1. Pasii unei aplicatii server UDP

d. socket, bind, recvfrom, sendto, close

2. inet\_aton()

c.converteste adresa IP in format binary, big endian

3. listen()

b.realizeaza o coada de conexiune

c. este apel de system

4. connect()

c.se foloseste de aplicatie client TCP

5.Ce face bind?

a.se foloseste in aplicatia server

6. Ce semnificatie are campul2 de la socket?

d.specifica tipul de socket

7.Pasii unei aplicatii server TCP

a.gethostbyname, socket, bind, listen, accept, send, close

7.1. Pasii unei aplicatii client TCP

a.gethostbyname, socket, connect, send, recv, close

8. getportobyname()

a.foloseste fisierul /etc/protocol

b.converteste numele protocolului intr-un numar recunoscut de SO

9.inet\_addr()

Converteste adresa IPv4 in format binary, big endian

10. Ce face socket(AF\_INET, SOCK\_DGRAM, 0)?

b.deschide un socket pe UDP

d.deschide un socket care suporta toate protocoalele familiei ipv4

11.htonl()

c.converteste un intreg fara semn format scurt din ordinea de retea in cea de host

12.accept()

a.se foloseste de aplicatie server TCP

d.returneaza un nou socket pentru aplicatiile cu conexiune

13.sendmsg()

a.se foloseste pt trimiterea de mesaje in aplicatiile client/server UDP

b.este apel de system

c.se foloseste pt trimitearea de mesaje in aplicatiile client/server TCP

14.inet\_ntoa()

a. converteste adresa hostului din format binary, big endian, in format zecimal IPv4

d. este functie de biblioteca

15. sendto()

a.este apel de system

c. se foloseste pentru trimiterea de mesaje in aplicatiile client/server UDP

16.Ce semnificatie are campul1 de la socket?

b.specifica tipul de familie de adrese(AF\_inet, AF\_local)

17.Ce face server.sin\_addr.s\_addr = INADDR\_ANY

a.este gesionat de SO

18. gethostbyname()

b.este functie de biblioteca

c.foloseste fisierul /etc/hosts

19.ntohl()

a.converteste un intreg fara sem format lung din ordinea de retea in cea de host

20. Ce face server.sin\_family = AF\_INET?

c.stabileste familia de adrese de tip ipv4

21. Pasii crearii unui server UDP:

Creare socket, se obt adresa (ip, port), de det portul, se face receptive, se inchide sesiunea

22. Pasii crearii unui client UDP:

Creare socket, obt adresa ip si port, se trimite mesajul serverului (sendto), se inchide clientul

client\_sock, client\_addr = serv\_sock.accept() *#folosim metoda accept pentru a primi conexiuni. Prin aceasta metoda primim un obiect de tip socket nou pentru a trimite si primi date si adresa clientului.*

client\_sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) *#cream un socket, ca parametru ii dam familia de protocoale INET si protocolul de transmisie (TCP – in cazul dat)*

*server tcp: creare socket, bind, listen, accept, prelucrare?, transmitem date clientului, inchidem serverul*

*client tcp: creare socket, atribuim port si ip, connect la socket, prelucrare, inchidem clientul*