

NIST - Second Partial. 2025-01-13

This exam is worth 10 points, and consists of 22 questions. Each question poses 4 alternatives and has only one correct answer. Once discarded the two worst answers, each correct answer earns 0.5 points, and each error deducts 1/6 points. You must answer on the answer sheet.

A

1. Consider the following JavaScript program:

```
const z = require("zeromq")
const r = z.socket("router")
r.bindSync("tcp://127.0.0.1:8999")
r.on("message", (x, y, z) => {
  console.log(x+ " "+y+ " "+z)
  r.send([x, parseInt(y+"")*parseInt(z+"")])
})
```

Choose which statement describes the behavior of that program:

- A. If the fifth line displays the string `C1 4 5` when receiving a message, then whoever sent that message could have used an `s.send([4,5])` to do so.
- B. If the fifth line displays the string `C1 4 5` when receiving a message, then whoever sent that message could have used an `s.send(["C1",4,5])` to do so.
- C. If the fifth line displays the string `C1 4 5` when receiving a message, then whoever sent that message could have used an `s.send("C1 4 5")` for this.
- D. None of the other statements are correct.

2. Consider the following JavaScript program, used to run a server process `S` with which some client process `C` will interact:

```
const z = require("zeromq")
const r = z.socket("router")
r.bindSync("tcp://127.0.0.1:8999")
r.on("message", (x, y, z) => {
  console.log(x+ " "+y+ " "+z)
  r.send([x, parseInt(y+"")*parseInt(z+"")])
})
```

Select the correct statement:

- A. `C` could have used two sockets (e.g., PUSH and PULL) to send requests to and receive responses from `S`.
- B. There cannot be more than one client process `C` connected to `S`, since `S` has only defined one listener for the `message` event.
- C. The parameter `x` defined in the fourth line of that program will receive the identity of the socket used in `C` to connect to `S`.
- D. `C` could have used a PUSH socket to receive responses from `S`.

3. When sending messages using sockets of type DEALER...:

- A. All other statements are false.
- B. A round-robin sending policy is followed.
- C. You can choose, using the first segment of the message, to which other socket connected to the DEALER the message is being sent.
- D. The socket always automatically adds an additional first segment with the identity of the sending socket.

4. May a Dockerfile have no `CMD` or `ENTRYPOINT` instruction?

- A. No, because in that case the `docker build` command produces an error and does not build any image.
- B. Yes, because the `WORKDIR` instruction has a similar purpose and can replace them.
- C. Yes, because the image to be generated could be used as a base for other Dockerfiles that do use `CMD` or `ENTRYPOINT`.
- D. No, because if those instructions are not there, the `RUN` instructions have no effect.

(Questions 5 to 11 refer to this description) Let us consider a system with three components (A,B,C) whose

code is shown subsequently and that connect to each other as the figure depicts. To avoid the loss of the message published by A, there is a delay before that sending: all the other instances are started in that interval.

A.js

```
const zmq = require('zeromq')
let sin = zmq.socket('pull')
let sout = zmq.socket('pub')
sin.bind('tcp://*:1111')
sout.bind('tcp://*:2222')
sin.on('message', msg => {
  console.log('Recibido '+msg)
})
setTimeout(() => {
  console.log('empezamos')
  sout.send('hola')
}, 5000)
```

B.js

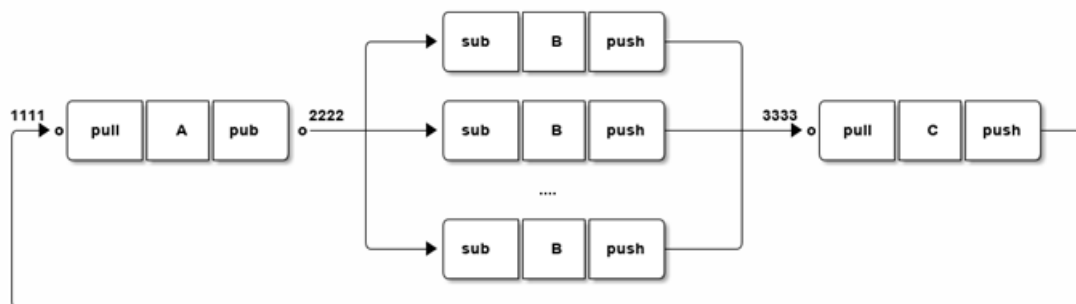
```
const zmq = require('zeromq')
let sin = zmq.socket('sub')
let sout = zmq.socket('push')
sin.connect(process.argv[2])
sout.connect(process.argv[3])
sin.subscribe('')

sin.on('message', msg => {
  console.log('recibido '+msg)
  sout.send(msg)
})
```

C.js

```
const zmq = require('zeromq')
let sin = zmq.socket('pull')
let sout = zmq.socket('push')
sin.bind('tcp://*:3333')
sout.connect(process.argv[2])

sin.on('message', msg => {
  console.log('recibido '+msg)
  sout.send(msg)
})
```

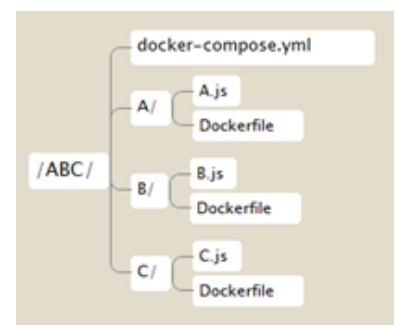


To start this system, we may use several terminals, one per instance, as follows:

```
node A.js
node B.js tcp://localhost:2222 tcp://localhost:3333
node B.js tcp://localhost:2222 tcp://localhost:3333
node B.js tcp://localhost:2222 tcp://localhost:3333
node C.js tcp://localhost:1111
```

The result is correct, independently on the number of B instances, and on the sequence in which those instances start. We want to automate that deployment using docker (i.e., all instances should run in containers on the same host):

- Let us consider that the `tsr-zmq` image is available (base of the images for A, B and C)
- In component C, the port for its PULL socket is 3333, and it receives from the command line the URL of A's PULL socket.
- B receives from the command line the URLs of A's PUB socket and C's PULL socket.



Files have been organised following the directory hierarchy shown on the right. Please, answer questions 5 to 11 on that docker-based deployment.

5. The file `/ABC/B/Dockerfile` must contain the following line:

- `CMD node B.js tcp://localhost:1111 tcp://localhost:3333`
- `CMD node $URL_A $URL_B`
- None of the remaining answers are correct
- `CMD node B.js`

6. We want to launch an execution with 1 instance of A, 5 instances of B, and one instance of C. To do this we must execute the command:

- `docker build B 5`
- `docker compose up --scale B=5`
- `docker compose up`
- `docker run --scale B=5`

7. If we are located in the `/ABC/` directory, choose which of the following commands does not run correctly

- A. `docker build -t a .`
- B. `docker compose up`
- C. `docker images`
- D. `docker ps`

8. Regarding the content of the `docker-compose.yml` file

- A. In the section corresponding to component A, a `links` section should appear
- B. In the section corresponding to component C, a `links` section should appear
- C. In the section corresponding to component A, an `environment` section should appear
- D. In the section corresponding to component B, an `expose` section should appear

9. Assuming we have launched a container running an instance of A, if we want to find out its IP...

- A. We must run `docker ps` to find out the identity of said container followed by `docker inspect identity`.
- B. Running the container using `docker run A` results in the IP associated with it.
- C. We must run `docker images` to find out the identity of the image of A, followed by `docker rmi identity`.
- D. It is internal information that we cannot access.

10. The order of deployment of these containers must be:

- A. First A, then C and then B
- B. First C, then A and then B
- C. First A, the rest in any order
- D. First C, the rest in any order

11. About the respective Dockerfiles...

- A. All options are correct
- B. A's should contain `EXPOSE 1111 2222`
- C. B's should contain `EXPOSE 1111 2222`
- D. C's should contain `EXPOSE 1111 2222`

12. Let us consider that we add a ROUTER socket bound to port 3333 of component A in the system described in previous questions, with appropriate changes to the A.js code, and we want it to be accessible from the outside via port 99 on the host.

After generating the new image, we will need to...

- A. We can't co-deploy the new system because we need to manually start A with `docker run -p 99:3333 ...`
- B. Modify `docker-compose.yml` to place an appropriate `ports` entry on service A
- C. Build the images for the rest of components B and C
- D. It won't work because using a single port is incompatible with a multi-queue socket

13. Select the correct statement about passive replication:

- A. It allows server replicas to execute each client request locally without having to send the generated modifications to the other replicas.
- B. It allows a single replica to execute the requested operation and broadcast the modifications to the others.
- C. It generally tolerates arbitrary failures.
- D. It requires that each client request reach all server replicas before any of them begin processing that request.

14. To deploy the CBWL system, a `docker-compose.yml` file is used, in which, within the section corresponding to the `bro` component, these lines appear:

```
environment:
  - LOGGER_HOST=log
  - LOGGER_PORT=9995
```

Select the correct option regarding the implications of that `environment` subsection:

- A. The `wor` and `cli` components must be started before `bro`, as they are not mentioned in that subsection
- B. In the Dockerfile of the image used to generate the `bro` component, the environment variables `LOGGER_HOST` and `LOGGER_PORT` are used
- C. The `bro` component must be started during deployment before the `log` component
- D. The `log` component must connect to a port, which cannot be `9995`, of the `bro` component

15. The `docker commit` command can be used to:

- A. All other options are incorrect
- B. Build a docker image from a Dockerfile
- C. Generate a docker image from the current state of a given container
- D. Start running a container using the image specified in its arguments

16. *If an execution respects causal consistency, then it will also respect this consistency model...*

- A. Strict
- B. FIFO
- C. Cache
- D. Sequential

17. *Let us consider a system consisting of three processes P1, P2 and P3, where the following execution has occurred: $\overline{W1(x)2, W2(x)1, R1(x)1, R2(x)2, R3(x)2, R3(x)1}$. This execution respects, among others, the following consistencies:*

- A. FIFO and causal
- B. FIFO and cache
- C. Only cache consistency
- D. Strict and sequential

18. *When considering the CAP Theorem for a scalable service, it is often recommended to drop strong consistency. Based on that, select which consistency model could not be supported when there are connectivity problems that generate a network partition:*

- A. In general, any "fast" model
- B. FIFO
- C. Causal
- D. Sequential

19. *Select which characteristic is respected in multi-master replication:*

- A. It generally offers sequential consistency
- B. All replicas execute the sequence of instructions for each operation requested by client processes
- C. Can tolerate arbitrary failures
- D. It is generally more efficient than active and passive replication models

20. *The `tsr-zmq` image, mentioned throughout practice 3, must be built using the Dockerfile provided in the practice bulletin. Choose which of the following statements about that Dockerfile is true.*

- A. It does not include the `tsr.js` library in the produced image, but it does include the ZeroMQ library.
- B. It is based on an Ubuntu image
- C. It provides support for NodeJS
- D. All answers are correct

21. *The deployment performed in the first session of practice 3 was...*

- A. A chat system with PUB/SUB and PUSH/PULL patterns
- B. A client-broker-worker system with dual broker
- C. None of the other options are correct
- D. A client-broker-worker system with a fault-tolerant broker

22. *Select the correct statement about the `logger` component that we added to the CBW pattern to generate the CBWL pattern:*

- A. It adds each message to a file that is part of a volume accessible from the host machine.
- B. It keeps only the last message received in the file (each message replaces the previous content)
- C. It dumps the messages it receives through the PULL socket to an internal file, accessible only from the container that runs the `logger` instance.
- D. It broadcasts the messages that arrive from the broker to the rest of the components.