to Alice. Alice has a public-private key pair (K_a^+,K_a^-), and **Bob has Alice's certificate** and therefore also Alice's public key. But Bob does not have a public-private key pair himself. Alice and Bob (and the rest of the world) share the same hash function $H(\cdot)$.

a) Is it possible to design a scheme so that Alice can verify that Bob created the message? If yes, provide a block diagram that shows how for Alice and Bob. If not, explain in one sentence why not. (2 points)

b) Is it possible to design a scheme that provides confidentiality for sending the massage from Bob to Alice? If yes, provide a block diagram that shows how for Alice and Bob. If not, explain in one sentence why not. (3 points)

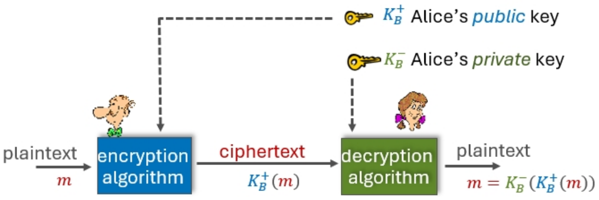
c) Assume for the rest of this problem that **Bob also has a public-private key pair (**K_b^+**,**K_b^-**) and that Alice also has Bob's certificate**. Explain briefly how Bob can ensure the *integrity* of the message using a hash function and his private/public key pair. How can Alice test the message *integrity*? (3 points)

d) Can this approach even be used to ensure *authenticity* of the message from Bob to Alice? Argue briefly why or why not. (2 points)

Answers:

a) No, without a public-private key pair or a pre-shared secret, Alice cannot verify that Bob created the message. Anyone could have composed and encrypted the message.

b) Yes, Bob encrypts the message with Alice's public key (from her certificate) and sends the encrypted message to Alice.



c) Bob produces a hash of the message which he encrypts with his own private key. Alice decrypts the hash using Bob's public key, applies the hash function to the message and obtains an own hash. She can then compare her hash with the received hash from Bob. If they are the same, the message's integrity is given.

d) Yes, because the encrypted hash could only have been created with Bob's private key, and Alice uses Bob's public key from his certificate (which vouches for the public key owner).