

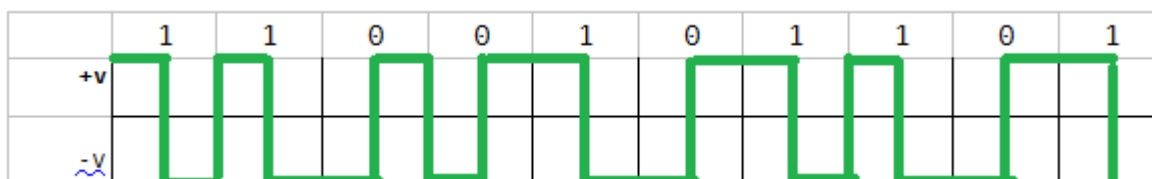
Hálózatok 1.ZH  
**Minta ZH 2016**

Név: \_\_\_\_\_

Neptun kód: \_\_\_\_\_

1. **Feladat:** Hogyan működik az PE kódolás? Kódold a lenti bitsorozatot eszerint a kódolás szerint. Az eredmény bemutatására használd a sablont:

Az 1 jel átviteli idejének felénél feszültség szint váltás történik lefelé, a 0 jel átviteli idejének felénél feszültség szint váltás történik felfelé.



2. **Feladat:** Hogyan íratnád ki az arp táblát? Mit tartalmaz? Hogyan kerül bele új bejegyzés? Hogyan kerülhet ki onnan?

arp, (arp -n – numerikus forma, Windows alatt arp -a)

A tábla fizikai cím IP cím párokat tartalmaz.

Belekerülhet új bejegyzés dinamikusan bármiféle adatkapcsolati rétegbeli kommunikáció eredményeképp, vagy statikusan az arp -s IP MAC paranccsal (ahol IP egy ip cím és MAC a hozzá tartozó fizikai cím.)

Kikerülhet a bejegyzés a táblából, ha valamiért kilépünk a hálózatból. Statikus bejegyzés törölhető az arp -s paranccsal.

3. **Feladat:** Az alábbi karaktorsorozatok közül melyek lehetnek fizikai címek? A lehetséges címek közül melyek jelölnek azonos gyártóhoz tartozó eszközt?

```
30:e4:db:68:e1:15    00:17:31:68:e1:15    00:1a:4d:68:e1  
00:1a:4d:a3:c8:49    00:16:cg:ac:77:46    00:17:ac:77:46  
00:1a:4c:7f:88:9f    00:1a:4d:94:55:92
```

Az áthúzottak nem lehetnek fizikai címek. A harmadik oszlopban lévők túl rövidek, a középső pedig tartalmaz egy g betűt.

~~Nincs két azonos gyártótól származó eszköz a listában.~~ Az első oszlop második és a második oszlop harmadik címe származik azonos gyártótól. (Az első három bajtnak, azaz 12 hexadecimális karakternek kellene egyeznie)

4. **Feladat:** Hogyan íratnád ki a route tábla tartalmát numerikus formában?

route -n, vagy netstat -rn

Az alábbi route tábla szerint melyik sor alapján melyik átjárón át, milyen interfészt

használva küldjük ki a csomagot a csomóponttól, ha a csomag célcíme: 183.65.176.34

célháló	átjáró	netmaszk	interfész
193.6.181.0	0.0.0.0	255.255.255.0	eth0
183.65.160.0	193.6.181.128	255.255.240.0	eth0
172.16.0.0	0.0.0.0	255.240.0.0	eth1
0.0.0.0	193.6.181.1	0.0.0.0	eth0

1. sor:

183.65.176.34

& 255.255.255.0

-----

= 183.65.176.34 != 193.6.181.0 → nem jó

2. sor

~~183.65.176.34 = 183.65.10101000.34~~

183.65.176.34 = 183.65.10110000.34

& 255.255.240.0 = 255.255.11110000.0

-----

~~= 183.65.10100000.0 = 183.65.160.0 → jó~~

= 183.65.10110000.0 = 183.65.160.0 → nem jó

~~A csomagot a második sor alapján küldöm ki a 193.6.282.228 193.6.181.128 átjáróra az eth0 interfészen át.~~

3. sor

183.65.176.34 = 183.01000001.176.34

& 255.240.0.0 = 183.11110000.0.0

-----

183.64.0.0 → nem jó

A csomagot a 4 (utolsó) sor alapján küldöm ki az alapértelmezett (193.6.181.1) átjáróra, az eth0 interfészen át.)

~~(Ha csak az utolsó sorra találnék illeszkedést pl. 166.25.84.14 IP cím esetén, akkor azt írnám: A csomagot a 4 (utolsó) sor alapján küldöm ki az alapértelmezett (193.6.181.1) átjáróra, az eth0 interfészen át.)~~

5. **Feladat:** Adott a szabványos osztályba sorolható 163.29.0.0 címtartomány. Alakíts ki a tartományon belül megfelelő alhálózatokat a következők szerint.

- Összesen 4 igény érkezik: 2000, 1500, 3000 és 6000 csomóponttal
- Az igények kis időkülönbséggel érkeznek. (→ sorba lehet rendezni)
- Minden alhálózatához a lehető legkisebb kiosztható területet rendel (azaz ne négy azonos méretűt hozz létre, hanem ahogy az órán csináltuk, szépen a lehető legkevesebb csomópontot elpazarolva)

Add meg az egyes alhálók hálózati azonosítóit netmaszkkal együtt.

A hálózat B típusú (binárisan 10....-val kezdődik), azaz a prefix 16.  
 2000,1500,3000,6000 → sorba rendezve → 6000, 3000, 2000, 1500  
 Hány bit kell a csomópontok azonosítására?

6000 <  $2^{13}$  → 13 bit kell csomópont azonosításra

→ (32-16)-13=3 bit marad alháló azonosításra

3000 <  $2^{12}$  → 12 bit kell csomópont azonosításra

→ (32-16)-12=4 bit marad alháló azonosításra

2000 <  $2^{11}$  → 11 bit kell csomópont azonosításra

→ (32-16)-11=5 bit marad alháló azonosításra

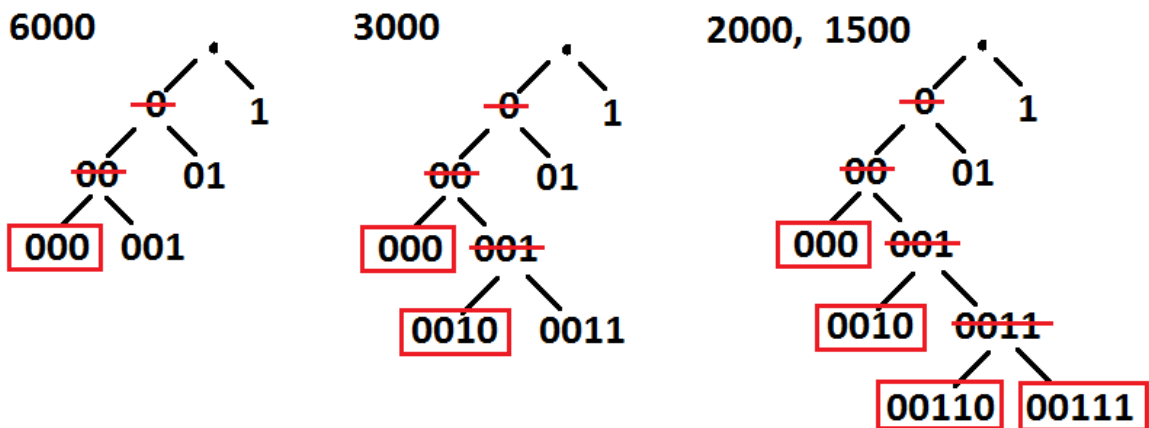
1500 <  $2^{11}$  → 11 bit kell csomópont azonosításra

→ (32-16)-11=5 bit marad alháló azonosításra

Az első azonosító mindig csupa 0

6000 → 000

keressük meg a többi azonosítót is:



A megoldás:

163.29.00000000.0 → 163.29.0.0/19

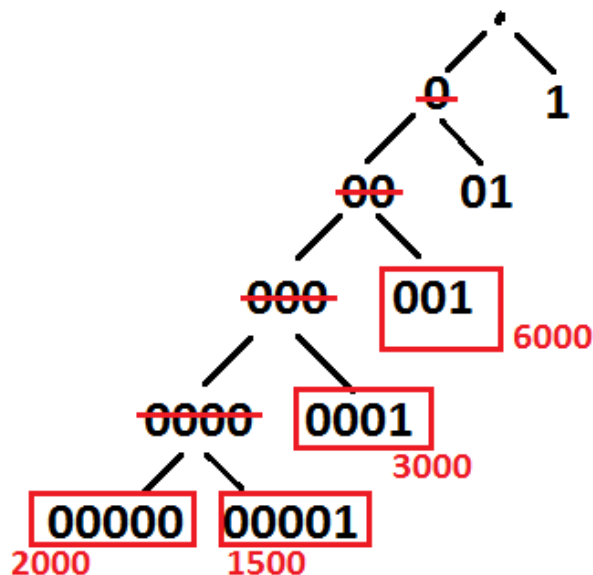
163.29.00100000.0 → 163.29.32.0/20

163.29.00110000.0 → 163.29.48.0/21

163.29.00111000.0 → 163.29.56.0/21

Ha ugyanez lett volna a feladat, de nagy időkülönbséggel, akkor nem rendezhettünk volna sorba.

Ekkor az azonosítók valahogy így jöttek volna ki:



**A megoldás pedig ez lett volna:**

163.29.00000000.0 → 163.29.0.0/21  
163.29.00001000.0 → 163.29.8.0/21  
163.29.00010000.0 → 163.29.16.0/20  
163.29.00100000.0 → 163.29.32.0/19

6. **Feladat:** Kiszolgálható-e a maradék csomópontokból annak az igénylőnek az igénye, aki 35000 csomópontot szeretne csatlakoztatni hálózatához?

Nem, mert a legnagyobb egyben kiosztható darab nagyjából 32000 csomópontot tartalmaz csak. Ha mondjuk A osztályúból indultunk volna ki, akkor még kis lehetne osztani. Egyébként összességében maradt ki elegendő csomópontunk (kb 48000 maradt), csak ezek nem oszthatók ki egyetlen hálózatként.

7. **Feladat:** Oszd a 6000 csomópontot tartalmazó hálózatot négy egyenlő részre.

Add meg a kapott hálókat

- hálózati azonosítói netmaszkkal együtt, valamint
- a hozzájuk tartozó broadcast címeket.

Legyen a kis időkülönbséges eset.

Négy hálóhoz további két bitre lesz szükség a hálóazonosítóban.

163.29.0.0/19 →

163.29.0.0/21	BC: 163.29.7.255
163.29.8.0/21	BC: 163.29.15.255
163.29.16.0/21	BC: 163.29.23.255
163.29.24.0/21	BC: 163.29.31.255

Ha kérdésetek van, vagy ha esetleg hibát találtok, írjatok nyugodtan mailt.

Volt is aki hibát talált. Az update-et láthatjátok pirossal áthúzva.