Információs Rendszerek Üzemeltetése - Vizsgasegédlet

készítette: Bartók Tamás (2013.06) és TI

Elöljáróban: Azért készült ez a kis elméleti agytorna, mert a korábbi években nagy mértékű volt a bukási arány vizsgákon. A tapasztalatok szerint sok helyen tévesek/pontatlanok a korábbi Wikis anyagok, egy mondatos kérdésekre adott válaszok. Ez az anyag ezeket próbálja meg kijavítani. Nem csak korábbi évek kérdéseit dolgozza fel, hanem saját szubjektív meglátásom szerinti értelmes, lehetséges kérdéseket is belefaragtam.
FONTOS!!! Nem feltétlenül tartalmaz ez sem 100%-os megoldásokat és az, hogy minél jobb legyen, a ti feladatotok is, hogy szerkesszétek, ha találtok valami csiszolnivalót.
Nem csak vakon a Wikipédiára és a diákra támaszkodva próbáltam összeállítani (azért 85%-ban diákból), hanem több hitelesebb forrásból. Remélem segít, ha már csak pár pontot is dob a vizsgán, már megérte. Fogyasszátok egészséggel.
// A KOCKÁZATOK ÉS KELLÉKHATÁSOK TEKINTETÉBEN KÉRJÜK, OLVASSA EL A RETEKTÁJÉKOZTATÓT VAGY MÉRGEZZE MEG KEZELŐORVOSÁT, ÓVSZERÉSZÉT! //

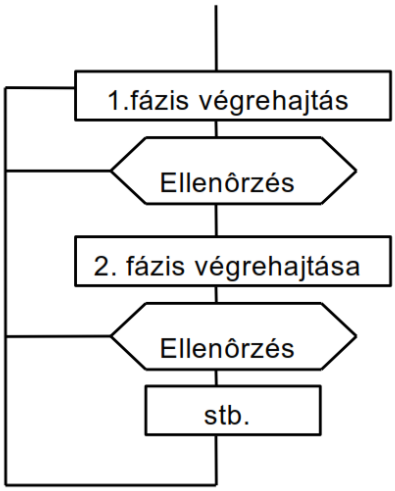
A doksi egyes részeinek érthetőbbé tétele miatt köszönet Haraszin Péter, Somogyi Péter és Bodnár Dániel kollegáknak.

A módosításokról: Kérlek titeket, hogy ide írjátok be, hogy mikor módosítottátok utoljára, esetleg melyik részt/részeket:
Megszületése: 2013.06.17

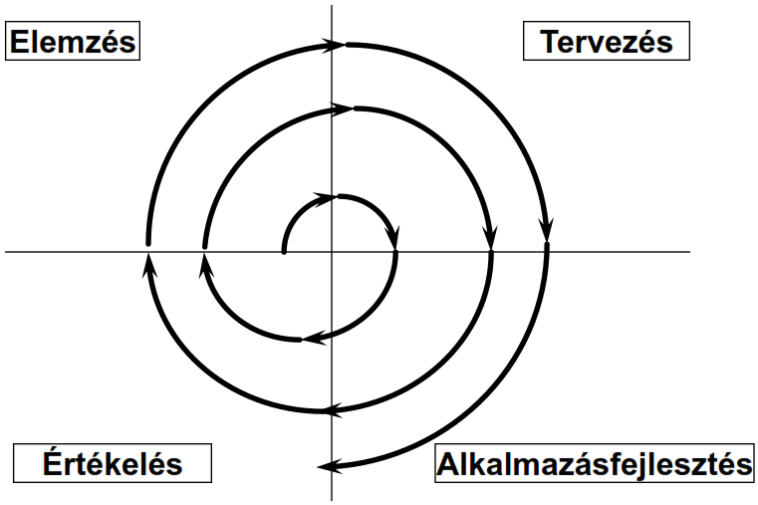
Tartalomjegyzék:

Bevezető diasor (IRU_2013_1):

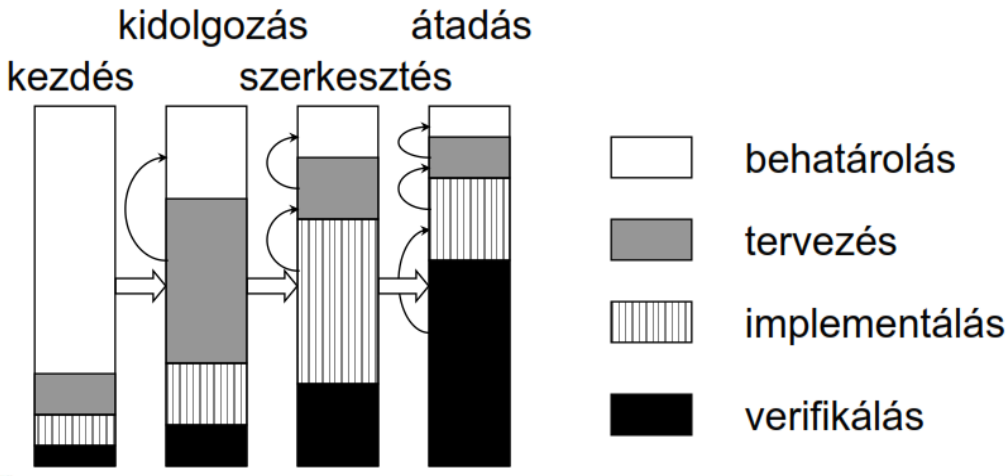
1. *Az információs rendszerek kialakulásának fontosabb szereplői:*
   - Felhasználó (körülötte forog a világ)
   - Folyamattervező (inkább business mint IT)
   - Rendszertervező (valahol az IT és a business között, IT beütéssel)
   - Programozó
   - Tesztelő
   - Üzemeltető (rendszeradminisztrátor)
2. *Mit értünk életciklus alatt?*
   - Egy rendszer teljes élettörténete az ötlet megszületésétől a használatból való kivonásig. Hasznos, mert mások tapasztalatára építhetünk, módszeresen végiggondoljuk a feladatokat.
3. *Vízesés modell fázisai (5 db):*
   - Analízis, Tervezés, Implementáció, Teszt, Integrálás
   - *info: A fázisok egymásra épülnek. Az egyes fázisok végén döntési pontok (mérföldkövek) vannak, amikben értékelik, elemzik az előző fázis eredményeit.



4. *Inkrementális modell lényege?*
   - A teljes program egyenként különálló és működő kisebb programokból épül fel. A kezdeti tervezési fázisban az akkor elkészülő első kis programot teljesnek feltételezzük és utána fokozatosan fejlesztjük és adunk hozzá újabb, az előzővel kompatibilis és működő programokat.
5. *Spirál modell lényege?*
   - A spirál modell iterációkból épül fel, melyek ismétlődnek a projekt során. Egy hibrid modell, mivel megtartotta a vízesés modell előnyeit és nem zárja ki a prototípus készítésének lehetőségét sem.



6. *<u>Érettség modell 4 fázisa?</u>*
   - kezdés, kidolgozás, szerkesztés, átadás

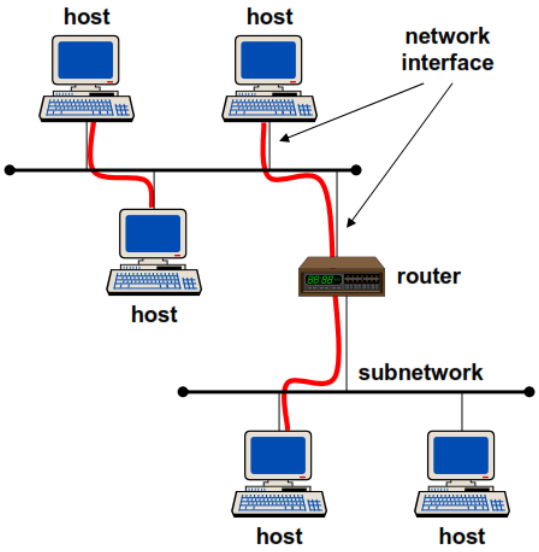


IRU_2013_Ism_Hal_Serv_deskt_1 diasor, kulcsszavak OSI/ISO, NA(P)T, DNS, szerverek, frissítés:

7. *Sorolja fel az OSI 7 rétegét:*
    1. Fizikai réteg (Az eszközök közötti fémes vagy optikai átvivő közeg; hub)
    2. Adatkapcsolati réteg (Interfész (MAC) szintű címzés, folyamvezérlés, (bit)hibadetektálás, hibajavítás (bridge, switch))
    3. Hálózati réteg (Logikai címzés (pl. IP-címek) és azon alapuló irányítás, útvonalválasztás (routerek))
    4. Szállítási réteg (Végpontok közötti adatátvitel, megbízhatóság, virtuális áramkörök (pl. TCP kapcsolatok, port számok))
    5. Viszony réteg (Kommunikációs viszonylatok vezérlése (SCP – Session Control Protocol))
    6. Megjelenítési réteg (Adat megjelenítése és kódolása. Adatformátumok (pl. MPEG), karakterkódolás, tömörítés, titkosítás)
    7. Alkalmazási réteg (Alkalmazási protokollok, pl. SMTP (email – Simple Mail Transfer Protocol), HTTP (web), FTP (fájltranszfer))
8. *Mi a protokoll?*
    - Szabályok gyűjteménye, mely vezényli a kommunikációt hálózati elemek között.
9. *Hanyadik rétegbeli eszköz a hub, és mi a feladata?*
    - A hub (kb. középpont, csomópont; magyar neve: többportos jelismétlő) Layer 1-es eszköz (fizikai rétegbeli), feladata, hogy a bemenetére érkező jelet minden portra továbbítsa (broadcast eszköz).
10. *Hanyadik rétegbeli eszköz a bridge, és mi a feladata?*
    - A bridge (híd) Layer 2-es eszköz (adatkapcsolat rétegbeli), feladata MAC-cím alapú irányítás, keret analízis.
11. *Hanyadik rétegbeli eszköz a switch, és mi a feladata?*
    - A switch (adatátviteli kapcsoló) Layer 2-es eszköz (adatkapcsolat-rétegbeli), DE vannak magasabb rétegbeli switch-ek is (4. rétegbeli NAT switch, 7. rétegbeli). Feladata - a 2. rétegbelinek - MAC-cím-alapú irányítás, a bemenő jelet a megfelelő portra irányítja.
12. *Hanyadik rétegbeli eszköz a router, és mi a feladata?*
    - A router (útválasztó) Layer 3-as eszköz (hálózati rétegbeli), feladata útvonalválasztás/két vagy több alhálózat összekötése, irányítás IP-cím alapján.

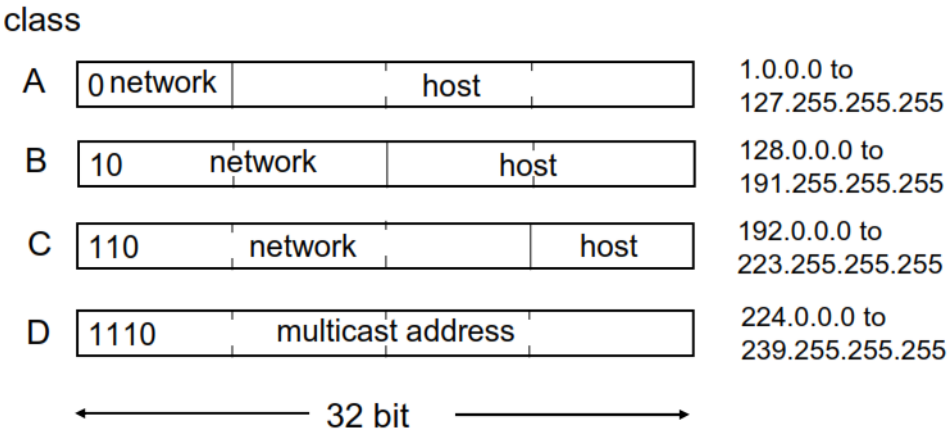
Sorolja fel az IP-hálózat elemeit (4 db):

- Host (kommunikációs végpont),
- Alhálózat (subnetwork; fizikai hálózat, a hozzákapcsolt csomópontok közvetlenül tudnak kommunikálni),
- Hálózati interfész (csomópont kapcsolódási pontja az alhálózathoz),
- Útvonalválasztó (router; közvetítő a külön alhálózaton található, kommunikáló hosztok között)



13. *Mennyi és milyen IP-címosztályok vannak, mi a hálózati cím, és hogyan határozható meg?*
    - 4 címosztály van, amiket A, B, C, D-vel jelölünk, meghatározásuk pedig a következőképpen történik:

        i. Egy IP-cím két részből épül fel, egy



        hálózati címből (ami az IP-cím első fele) és a host címből (ami az IP-cím második fele).

        ii. Azt, hogy melyik rész a network-cím, és melyik azonosítja a hostot, a **netmask** határozza meg. A hálózati címet úgy határozzuk meg, hogy az alhálózati maszkot összeéseljük az IP-címmel.

            1. Alhálózati maszk:
                a. Ha A IP-osztálybeli, akkor az első byte 255 (első 8 bit csupa 1)
                b. Ha B IP-osztálybeli, akkor az első 2 byte 255 (első 16 bit csupa 1)
                c. …

/**/

SZÁMOLÓS PÉLDÁK (lehet, hogy kicsit szájbarágósak, de az szerintem sosem baj)

1. *Határozzuk meg a 192.168.2.1 IP-cím hálózati címét.*

Az első byte binárisan 192-->1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 → vagyis C-osztálybeli, ezért az alhálózati

maszk:255.255.255.0
Bitenként összeéselve az IP címmel: **192.168.2.0, ez lesz a hálózati cím**.
(ez még segítheti az áttekintést: http://jodies.de/ipcalc?host=192.168.2.1)
11000000.10101000.00000010 .00000001 (192.168.2.1)
11111111.11111111.11111111 .00000000 (255.255.255.0)
11000000.10101000.00000010 .00000000 (Class C) (192.168.2.0/24)

2. *Változó hosszúságú alhálózati maszk esetén hány hostnak osztható ki IP cím, ha a cím 152.130.246.0/27 ?*

A változó hosszúságot a '/' jel után jelzett szám jelenti. Jelen esetben 27 bites az alhálózati maszk, vagyis mivel 32 bites az IP cím, az utolsó 5 bit (32-27) jelzi a hostok címét.
. . . . . | xxx**HHHHH** **a 'H'-val jelzettek használhatók a hostok megkülönböztetésére az adott alhálózaton belül. 2^5=32, azonban ebből 2 darab cím lejön, mivel a csupa nulla a network cím, a csupa 1 pedig a broadcast cím.**

Azaz a megoldás 32-2=30 hostnak osztható ki IP-cím
(http://jodies.de/ipcalc?host=152.130.246.0&mask1=27)

```
Address:   152.130.246.0          10011000.10000010.11110110.000 00000
Netmask:   255.255.255.224 = 27   11111111.11111111.11111111.111 00000
Wildcard:  0.0.0.31               00000000.00000000.00000000.000 11111
=>
Network:   152.130.246.0/27       10011000.10000010.11110110.000 00000 (Class B)
Broadcast: 152.130.246.31         10011000.10000010.11110110.000 11111
HostMin:   152.130.246.1          10011000.10000010.11110110.000 00001
HostMax:   152.130.246.30         10011000.10000010.11110110.000 11110
Hosts/Net: 30
```

3. *Mi a netmask a következő alhálózatban: 192.168.1.0/5?*

248.0.0.0 (első 5 bit 1-es, többi 0)

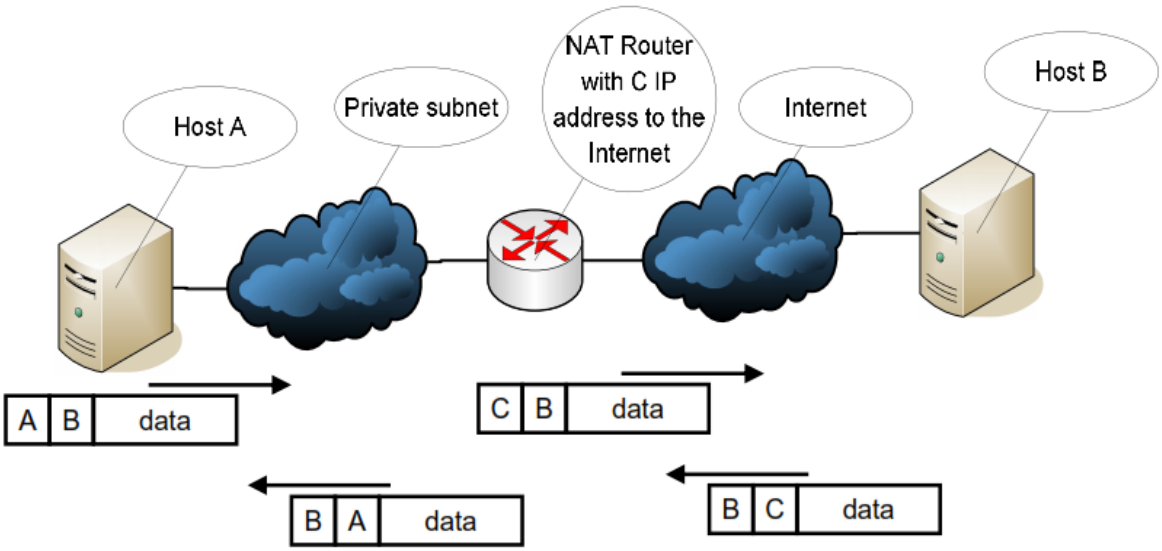
(http://jodies.de/ipcalc?host=192.168.1.0&mask1=5)
11000 000.10101000.00000001.00000000 (192.168.1.0)
11111 000.00000000.00000000.00000000 (248.0.0.0)

/**********/

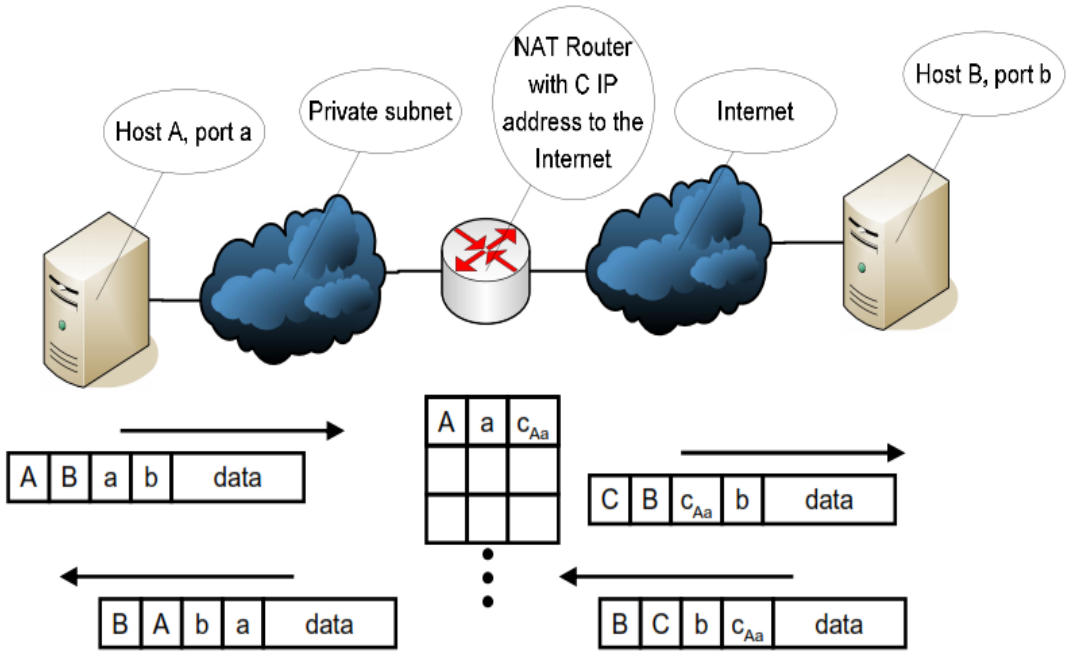
15. *Mi a loopback cím?*
    - A loopback vagy localhost címmel a saját gépünkkel tudunk kommunikálni. Bármelyik cím a 127.0.0.0/8 tartományon belül a saját számítógépünkkel kommunikál. Például loopback cím a 127.0.0.1.
16. *Mi a DHCP, és mi a feladata?*
    - Dynamic Host Configuration Protocol (dinamikus állomáskonfiguráló protokoll). Ez a protokoll azt oldja meg, hogy a TCP/IP hálózatra csatlakozó hálózati végpontok (például számítógépek) automatikusan megkapják a hálózat használatához szükséges beállításokat. Ilyen szokott lenni például az IP-cím, hálózati maszk, alapértelmezett átjáró stb.
    - A DHCP-vel dinamikusan oszthatóak ki IP-címek, tehát a hálózatról lecsatlakozó számítógépek IP-címeit megkapják a hálózatra felcsatlakozó számítógépek, ezért hatékonyabban használhatóak ki a szűkebb címtartományok. Ennek hátránya,

hogy minden kérésnél új IP-címet kap (IP-címek “újrahasznosítása”).

17. *Mi a MAC-cím, és mi a feladata?*
    - A gép fizikai címe, hexadecimális számsorozat, amellyel még a gyártás során látják el a hálózati kártyákat.
18. *Mi az ARP, és mi a feladata?*
    - Address Resolution Protocol (címfeloldási protokoll), módszer az IP-címek és fizikai címek egymáshoz rendeléséhez
    - A forrásnak tudnia kell a cél hardvercímét (MAC address), mielőtt IP-csomagokat küldhetne neki. Az ARP segítségével megtudhatjuk egy másik gép MAC címét, ha ismerjük az IP címét.
19. *Mi a RARP, és mi a feladata?*
    - Reverse ARP, a feladata az ARP-val ellentétes, vagyis ismerjük a célgépnek a MAC-címét, és az IP-címét kapjuk meg a RARP segítségével.
    - olyan táblázattal dolgozik, amelyben az van felsorolva, hogy milyen IP-cím milyen Ethernet- (fizikai) címnek felel meg
20. *Mi a NAT, és mi a feladata?*
    - Network Address Translation ([1]), feladata a hálózati címfordítás, lehetővé teszi a belső hálózatra kötött gépek közvetlen kommunikációját tetszőleges protokollokon keresztül külső gépekkel anélkül, hogy azoknak saját nyilvános IP-címmel kel lene rendelkezniük (NAT-oljuk a host IP címét; tehát a NAT több - a belső hálózatban privát IP-címmel rendelkező - eszköz csatlakozását teszi lehetővé publikus hálózatra (pl. router segítségével) ugyanazzal a publikus IPv4-címmel).
    - (kialakulásának oka: IPv4-címek szűkössége (32 bit viszonylag kevés, össz. 2^{32} címet biztosít))



21. *Mi a NAPT, és mi a feladata?*
    - NAPT (network address and port translation) = NAT+port transzláció. Portot is fordít, nem csak címet
    - j



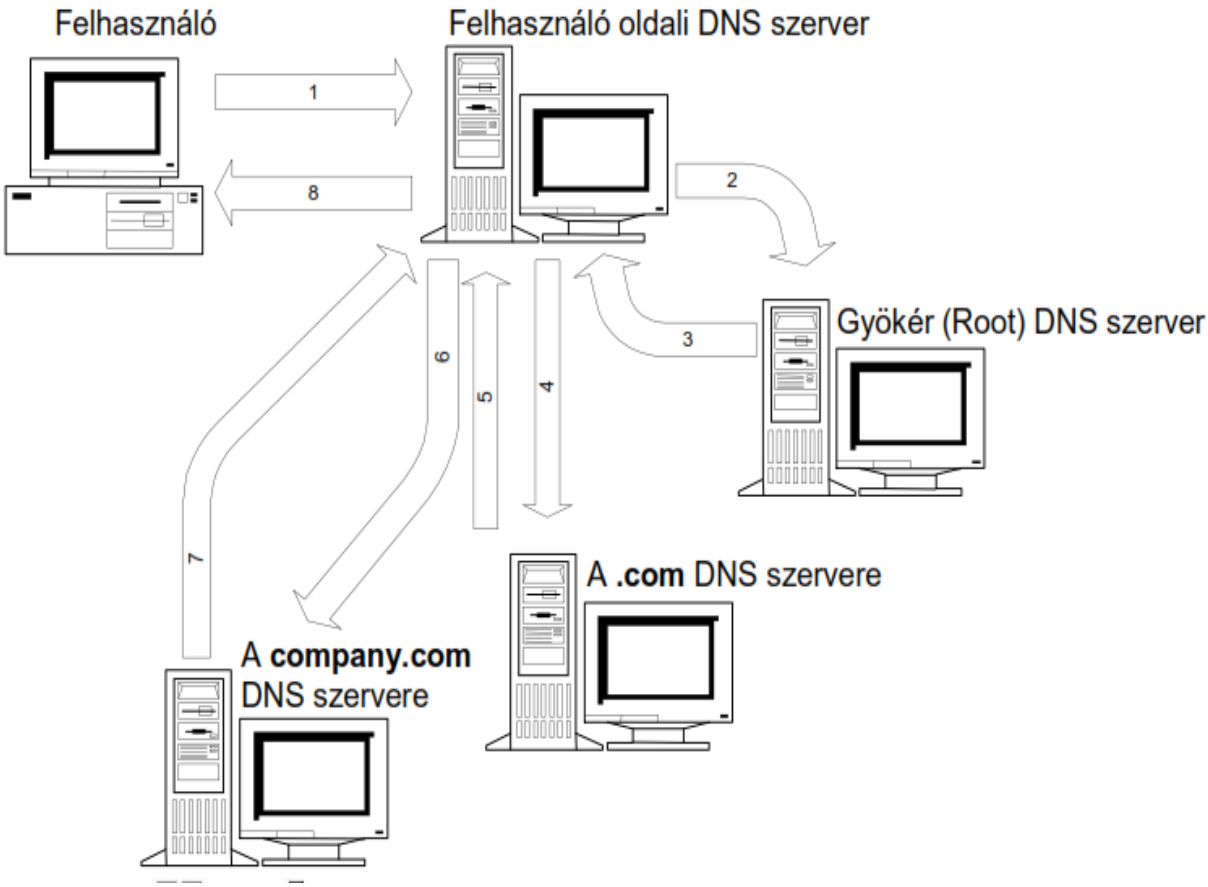
- [1] a kimenő router cserélje ki a forrás IP-címét a saját publikus IP-címére; a router nyilvántartja, ki volt az eredeti küldő, pl. forrásportszámokkal (amelyek csak *gépenként egyediek*), ezeket a routeren egyedi portszámokra cseréli. Az IP-címek és a célportszám mellett ezekkel már egyértelműen azonosítani tudja a kapcsolatokat. Kapcsolatonként nyilvántartja, hogy mit mire cserélt ki. A bejövő csomagoknál a cél IP-címen kívül a cél portszámot is vissza kell cserélnie.

22. *Mi a DNS, és mi a feladata?*
    - Domain Name System, feladata, hogy az IP-címekhez valamilyen emészthetőbb megnevezést rendeljen.
    - Legfontosabb funkciójaként az emberek számára értelmes tartományneveket a hálózati eszközök számára érthető numerikus azonosítókká „fordítja le”, „oldja fel”, melyek segítségével ezeket az eszközöket meg lehet találni, meg lehet címezni a hálózaton.
23. *Mi az ICMP feladata? Mondjon 2 példát használatára!*
    - Internet Control Message Protocol, hibajelzésre, illetve IP-szintű kontrollüzenetek továbbítására használjuk (hibák és azok típusa, hálózati diagnosztizálás).
    - az IP-t használja borítékként (ICMP csomagok csak IP hálózaton mehetnek)
    - példák: ping, traceroute
24. *Mire használjuk a ping-et, és az melyik protokoll része?*
    - A pinget végpontok tesztelésére használjuk (annak ellenőrzésére, hogy az adott távoli számítógép elérhető-e egy IP-hálózaton keresztül), az ICMP része.
    - az eszköz az ICMP protokoll ECHO parancsát küldi az ellenőrizni kívánt számítógépnek, melynek hatására az változtatás nélkül visszaküldi a kapott adatcsomagokat. A parancs elküldése után a program várja a ECHO válaszokat, majd megérkeztük után kiszámolja az oda-vissza út idejét (round-trip time) és az adatveszteséget. Ha egy csomag nem érkezik vissza az élettartamán (TTL) belül, elveszettnek minősül.
    - pl.:
      Ping alpha [152.66.246.10] with 32 bytes of data:
      Reply from 152.66.246.10: bytes=32 time=114ms TTL=250

Reply from 152.66.246.10: bytes=32 time=26ms TTL=250
Reply from 152.66.246.10: bytes=32 time=23ms TTL=250
Reply from 152.66.246.10: bytes=32 time=27ms TTL=250

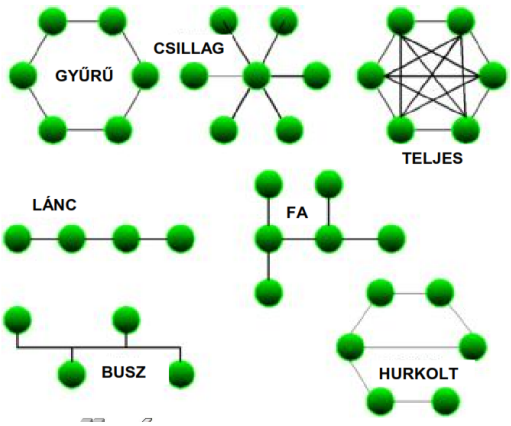
Ping statistics for 152.66.246.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 23ms, Maximum = 114ms, Average = 47ms

25. *Hogy működik a DNS, mik a domainnév feloldásának lépései?*



- [1] 1.) A hálózati gép kezdeti gyorsítótárában be vannak állítva a gyökér-névkiszolgálók ismert címei („hint”-ek). A „hint fájlt” a rendszergazda időközönként megbízható forrásból frissíti. 2.) Lekérdezi a gyökér-névkiszolgálók egyikétől, hogy mi a legfelső szintű tartomány (top-level domain (TLD); *pl.* `com`) mérvadó névkiszolgálója. 3.) Lekérdezi az imént visszakapott névkiszolgálótól, hogy mi a második szintű tartomány (*itt:* `company`) mérvadó névkiszolgálója. 4.) Az előző lépést megismétli a tartománynév minden címkéjére, míg a legutolsó lépésben megkapja a keresett név IP-címét.
- a gyakorlatban a DNS-kiszolgálók az erőforrásrekordok gyorsítótárazásával kerülik el, hogy minden lekérdezési folyamat elején a gyökér-névkiszolgálókat kelljen lekérdezni (így a gyökérkiszolgálókra csak kis terhelés jut)

26. *Mi az a 4 dolog, amit be kell állítani fix IP-címnél?*
    - IP-cím, gateway (átjáró), subnet mask, DNS
27. *Soroljon fel 6 hálózati topológiát! Melyek decentralizáltak (D)?*
    - gyűrű (D), csillag, teljes/full mesh (D), lánc (D), fa, busz, hurkolt (D)

- 

28. *Mit nevezünk logikai topológiának, mi a 3 típusa?*
    - Logikai topológiának nevezzük azt a topológiát, amiben csak 3. rétegbeli vagy afölötti eszközöket (router) használunk, tüntetünk fel./hálózatszámok, nevek, sebességek, protokollok, adminisztratív domének/. Fajtái:
        i. **Flat topologyk**
            - csak a kijáraton 3. rétegbeli entitás (router)
            - minden gép ugyanabban az IP-címtartományban („broadcast domain”)
        ii. **Location-based topology**
            - pl. emeletenként egy-egy alháló (saját IP-címtartomány)
        iii. **Functional-group based topology**
            - fizikai elhelyezkedéstől függetlenül logikai csoportok szerint (eladók, mérnökök, menedzsment, marketing) flat network
            - szolgáltatások (nyomtatás, fájl-, névszerver, autentikáció) tipikusan csoportonként
            - 3. rétegbeli eszközök kapcsolják a központi hálózathoz
29. *Mit nevezünk demarkációs pontnak?*
    - Demarkációs pont a vállalati hálózat és egy kommunális szolgáltató (telefon, hálózati szolgáltató) közötti határpont.
30. *Mit nevezünk szervernek?*
    - Szervernek nevezzük azt a számítógépet vagy szoftvert, ami lehetővé teszi más számítógépek számára a rajta tárolt **adatok, szolgáltatások**, illetve **erőforrások** elérését.
31. *Mik a homogén szerverek előnyei?*
    - egyszerűbb fenntartás
    - egyszerűbb oktatás
    - egyszerűbb pótalkatrész-raktározás (csak egy kell mindenből)
    - könnyebb javítás
32. *Mik a heterogén szerverek előnyei?*
    - nem “ragadunk be”, ha a szállítóval valami történik
    - minden feladathoz a legjobb berendezést választhatjuk
    - a gyártók közti versenyeztetés miatt olcsóbb beszerzési költség
33. *Szerverek telepítésekor mikre kell figyelnünk, miket kell biztosítanunk?*
    - fizikai védelem
    - elektromos zavarok elleni védelem

- UPS (Uninterruptible Power Supply) → védett táp
- HVAC (heating, ventilating and air conditioning) → hőmérséklet és páraszabályozás
- tűzbiztosság

34. *Szerver frissítésének 11 lépését sorolja fel:*
    - *i.* Feladatlista-készítés
        - rejtett függések feltárása és dokumentálása, kevesen olvassák el általában
    - *ii.* Kompatibilitás ellenőrzése
        - a SW-t az új OS nem támogatja: olyan verzióra frissítünk, amit még/már támogat
        - a SW-t csak az új OS támogatja: csak új OS-en lehet tesztelni → tesztgép
        - a SW-t az új OS semmiképpen nem támogatja:
            - *a.* meggyőzzük a usereket, hogy nem kell a SW
            - *b.* el kell tekinteni az OS-frissítéstől
    - *iii.* Ellenőrző tesztek elvégzése:
        - automatikus tesztek (scriptek): OK/NOK eredmények
        - manuális tesztek
        - regressziós tesztek: ugyanolyan kimenetet ad-e az új és a régi rendszer?
    - *iv.* Visszakozz-terv elkészítése: ha nem sikerülne a rendszerfrissítés, akkor vissza kell állítanunk a régi állapotba
        - *Frissítésre szánt idő = Karbantartási idő - visszakozz-idő - sikerességi teszt*
    - *v.* Karbantartási időszak:
        - *Karbantartási idő = (Frissítési idő+Teszt idő + Visszakozz-idő+ Visszakozz-teszt idő) * [2...3]*
    - *vi.* Frissítések hirdetése a felhasználók körében: LÁTVÁNYOSAN, figyelem **FE**lkEl*TŐ*Ő módon
    - *vii.* Tesztek végrehajtása
    - *viii.* Frissítések elvégzése
    - *ix.* Frissítés tesztelése
    - *x.* ...Ha nem volt sikeres, akkor Visszakozz
    - *xi.* Karbantartási eredmények hirdetése a userek közt

35. *Mit nevezünk friss installnak?*
    - Friss install a tényleges újratelepítés, ami néha lehet előnyösebb a frissítésnél. Kis luxus, klónozás, nagy luxus teljesen új rendszeren végezzük (új HW is)

36. *Mi az a redundáns tápellátás?*
    - Nem azt jelenti, hogy két táp van, hanem azt, hogy bármelyik meghibásodása esetén a rendszer működőképes marad, ha az egyik elromlik (n+1 redundancia).

37. *Mit nevezünk melegtartaléknak (hot swap)?*
    - Ezzel a technológiával felruházott rendszerekben az operációs rendszer újraindítása nélkül, menet közben lehet a meghibásodott diszket kicserélni, új diszket behelyezni. `n+1+1` redundanciát biztosít

- Legfőbb felvetődő kérdés: Mely részek legyenek ilyenek?...

38. *Sorolja fel a Desktop management szolgáltatásokat (5 db):*
    - *i.* Rendszerképkészítés (system image), automatikus géptelepítés
        - mintatelepítés a system image által
        - Wake On LAN funkció támogatása
    - *ii.* Személyre szabott SW-telepítés, alkalmazásfelügyelet, használat mérése
        - felhasználói jogosultságok meghatározása, felhasználói beállítások (háttérszín, stb...)
        - a tárolt rendszer képek általi “öngyógyítás lehetősége”
        - használat mérése: pl. hány liszenszelt felhasználó használja, mennyi a liszenszkorlát, és ha elértük, akkor tiltsuk le a hozzáférést mások számára
    - *iii.* Policy management
        - vállalati szinten meghatározott, hogy az adott felhasználók mire jogosultak, azaz felhasználó-, nem pedig gépfüggő!
    - *iv.* Távoli felügyelet
        - nem kell a felhasználónak érteni a hibakezeléshez → ”help request” opció → rendszergazda távolról tudja felügyelni és orvosolni a problémát, ha van hozzá jogosultsága
    - *v.* Teljes körű SW és HW leltár
        - SQL-ben tárolt adatok
        - lista a HW- és SW-eszközökről

39. *Rendelkezésre állási idők:*

Nagyságrend	A	Max. kiesési idő 1 év alatt
1 9-es	90 %	36,5 nap (1 hónap)
2 9-es	99 %	3,5 nap
3 9-es	99,9 %	9 óra
4 9-es	99,99 %	1 óra
5 9-es	99,999 %	5 perc
6 9-es	99,9999 %	32 mp
7 9-es	99,99999%	3 mp

- értelmezés: Pl. három kilences rendelkezésre állás = az idő 99,9%-ában jó, 100%-99,9%=0.1%-ában (0,001) rossz.
- Számítási mód: 24 óra → 1 nap, 1 év 365 nap→~8 760 óra→~31536000 másodperc,
- 24*365*X, ahol ‘X’ a kilences rendelkezésre állásoknak megfelelően alakul, azaz 1 kilences rendelkezésre állásnál 24*365*0,1 (mivel egy évben 10%-os kiesés lehet, ami 0,1) = 876 óra = 36,5 nap

Példa (diasorból):

Háromfős IT csapatával egy rendszert üzemeltet, amelyben 4 szerver és 30 desktop gép van. A vezetés rossz pénzügyi politikája miatt a rendszer nem redundáns.

– Három kilences rendelkezésre állást feltételezve, éves átlagban (egy év 31 536 000

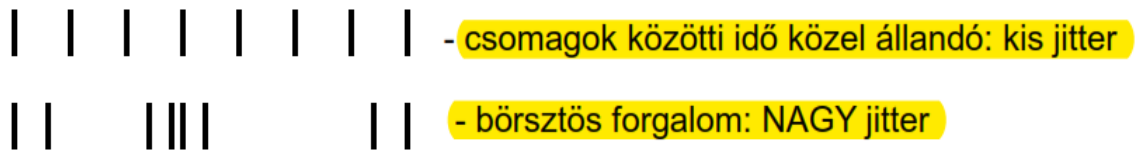
másodpercből áll) mennyi ideig megengedhető, hogy ne működjön a rendszer?
Három kilences rendelkezésre állás = az idő 99,9%-ában jó, 100% - 99,9%-ában (0,001) rossz.
31 536 000 * 0,001 = 31 536 másodperc = 525,6 perc (~ 8 és ¾ óra)

– Tervezetlen leállás nincs a rendszerben. Ez esetben mekkora a negyedéves szerver-karbantartási ablak?
Nincs tervezetlen leállás → az előző pontban kiszámolt a tervezett leállás (karbantartás) egy évben
Negyedéves: 31 536 / 4 = 7884 másodperc (~ 2 óra 11 perc)

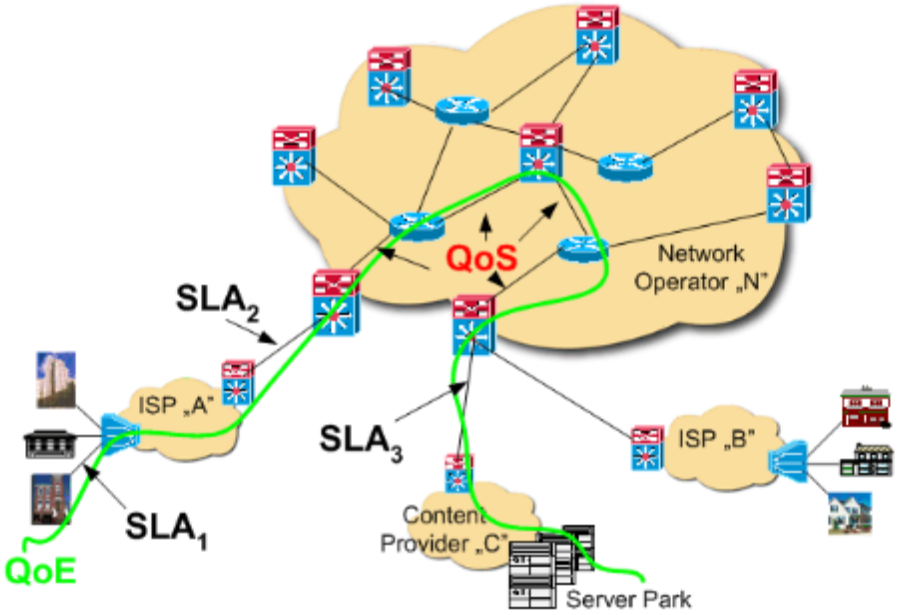
– A legkomplexebb szerveren a frissítési idő 20 perc. A régi rendszer visszaállítása 15 percet vesz igénybe. A rendszer akármilyen állapotban történő tesztelésére 10 perc kell.
• Mennyi időt tervez ennek a szervernek a karbantartására?
*Karbantartási idő = (Frissítési idő+Teszt idő + Visszakozz-idő+ Visszakozz-teszt idő) * [2...3]*
20 perc update + 10 perc teszt + 15 perc visszakozz + 10 perc teszt = 55 perc - szorozva 2..3 közötti számmal, a biztonság kedvéért.
De maximum a karbantartási ablak (2 óra 11 perc)!

• Ha látszik, hogy nem sikerül a frissítés, mikor kezdi a visszakozzt?
A karbantartási ablak vége előtt 15 perc (visszakozz) + 10 perc (teszt) = 25 perccel

40. *Mikor elégedett a felhasználó egy szolgáltatással?*
    - Ha kéréseit kiszolgálják
    - Ha a szolgáltatás minősége is kielégítő
    - Ha a felmerülő problémákat minél hamarabb orvosolni tudják
41. *QoS tipikus mércéi?*
    1. Rendelkezésre állás
    2. Throughput (áteresztőképesség)
    3. Csomagkésleltetés
    4. Csomagkésleltetés-ingadozás (*jitter*)
    5. Csomagvesztés
42. *A csomag teljes késleltetése milyen összetevőkből áll össze?*
    - feldolgozási késleltetés (processing): Csomagok feldolgozása és felkészítése az újraküldésre
    - sorbanállási késletetés (queuing delay): Csomagok sorbanállási ideje (a terhelés és az alkalmazott ütemezési eljárás határozza meg)
    - terjedési késletetés (propagation delay): A csomagok kapcsolaton való terjedési ideje
    - továbbítási késleltetés (transmission time): A teljes csomag megérkezésének ideje, az első bit beérkezésétől az utolsóig.
    - teljes csomagkésleltetés = (feldolgozási idő) + sorbanállási idő + (terjedési idő) + továbbítási idő
43. *Mi az a jitter, és minek a része?*
    - A jitter a QoS (Quality of Service) egyik paramétere, csomagkésleltetés ingadozását jelenti. Intenzív Audio-/Videoátvitelnél érdekes igazán. Nagy jitter-> börsztösebb folyamot eredményez



44. *Kik közötti megállapodás az SLA (Service Level Agreement)?*
    - Az SLA (Service Level Agreement) a hozzáférési hálózatot biztosító szolgáltató és az előfizető közötti megállapodás.
    - A szolgáltatók / hálózat-operátorok közötti megállapodás is.
45. *Mi az az SLS?*
    - Service Level Specification, az SLA műszaki melléklete, amelyben a műszaki és nem műszaki paraméterek, és azoknak a határértékeik vannak leírva.
46. *Mi az a QoE?*
    - A Quality of Experience a szubjektív felhasználói elégedettséget jelenti. Mércék típusai pedig a felhasználó elégedettségi típusaival egyezik meg (lásd korábban, “Mikor elégedett a felhasználó egy szolgáltatással?” kérdést), azaz rendelkezésre állás, kielégítő szolgáltatásminőség, hibák záros határidőn belüli orvoslása.
47. *Hol használjuk az SLA-t, QoS-t és QoE-t?*
    - Az alábbi remek ábra foglalja ezt össze



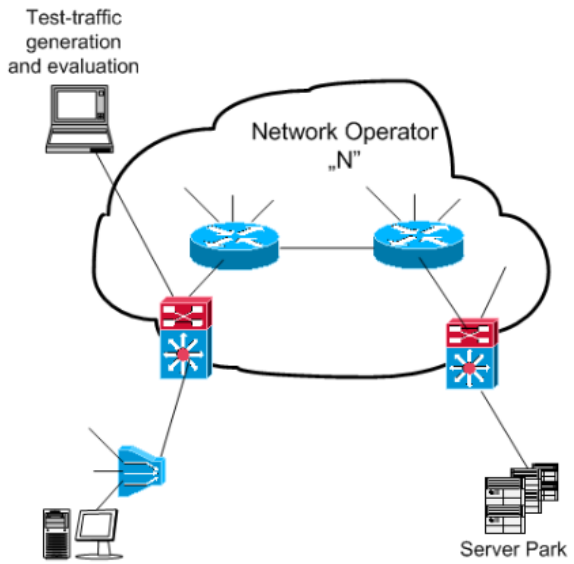
48. *Mi az a TMN, és mire használják?*
    - Telecommunications Management Network segítségével a szolgáltatók tudják menedzselni a hálózati elemeken, operációs rendszerekben, hálózattípusokon átívelő kapcsolatot és kommunikációt.
49. *Sorolja fel a TMN logikai modell elemeit:*
    1. Network Element
    2. Element Management
        - az egyes hálózati elemek, mint különálló funkcionális egységek kezelése, felügyelete
    3. Network Management
        - A hálózat, mint elkülöníthető funkcionális egység felügyeletére és vezérlésére vonatkozó feladatok
        - CM → Configuration Management
    4. Service Management
        - felhasználóval való kapcsolattartás
        - számlázási adatok
        - PM és FM → Performance- és Fault Management
    5. Business Management
        - Magas szintű tervezés
        - Pénzügyi tervek és ellenőrzés
        - Célok definiálása
        - Döntéshozás
        - Üzletszintű egyezmények (Business Level Agreements, BLAs)
50. *Minek a rövidítése az FCAPS?*
    - Fault Management, Configuration Management, Accounting, Performance Management, Security Management
    1. **Fault Management**
        - Azért felelős, hogy a szolgáltatások mindig elérhetőek legyenek
        - feladatai: hiba detektálása, jelzése az operátor felé, hibaok feltárása, hiba javítása

2. **Configuration Management**
   - A hálózat elemeinek/felépítésének és egységei változásának részleteivel foglalkozik
   - ide tartoznak: erőforrás-kihasználtság, Backup and Restore, hálózatfenntartás
3. **Accounting**
   - Felhasználói adatok kezelése
4. **Performance Management**
   - Teljesítményre jellemző mércék (pl. QoS) gyűjtése, elemzése, értékelése
5. **Security Management**
   - Feladata a nem jogosult rendszer hozzáférések minimalizálása.
   - AAA:
     a. Authentikáció → Kik férhetnek hozzá a rendszerhez?
     b. Authorizáció → Mihez férhetnek hozzá a rendszerben?
     c. Accounting → Mit csinált a rendszerben?

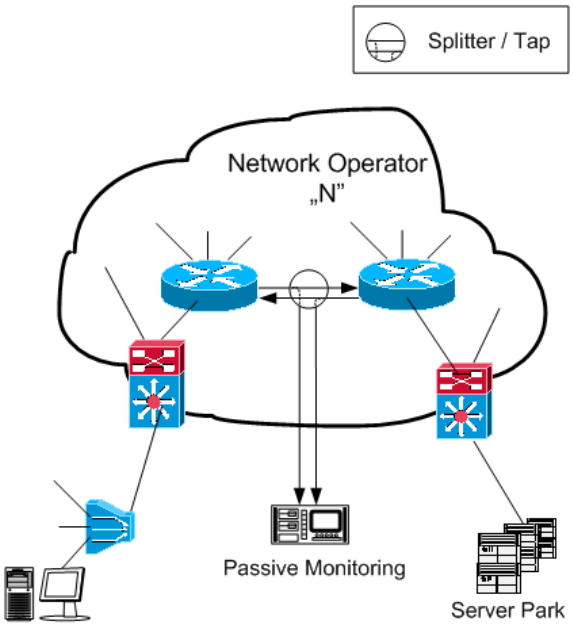
51. *Mit értünk monitorozás alatt?*
- A monitorozó eszköz csatlakoztatását a rendszerre, és segítségével adatok gyűjtését, feldolgozását, majd értékelését.

52. *Milyen monitorozási módszerek vannak?*
- Aktív és passzív monitorozás:

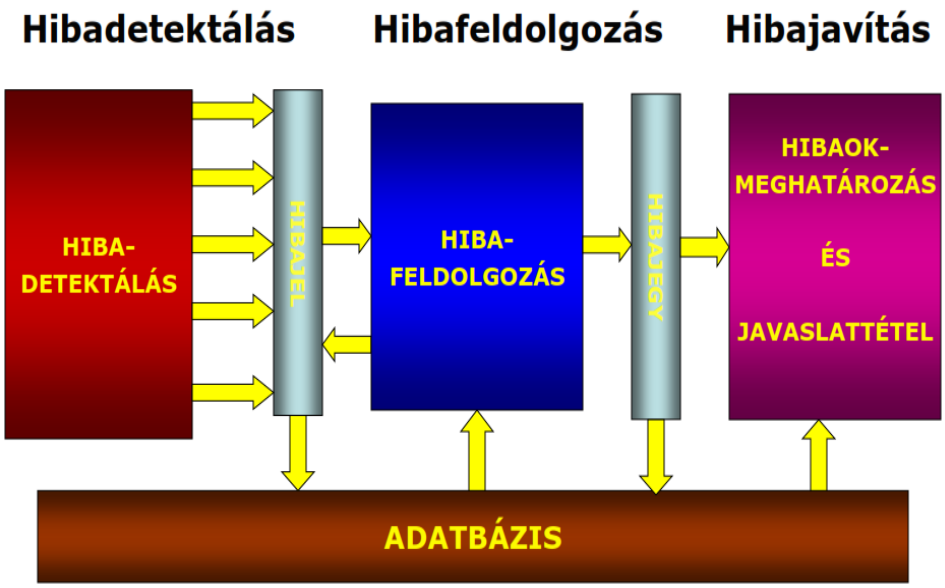
1. Aktív monitorozás:
   - Próbaforgalom beiktatása és a “hatás” vizsgálata
   - csak mintavételezés jellegű eredményekkel szolgálhat
   - a mesterséges, általunk generált forgalom torzíthatja a mérési eredményeket
   - Mérési összeállítás az alábbi ábrán látható:



2. Passzív monitorozás:
   - Hálózati forgalom külső szemlélőként való figyelése
   - hibátlan eredményt ad teljes időskálán
   - az alábbi ábra ismét megvilágosítja a sötét elmét:



53. *Milyen típusú adatokat gyűjthetünk?*
    - Topológiai
    - naplóállományok
    - nyers forgalmi szintű adatok
54. *Tranzakció azonosításának fajtái?*
    - **5-tuple**: forrás IP, cél IP, forrás port, cél port, IP protokoll
    - **3-tuple**: forrás IP, cél IP, IP protokoll
    - **N-tuple**...
55. *Sorolja fel a hibamenedzselés folyamatának lépéseit és azok eredményeit!*
    1. Hibadetektálás, eredménye: HIBAJEL (EVENT)
       - A szolgáltatást kifejezetten hátrányosan érintő események észlelése, és a hibamenedzselő rendszer minél hamarabbi értesítése.
    2. Hibafeldolgozás, eredménye: HIBAJEGY (ALARM)
       - A detektált hibajelekből hibajegy generálása a feladata
    3. Hibaok-meghatározás és hibajavítás
       - A keletkezett hibajegyekben megfogalmazott hibajelenségek okainak felderítése és a hiba javítása.



56. *Adja meg a hibajel-feldolgozás 3 típusát:*

1. **Szűrés**: A beérkezett hibajelekre különböző szűrőszabályok definiálhatók, amelyek alapján szabályozható a hibajegy-generálás
2. **Korreláció**: A beérkezett hibajelekből korrelációs szabályok alapján új, összetettebb hibajelek generálhatóak, melyek a szabályokban megfogalmazott hibajel-összefüggések alapján pontosabb információt adnak a hibajegygeneráláshoz
3. **Trendanalízis**: A beérkezett hibajelek hosszabb távú elemzése alapján, trendszabályok definiálásával olyan folyamatokból generálható hibajel, melyek feltételezhetően az adott szolgáltatást sérteni fogják, amennyiben a folyamat trendje nem változik

57. *Sorolja fel a hibaok-analízis módszereit:*

- *Alarm vektor, Szabály alapú, Eset alapú (case-based), Modell alapú, Fuzzy, Neurális hálózatok, Oksági hálózatok, Szavazás, Adatvezérelt modell*

1. *Alarm vektor*
   - A lényege az, hogy vannak különböző alarmok (mint pl.: link nem elérhető, magas jitter, zizis a kép, stb...), ezek megfeleltethetők egy tömb egyes elemeinek. A tömbbe (nevezzük vektornak), tehát a vektornak azon celláiba írunk 1-est, ahol amelyik ALARM teljesült, a többi helyre pedig 0 kerüljön. Ekkor kapunk egy vektort és a javasolt ALARM *(ez volt a diasoron, de mivel hibaok-analízist végzünk, és már ekkor megvannak az alarmok, így szerintem itt az alarm megnevezés kimenetnek nem pontos)* az lesz, amelyiknek az előre rögzített hibaok-vektorokhoz képest a legkisebb lesz a Hamming-távolsága.
   - példában mutatva:

	link nem elérhető	útvonal nem elérhető	"interface down"	eszköz nem válaszol	magas vesztés	magas jitter	...	...
Link x hibás	1	1	0	1	0	0	...	...
Link x túlterhelt	0	1	0	0	1	1	...	...
"interface misconfig"	1	1	1	1	0	0	...	...
xy hardware hiba	0	1	0	1	1	0	...	...
xy irány túlterhelt	0	0	0	0	0	1	...	...
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

t1 és t2 időpillanatok között:

1	1	1	0	0	0	...	...

a.

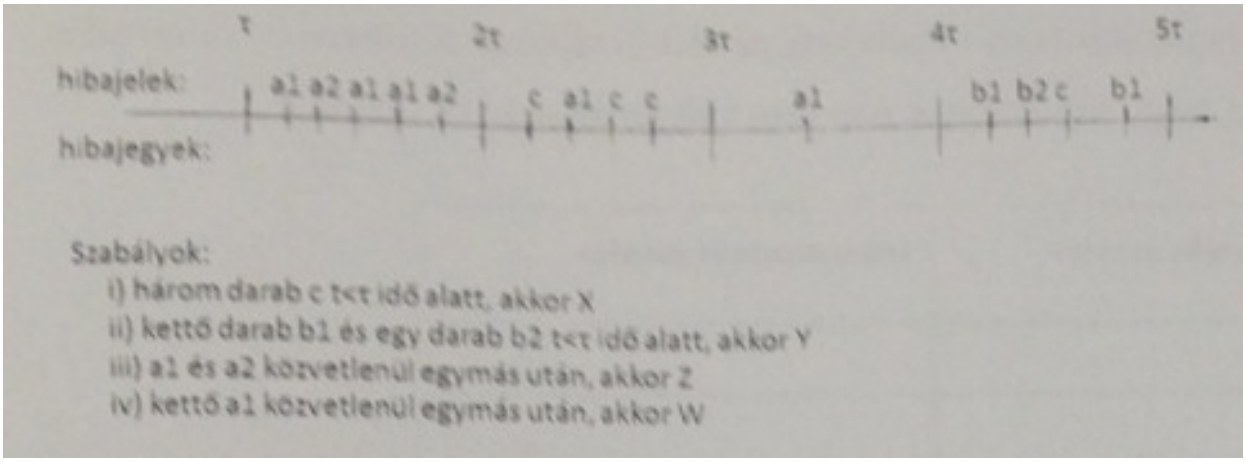
b. A t1 és t2 között beérkezett ALARMokat jelöltük a megfelelő helyen 1-gyel, vagyis t1 és t2 között nem volt elérhető a link, nem volt elérhető az útvonal, és interface down alarmokat kaptunk. Így jött létre a 111000... vektorunk, a lehetséges hibaok-vektorok közül pedig az “interface misconfig”-tól legkisebb a Hamming távolsága, így az lesz a

hibaok, aminek az elhárítása azonban a **HÁLÓZATFELÜGYELETRE HÁRUL.**

2. *Szabály alapú*
   - Az alapja egy tudásbázis, ami leírja, hogy milyen összetett alarmmal kell helyettesíteni a bejövő elemi hibajeleket.
   - Alapvetően Boole-algebrára jellemző relációként jelennek meg.
   - A 2013.05.30-ai vizsgában szerepelt egy ilyen feladat, aminek megoldását érdemes megnézni:

   a. *Egy szabály alapú esemény-korrelációval működő hibajel-feldolgozó eljárás az alábbi ábrán feltüntetett szabályok alapján működik*

   b.



Szabályok:

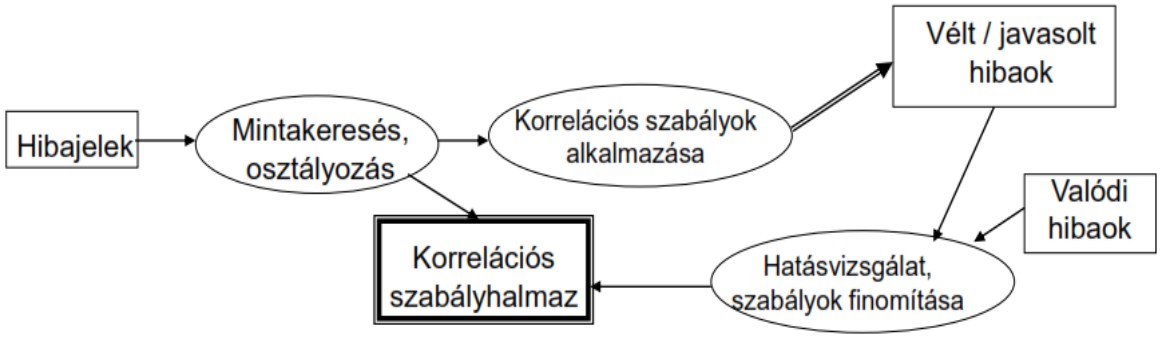
1. három darab c t<τ idő alatt, akkor X
2. kettő darab b1 és egy darab b2 t<τ idő alatt, akkor Y
3. a1 és a2 közvetlenül egymás után, akkor Z
4. kettő a1 közvetlenül egymás után, akkor W

   i. **a*, Hogyan bővítené a rendszert ahhoz, hogy eset alapú hibaok-analízis megoldássá alakuljon?***
      - *- Vélt és valódi hibaokokkal és visszacsatolóval.* LÁSD: eset alapú hibaok módszer ábránál

   ii. ***b, A visszajelzések azt mutatják, hogy csak a második és a negyedik periódusban mutatkoznak valódi rendszerhibák. Mi a legegyszerűbb, új korrelációs szabály, amit ilyenkor érdemes definiálni?***
      - *Ott generálunk hibajegyet, ahol c van, mivel c csak a második és negyedik periódusban található meg.*

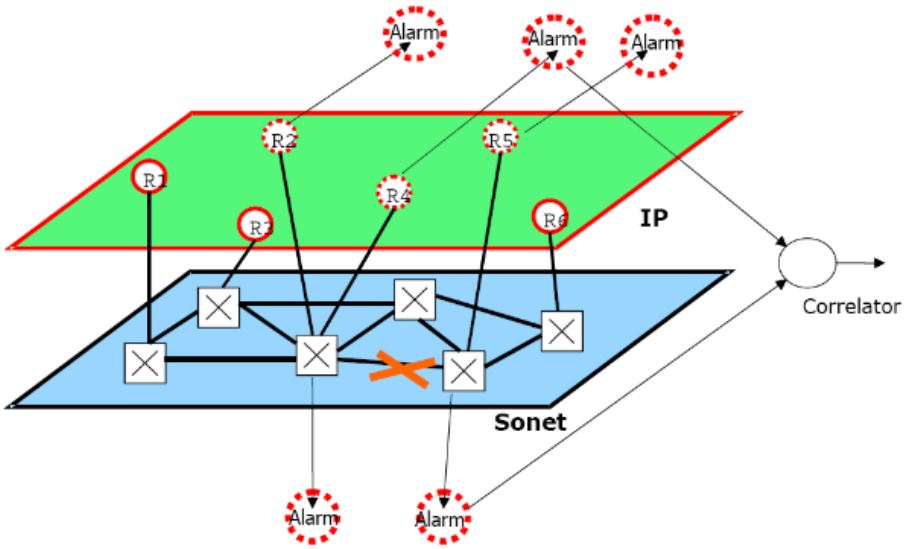
   iii. ***c, Milyen elvek mentén határozná meg a hibajeleket, eszközöket és a szabályokat egy modell alapú hibaokanalízis megoldásához?***
      - *Hierarchikus szabályok bevezetése (, rugalmas modell létrehozása a hálózati topológiából).*

3. *Eset alapú (case-based)*
   - Hasonló az előző, szabály alapú módszerhez, csak ki kell még egészíteni egy vélt, illetve valós hibaokokkal és visszacsatolóval.



-

4. *Modell alapú*
   - A korrelációs szabályok hierarchikusak, a hálózati topológiát egy rugalmas modell írja le. Bonyolult, de nagyon rugalmas megoldás



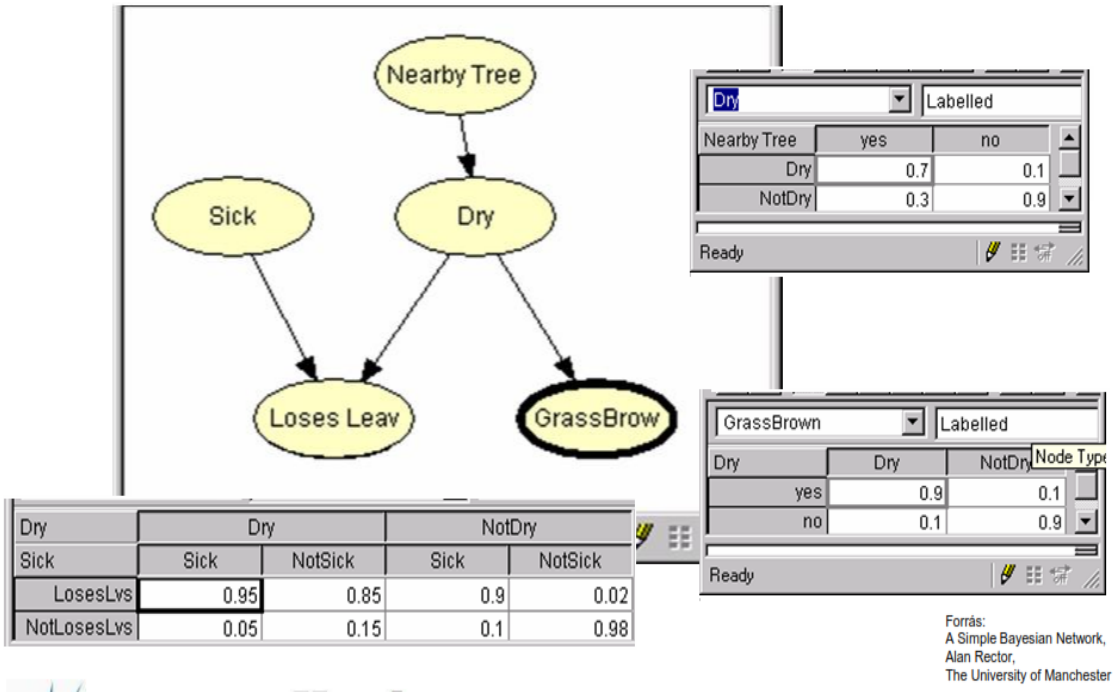
-

5. *Fuzzy*[1]
   - A lényeg, hogy az egyes hibaokokról való passzív hibakorrelációs döntés bizonytalan. A hálózatot és az alarmokat Fuzzy-halmazokkal leírva is lehet alarm-korrelációs rendszereket készíteni. Bonyolult, de gyors megoldás.
6. *Neurális hálózatok*
   - Eseménykorrelációra nehezen ráhúzható folyamat, sok állapottal, bonyolult.
7. *Oksági hálózatok*
   - ...vagy Bayes-hálózat, ami ugye valószínűségekkel operál, pontosabban a bizonytalanság leírásán van a hangsúly. Az egyes hálózati csomópontokhoz rendelt állapotoktól függően különböző valószínűségekkel jutunk el a legvalószínűbb hibaokhoz.
   - Megfigyelés-alapú.[2]

---

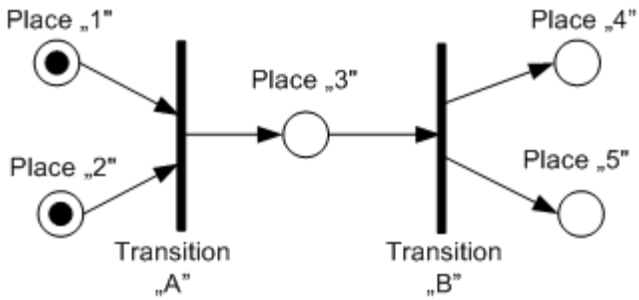
[1] Erre idéznék egy kis kérdezz-feleleket az előadásról: Varga Pali megkérdezte tőlünk, hogy: *"na akkor mi lehet a fuzzy módszer vajon?..."* A fele csoport próbált valamit nyökögni, hogy ez a... izéé.., a másik fele meg csak himbálta a fejét, nézte a nem létező felhőket az E1B mennyezeti freskóján, mire a Tanár úr csak ennyit mondott: *"Nagyon helyes! Így van! A fuzzy pont ilyen.. nem lehet biztosan megállapítani hogy pontosan az-e.. olyan fuzzys"*. Ezután meg elindított egy mexikói hullámot az előadóban ülő 30 ember segítségével :D

[2] Ismét Varga Pál Tanár urat idézném példa szemléltetésére: *"A legjobb példa rá a beérkező e-mailek közül a spamek kiválogatása. Amire gyakran nyomjuk rá, hogy spam, akkor azt egyre*

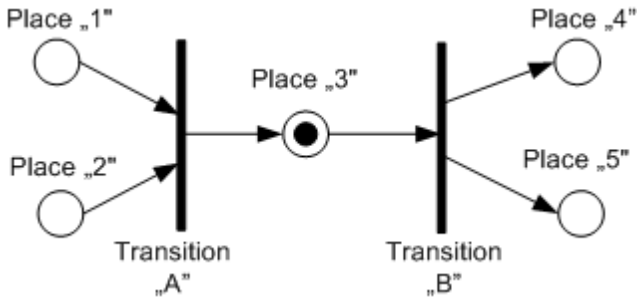


■

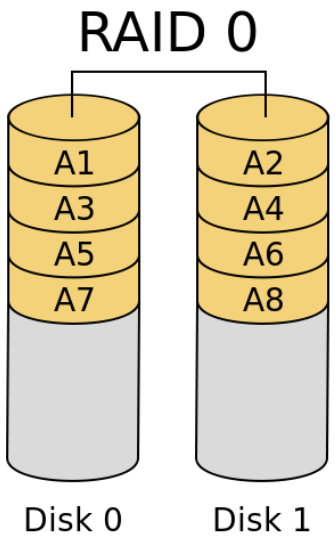
8. *Szavazás*
   - Központi döntés helyett elosztottan. Minden döntésképes csomópont megbecsüli, hogy a hozzá eljutott információk szerint milyen hibák korrelálhatóak, majd ezt egy dedikált csomópont kiértékeli.
9. *Adatvezérelt modell*
   - A hibajegy paramétereiből indulunk ki, és lehetséges hibaokok után kutatva aktív ellenőrzéseket kezdeményezünk. Csak abban az esetben hajtódik végre az ellenőrzés, ha a szükséges adatok rendelkezésre állnak.
   - Az adatvezérelt hibaok-feltárási módszer legismertebb leírása a Petri-háló. Vannak helyek, átmenetek és zsetonok. Az átmenet akkor tüzel (hajtódik végre egy vizsgálat) ha mind a 2 helyen van zseton (fekete pont), vagyis rendelkezésre állnak az adatok. (én legalábbis így értelmeztem, fix me)



biztosabban pakolja bele a spamboxba. Ilyen például a viagra reklámok....mindig rányomsz, hogy spam és berakja a spamek közé....aztán az idő múlásával már egyre kevesebbszer nyomsz rá, hogy spam" :D

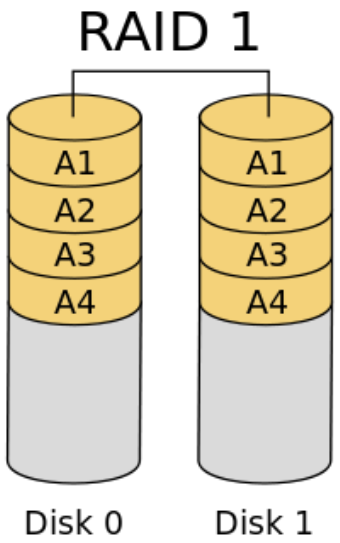

Place „1"
Place „2"
Transition „A"
Place „3"
Transition „B"
Place „4"
Place „5"

58. *Sorolja fel a HSM egyes állomásait és hogy hogyan jutunk el oda:*
    - HSM = Hierarchical Storage Management
    - --rögzítés->AKTÍV--publikálás-->REFERENCIA--archiválás->PASSZÍV--törlés->
59. *Soroljon fel 3 adattároló típust és röviden jellemezze őket!*
    (http://i.imgur.com/9LpeII2.png)
    1. **Diszk**
        - Előny: “azonnali” adatelérést biztosít, közvetlen írást/olvasást tesz lehetővé.
        - Hátrány: problémát jelent a diszkcsere, a tápellátás és hűtés, a 3-4 éves élettartam.
    2. **Optikai**
        - Előny: másodlagos tároló (automata könyvtárak), WORM (Write Once Read Many).
        - Hátrány: nem igazán tartott lépést a diszk- és szalagfejlődéssel, SOHO-eszköz (Small Office Home office).
    3. **Szalag**
        - Előny: 10-20x olcsóbb, mint a diszkes tároló, 30 éves adatmegőrzési idő.
        - Hátrány: nem azonnali elérés (lassú), sorosan olvasható/írható.
60. *RAID-típusok (DIA+EGYÉB, nem csak Wikipediás forrás)*
    - Először is a **RAID NEM VÉD A LOGIKAI HIBÁK ELLEN!!!** (Az ellen a mentés véd.) **CSAK A FIZIKAI HIBÁK ELLEN VÉD!**
    - RAID (RAID Redundant Array of Independent (Inexpensive) Disks): lemezek csíkokra (stripes) osztása. Tárolási technológia, mely segítségével az adatok elosztása vagy replikálása több fizikailag független merevlemezen, egy logikai lemez létrehozásával lehetséges. Minden RAID-szint alapjában véve vagy az adatbiztonság növelését vagy az adatátviteli sebesség növelését szolgálja.
        1. **RAID 0 - Striping (csíkokra bontás VAGY összefűzés):**
            - Nem a biztonság növelése a cél, hanem a **kapacitás növelése**, illetve a **sebességnövelés** (mivel párhuzamosan tudunk adatot írni/olvasni, ez sebességnövekedést eredményez). (Blokkszintű csíkozás, nincs paritásinformáció vagy tükrözés.)



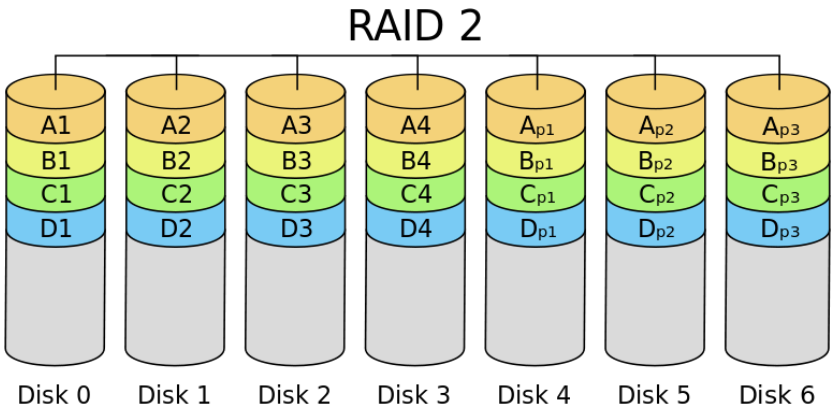
- Megvalósításához minimum 2 diszk szükséges.
- A lemezeket összefűzzük, azaz redundancia nélkül kapcsoljuk össze; nem biztosít hibatűrést, egyetlen meghajtó meghibásodása az egész tömb hibáját okozza.

2. **RAID 1 - Mirroring (tükrözés):**
   - Alapja a diszkduplikálás, vagyis az adatok tükrözése (disk mirroring), azaz az információk egyidejű tárolása a tömb minden elemén. Nagy megbízhatóság jellemzi, mivel ha meghibásodik valamelyik lemez, a másik helyettesíti (bármely meghajtó meghibásodása esetén folytatódhat a működés). Hátránya, hogy nagy (2x-es) méretnövekedést eredményez.



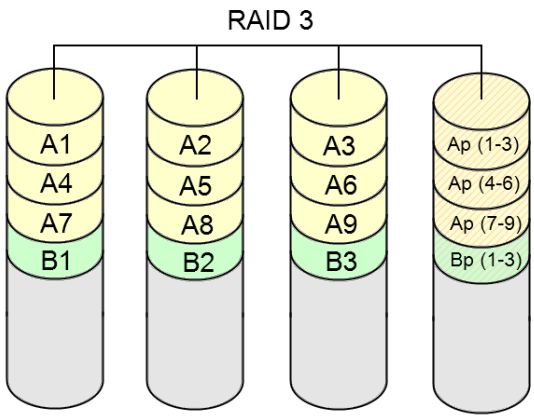
   - Minimum 2 diszk.
   - A kapott logikai lemez a tömb legkisebb elemével lesz egyenlő méretű.
   - Az adatok olvasása párhuzamosan történik a diszkekről, felgyorsítván az olvasás sebességét; az írás normál sebességgel, párhuzamosan történik a meghajtókon.
   - (önmagában nem használja a csíkokra bontás módszerét; nincs paritásinformáció)

3. **RAID 2 (hibajavító kód):**
   - Egyes meghajtók hibajavító kód (**ECC** - Error Correcting Code) tárolására vannak fenntartva, így a tömb képes a hiba detektálására, javítására. Ez megnövekedett adatmennyiséget eredményez.
   - Megvalósításához **minimum 3 diszk** szükséges.

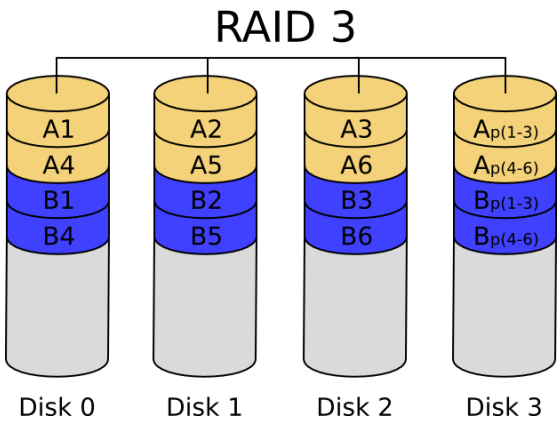


- használja a csíkokra bontás módszerét: a meghajtók egy-egy csíkjában a különböző lemezeken azonos pozícióban elhelyezkedő csíkokból képzett hibajavító kódot tárolnak (inkább bitszintű, mint blokkszintű csíkozás)
- Ma már nem használják, mert ma már a meghajtókon belül képeznek hibajavító kódokat.

4. **RAID 3 (paritásdiszk):**
   - hasonlít a RAID 2-re, viszont nem a teljes hibajavító kód, hanem csak egy lemeznyi paritásinformáció tárolódik
   - **1 paritásdiszk** van fenntartva, amely a többi diszkből XOR-művelet segítségével előállítható. Ha kiesik 1 diszk, akkor nincs baj, így n+1-redundáns!
   - Hibadetektálásra nem jó! Itt feltesszük, hogy a hibát valamilyen módon (például többszöri sikertelen olvasás hatására) észleljük, majd előállítjuk a meghibásodott lemez adatait (a többiből).
   - Paritásdiszk korlátozza a teljesítményt (a paritáscsíkot minden egyes íráskor módosítani kell, amihez szükséges a korábbi tartalom kiolvasása).
   - Csak **single-user** módban használható, egyszerre több kérés párhuzamos kiszolgálását nem támogatja.
   - **Nagy fájlok** (pl. videófájlok) írására és olvasására alkalmas, amelyek a legnagyobb átviteli sebességet igénylik, hosszú szekvenciális olvasás és írás formájában
   - kicsi a szektorok mérete!
   - Kisméretű csíkok - bájtszintű csíkozás, mindig egész stripe-művelet
   - Leggyakoribb előfordulásai: 2+1, 5+1, 8+1, 14+1.
   - Minimum 3 diszk.

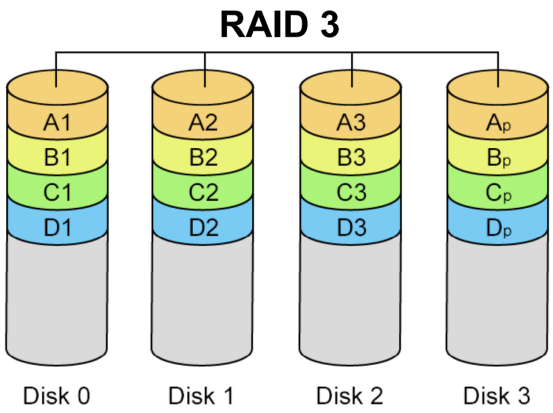


   - magyar Wikipédia:



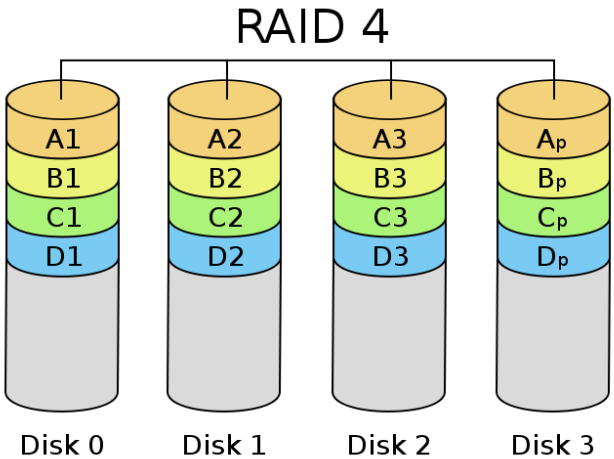
   - angol Wikipédia:

magyarázat: 6 bájtos blokkok, 2 paritásbájt, az ábrán két blokknyi adat látszik, különböző színekkel



- ~~diából:~~
- (A gyakorlatban ritkán használják.)

5. **RAID 4 (paritásdiszk, de nagyméretű csíkok):**
   - Hasonló a RAID 3-hoz, de itt nagyméretű csíkokat definiálnak, így egy rekord egy meghajtón helyezkedik el, lehetővé téve egyszerre több (különböző meghajtókon elhelyezkedő) rekord **párhuzamos írását, illetve olvasását** (közvetlenül bármelyik diszkhez hozzáférhetünk) - tehát támogatja a **multi-user** módot.
   - A **paritásdiszk nagyon korlátozza** a teljesítményt (sok a frissítés a sok párhuzamos írás/olvasás miatt: a paritásmeghajtó adott csíkját minden egyes íráskor frissíteni kell (plusz egy olvasás és írás). Ezenkívül valamely meghajtó kiesése esetén a rendszer olvasási teljesítménye is lecsökken, a paritás-meghajtó jelentette szűk keresztmetszet miatt.)
   - n+1-redundáns ez is
   - Minimum 3 diszk.

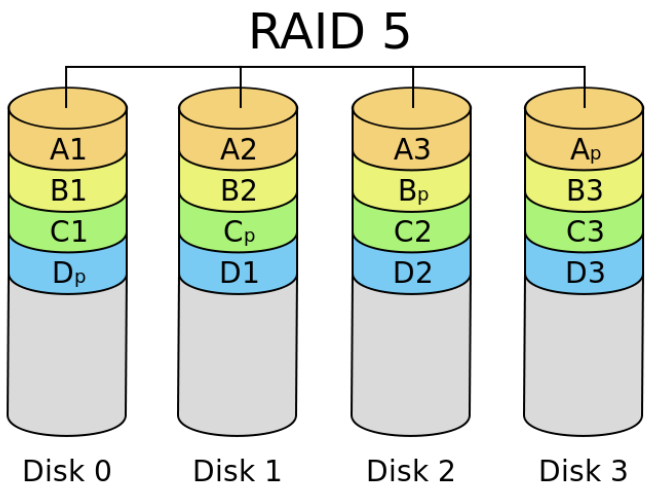


   - magyarázat: RAID 4, dedikált paritásdiszkkel, mindegyik szín a blokkok egy csoportját reprezentálja a megfelelő/vonatkozó paritásblokkban (egy csíkban). Példa: az *A1* blokkra vonatkozó olvasási kérést a *disk 0* szolgálná ki. Egy *B1*-re vonatkozó párhuzamos olvasási kérésnek várakoznia kellene, de egy *B2*-re vonatkozó olvasási kérést egyidejűleg ki tudna szolgálni a *disk 1*.
   - Nem használják a gyakorlatban.

6. **RAID 5 (elosztott paritás):**
   - blokkszintű csíkozást alkalmaz, **elosztott paritással**: a paritásinformációt nem

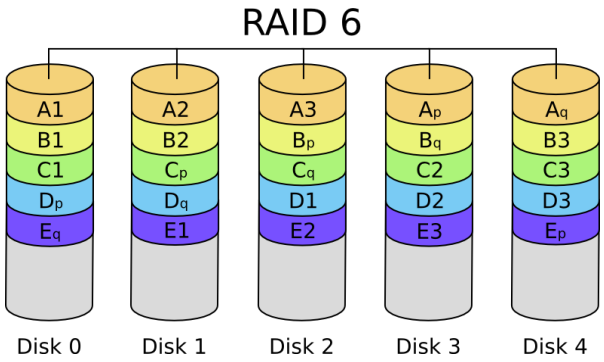
egy kitüntetett meghajtón, hanem „körbeforgó paritás” (rotating parity) használatával, egyenletesen az összes meghajtón elosztva tárolja, **kiküszöbölvén a paritás-meghajtó jelentette szűk keresztmetszetet.**

- Mind az írási, mind az olvasási műveletek párhuzamosan végezhetőek (közvetlenül elérhetők a diszkek).
- a csíkméret **változtatható** (kisméretű csíkok esetén a RAID 3-hoz hasonló működést, míg nagyméretű csíkok alkalmazása esetén a RAID 4-hez hasonló működést kapunk)
- **n+1**-redundáns: egy meghajtó meghibásodása esetén az adatok sértetlenül visszaolvashatóak, a hibás meghajtó adatait a vezérlő a többi meghajtóról ki tudja számolni. (Két meghajtó meghibásodása esetén azonban az adatok elvesznek.)
- **kapacitás** kiszámítása = *(legkisebb kapacitású diszken tárolható adatmennyiség) * (összes diszk - 1)*
  (Pl. 4 db egyenként 1 TB -os HDD *RAID 5*-be fűzésének eredményeként egy (*1*(4-1)*=) *3 TB* kapacitású logikai meghajtót látunk.)
- **olvasási sebesség** kiszámítása = *(leglassabb diszk olvasási sebessége) * (összes diszk - 1)*
- minimum 3 diszk szükséges a megvalósításához

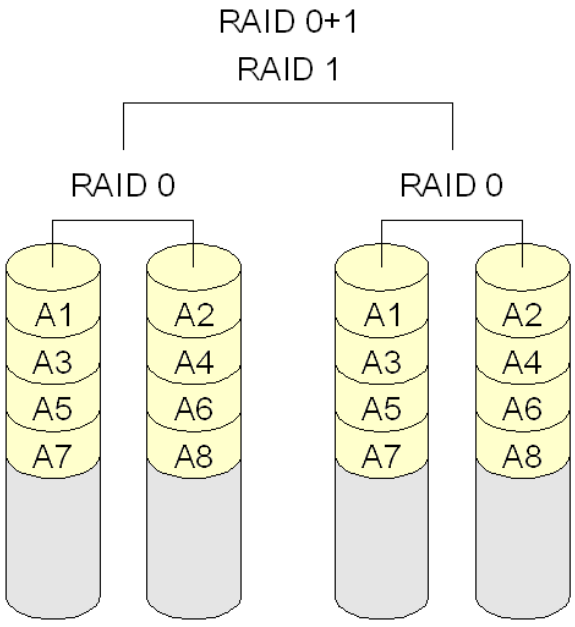


- magyarázat: RAID 5 elosztott paritással, mindegyik szín a blokkok egy csoportját mutatja a vonatkozó/megfelelő paritásblokkban (egy csíkban)

7. **RAID 6 (kettős paritás):**
   - A RAID 5 kibővítése, egy **további paritásblokk** hozzáadásával (**kétszeresen elosztott paritás**): itt nemcsak soronként (XOR – P), hanem oszloponként (Reed-Solomon kód – Q) is kiszámítják a paritást. Ezzel **n+2**-redundanciát biztosít, tehát a kétszeres meghajtómeghibásodás is kiküszöbölhetővé válik.
   - De **lassú**! Olvasási műveleteknél nem, de az írásnál a paritáskalkulációk nagy overheadet jelentenek.
   - A paritáscsíkokat (blokkszintű csíkozás) itt is az egyes meghajtók között, **egyenletesen elosztva** tárolják, de ezek természetesen kétszer annyi helyet foglalnak el, mint a RAID 5 esetében.
   - **Minimum 4 diszk.**

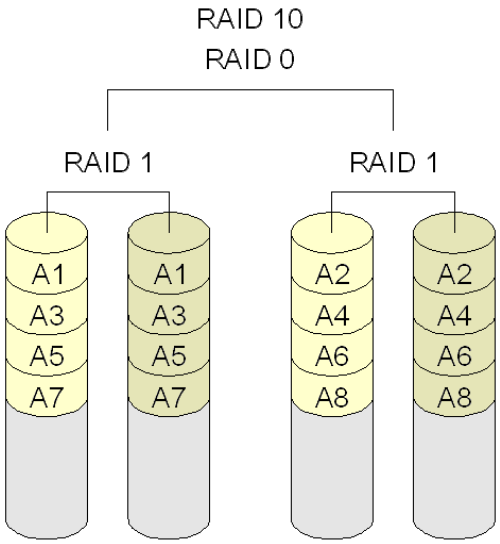


8. **RAID 01 (vagy RAID 0+1):**
   - Ez egy olyan hibrid megoldás, amelyben a RAID 0 által hordozott sebességet (párhuzamos írás/olvasás) a RAID 1-et jellemző biztonsággal (tükrözés) ötvözhetjük.
   - Minimum 4 diszk szükséges a használatához: 1-1-et RAID 0-val összefűzve, majd páronként RAID 1-gyel tükrözve építhetjük fel a tömbünket (csíkozott halmaz egy tükrözött halmazban, vagy csíkok tükre). Hátrány: így a teljes kinyerhető kapacitásnak mindössze a felét tudjuk használni.
   - használható **kapacitás** = *(legkisebb diszk kapacitása) * (összes diszk/2)*
   - Mivel a tükrözés (RAID 1) a két összefűzött (RAID 0) tömbre épül, ezért **egy lemez meghibásodása** esetén az egyik összefűzött tömb mindenképp kiesik, így a **tükrözés** is **megszűnik** - tehát az egész stripe-ot érinteni fogja a hiba (az egyik egész blokk elszáll). Így **helyreállításkor** az egész stripe-ot helyre kell állítani (tehát pl. nem csak egy diszket érint a hiba, hanem kettőt is).



9. **RAID 10 (vagy RAID 1+0):**
   - Itt is minimum 4 diszk szükséges
   - a RAID 01-hez hasonló, de itt a lemezeket először tükrözzük RAID 1 megoldással, majd a kapott tömböket fűzzük össze RAID 0-val (tükrözött halmaz egy csíkozott halmazban, vagy tükrök csíkja). Mivel az alsó szinten RAID 1 van, így egy diszk meghibásodása esetén nem esik ki az egész diszkblokk. Tehát **hiba** esetén csak a blokk felén nincs tükrözés, és csak a

rossz diszket kell **helyreállítani**.



○

61. *Jellemezze a DAS tároló rendszert:*
    - DAS = Directly Attached Storage.
    - A tároló közvetlenül csatlakozik a szerverhez
    - kis rendszereknél használják
    - Blokkszintű hozzáférést biztosít

62. *Milyen típusai vannak a DAS-nak?*

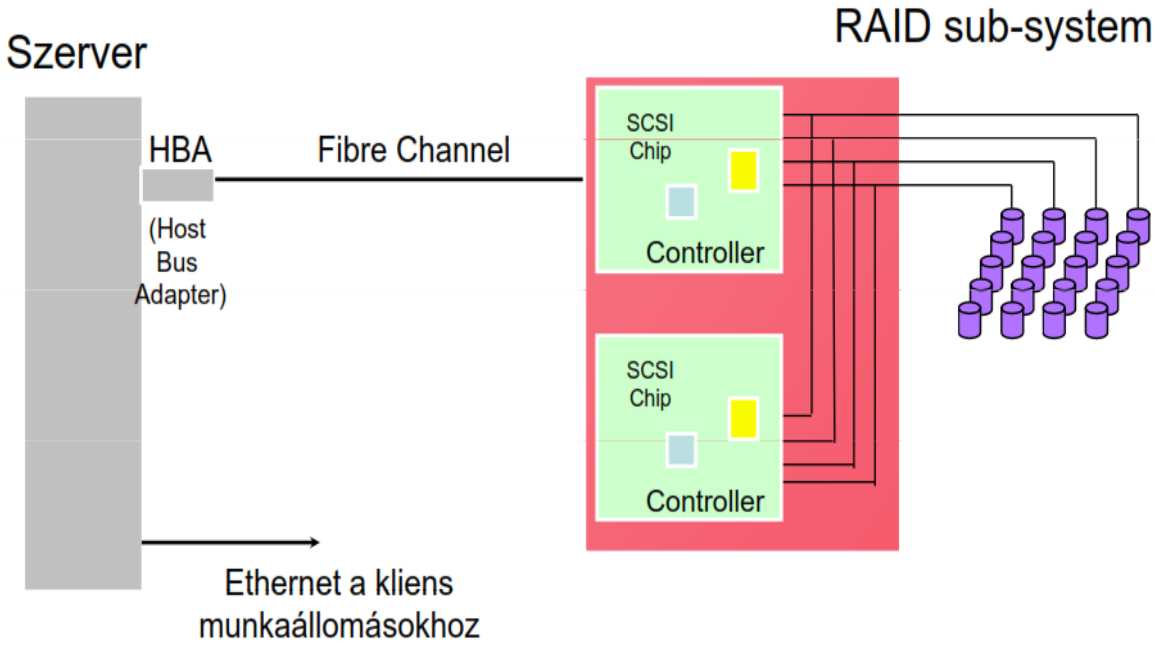
1. **Internal DAS (Belső)**
    - A **tároló közvetlenül a szerverhez csatlakozik**, belső soros vagy párhuzamos buszon keresztül.
    - a távolság korlátozott
    - a buszhoz csak korlátozott darabszámú eszköz csatlakoztatható (*(P)ATA*- (40 v. 80 eres kábel, párhuzamos elérés; max. 133 MB/s) vagy *SATA*-csatlakozóval (utóbbi: pont-pont összeköttetést teremt a SATA host adapter és a SATA eszköz között, soros elérés, 4 eres kábel; min. 150 MB/s))
    - nagy helyet foglal a szerveren belül → nehezebb a karbantartása
2. **External DAS (Külső)**
    - A szerver közvetlenül kapcsolódik **egy külső tárhoz**!
    - Nagyobb távolság
    - általában nem (annyira) korlátozott a csatlakoztatható eszközök száma.
    - SCSI- (Small Computer System Interface) vagy FC-csatlakozás, az összes tároló egy buszon osztozik,
    - LUN = Logical Unit Number, azonban nem számot azonosít, hanem az egy SCSI-csoporton belüli egységeket azonosítja
    - SAS = Serial Attached SCSI, gyorsabb mint a sima SCSI, full duplex átvitelre is képes

63. *Sorolja fel pár előnyét és hátrányát a DAS-nak:*
    - **Előnyök**:
        1. Jobb, mintha a kliens tárolná az adatokat
        2. egyszerű
        3. kis költségű

4. korlátozott redundanciát tud nyújtani

- **Hátrányok**:
  1. korlátozott darabszámú eszköz csatlakoztatható
  2. nehézkes a menedzselése
  3. költséges a backup
  4. elpocsékolt tárolási terület
  5. nehézkes az adatmegosztás
  6. nem (jól) skálázható
  7. kis teljesítmény, korlátozott sávszélesség

64. *Mi a SAN, hogy épül fel és ahol lehet, milyen típusú elérés megvalósítható?*
    - A SAN (Storage Area Network) egy központosított, nagy teljesítményű adattárolásra dedikált hálózat.
    - Részei: kliensek, LAN → fájlszintű elérés, szerverek, SAN → blokkszintű elérés, Storage Pool
    - Átviteli technológia: FC (Fiber Channel), nagyszámú eszköz nagy távolságban csatlakoztatható
65. *A SAN alapkonfigurációjában hány vezérlős a RAID, hány busz van, és az egyes buszokhoz hány lemez csatlakoztatható?*
    - 2 vezérlős RAID, ezek közül csak az egyik dolgozza fel az I/O-kéréseket, 4 SCSI busza van, mindegyiken 5 lemezzel



66. *Jellemezze a NAS tároló rendszert, milyen NAS-protokollok vannak?*
    - Network-Attached Storage, hálózatra csatlakoztatott adattároló eszköz, ami támogatja az adatmegosztást kliens-szerver között.
    - NAS protokollok:
      1. NFS (Network File System): UNIX alatt
      2. CIFS (Common Internet File System): OS-független, TCP felett
      3. FTP (File Transfer Protocol)
67. *Milyen diszkrendszerbeli másolás fajták vannak?*
    1. **Volume Copy**
       - Valódi kötet jön létre, firmware-eszközökkel megvalósított másolási technológia,

alkalmas backup-célra!

2. **Flash Copy**
   - Nem jön létre valódi másolat, hanem ha egy blokkot módosítunk, akkor azt nem írjuk felül, hanem máshová tesszük, és ezt jelöljük a Flash Copy táblában, ami által tetszőleges időpontbeli állapot helyreállítható (amelyiket természetesen rögzítettük (snapshotoltuk) a Flash Copyban)
   - Nem alkalmas backup-célra, mivel nem keletkezik valódi másolt állomány!
   - COW:Copy On Write
   - Flash Copy vizsgafeladat-megoldás levlistáról:

   a. kiindulás:

Blokktábla				Flashcopy tábla		
Időpontok	**T1**	**T2**	**T3**	**F1**	**F2**	**F3**
	B0		B8	B0		B8
	B1	B1		B1	B1	
	B2		B9	B2		B9
	B3	B3	B3	B3	B3	B3
	B4	B4	B4	B4	B4	B4
	B5	B5	B5	B5	B5	B5
	B6	B6	B6	B6	B6	B6
	B7	B7	B7	B7	B7	B7
Összes blokkszám	8			8	8	8
Delta (flashcopy inkrementum)	0			Látszólagos Volume A		

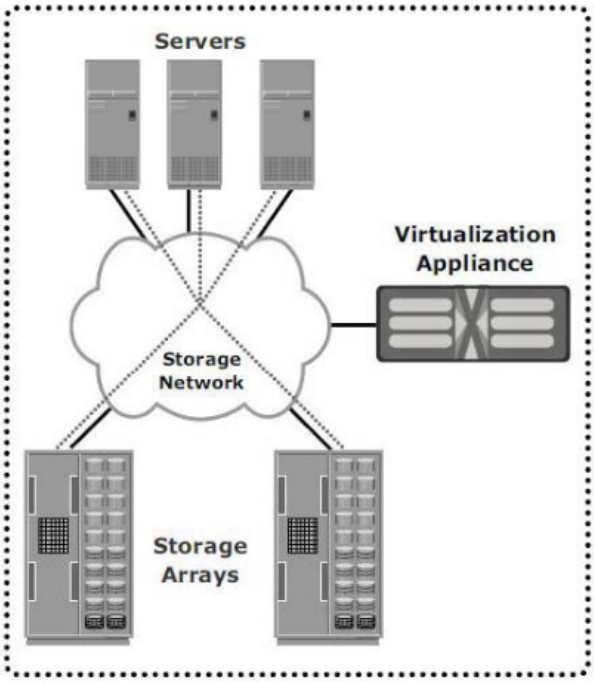
   b.

   c. *(Hajdu Ákos) A lényeg: **T2** időpillanatban a **B0** és **B2** blokkokat akarod írni, **T3** időpillanatban pedig **B1**-et. Annak érdekében, hogy a fájl tetszőleges időpillanatbeli állapota visszaállítható legyen, nem írsz felül semmit, hanem a **B0**, **B2**, **B1** blokkok helyett a **B8**, **B9**, **B10** (eddig üres) blokkokba írod a tartalmakat. A flashcopy táblába pedig feljegyzed az **F2 oszlopba, hogy a B0** blokk tartalma a **T2** időpillanatban valójában a **B8** blokkban van. (Hasonlóan a **B9**-et és **B10**-et is feljegyzed a táblába). Ezek után, ha valakit a fájl **T1,T2,T3 időpillanatbeli állapota érdekli, azt össze tudja halászni a flashcopy tábla F1,F2,F3** oszlopai segítségével.*

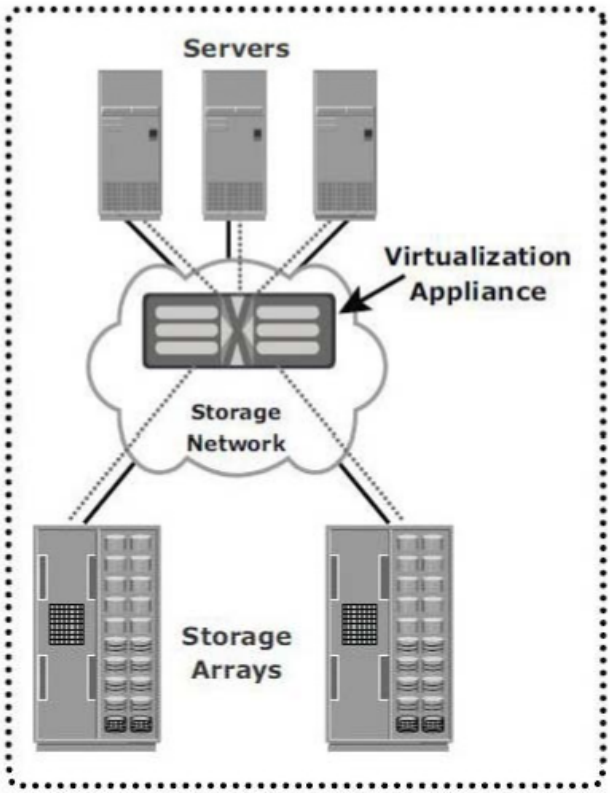
   d. *a megoldás:*

Blokktábla				Flashcopy tábla		
Időpontok	**T1**	**T2**	**T3**	**F1**	**F2**	**F3**
Írás t2	B0	**B0>B8**	B8	B0	**B8**	B8
Írás t3	B1	B1	**B1>B10**	B1	B1	**B10**
Írás t2	B2	**B2>B9**	B9	B2	**B9**	B9
	B3	B3	B3	B3	B3	B3
	B4	B4	B4	B4	B4	B4
	B5	B5	B5	B5	B5	B5
	B6	B6	B6	B6	B6	B6
	B7	B7	B7	B7	B7	B7
Összes blokkszám	8	10	11	8	8	8
Delta (flashcopy inkrementum)	0	2	3	Látszólagos Volume A	B	C

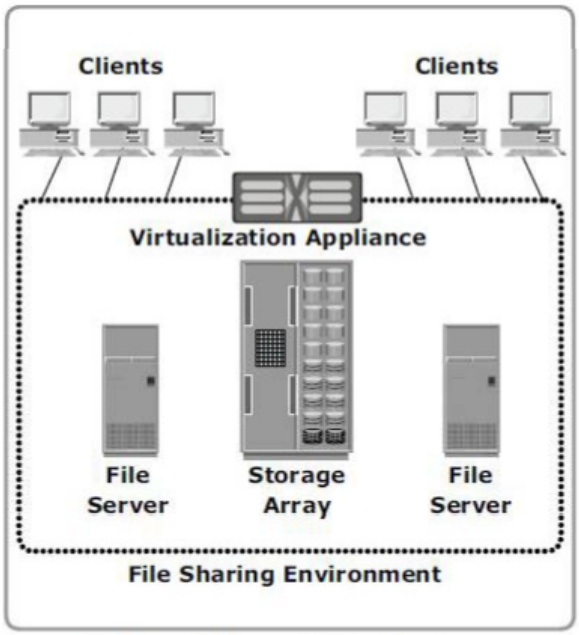
68. *Sorolja fel a NAS előnyeit:*
    - skálázható, bővíthető, de a LAN miatt a sávszélesség korlátozott
    - könnyen telepíthető, üzemeltethető eszköz
69. *Sorolja fel a SAN előnyeit:*
    - Skálázható, bővíthető, nagy adatátviteli sebesség
70. *Mit jelent a tárterület-virtualizáció?*
    - A virtualizáció olyan technológia, amely lehetővé teszi, hogy bizonyos erőforrások más erőforrásoknak tűnjenek, lehetőleg kedvezőbb tulajdonságokkal. A virtualizáció jellemzően elfedi a háttérrendszer komplexitását, és új, hatékonyabb funkciókat biztosít a háttérrendszer szolgáltatására építve.
71. *Virtualizációs motor működése:*
    - elfedi a diszkek különbözőségét
    - Szerver: LUN=1, LBA=32 LBA Logical Block Address
    - VM: táblázatból, ez megfelel a fizikai LUN=4, LBA=0 címnek
    - Elkéri az adatot a fizikai diszktől
    - A megkapott adatot úgy továbbítja a szervernek, mintha az a LUN=1, LBA=32 címről érkezett volna.
72. *Milyen virtualizációs konfigurációk vannak?*
    1. **out-of-band**
        - A vezérlés és az adatút elválik, a szerveren külön kell SW, mert először ez elkéri a VM-től az adat fizikai helyét/címét, majd közvetlenül eléri az adatot.



2. **in-band**
   - A virtualizációs motor az alagútban van, lassabb, mivel egy VM-en megy át az adat



73. *Mi a legnagyobb előnye a fájlszintű virtualizációnak a virtualizáció nélküli megoldással szemben?*
    - A fájlszintű virtualizációnál a kliensnek/hostnak nem kell tudnia, hogy fizikailag melyik fájlszerveren található a fájl, amelyet el szeretne érni, ez egyszerűbb fájlmozgatást, terhelésmegosztást és bővítést tesz lehetővé; valamint a fájlmozgatás nem érinti a klienst (cloud computing).

- 

74. *Mik a tárolóerőforrás-menedzsment (Storage Resource Management) főbb lépései (4 db)?*
    1. **Azonosítás (tárolóeszköz-azonosítás)**
        - eszközleltár
        - lefoglalt területek kigyűjtése
        - allokált, de nem használt területek
        - mi az aktuális kapacitás-kihasználás?
        - van-e kritikus fájlrendszer?
    2. **Értékelés (adatosztályozás)**
        - fájl- és könyvtárszintű analízis
        - a feleslegesen foglalt tárterületek azonosítása
        - árva, régi, nem használt, duplikált állományok azonosítása
    3. **Vezérlés (adattárolás-kontroll)**
        - központi riasztási rendszer
        - kvótakezelés
        - automatikus válaszakciók indítása
        - a menedzsment-funkciók automatizálhatóak
    4. **Előrejelzés (trendek alapján)**
        - leggyorsabban növekvő felhasználók, fájlrendszerek, adatbázistáblák
        - trendek azonosítása és előrejelzés
        - kockázatelemzés
75. *Mi a mentés/archiválás célja?*
    - a helyreállíthatóság biztosítása, adatvesztések elkerülése (minimalizálása) másolati adatpéldányok készítésével.
76. *Mi az archiválás célja?*
    - Referencia-időpontnak megfelelő adattartalom megőrzése.
77. *Mi a helyreállíthatóság szükségességének 3 fő oka?*
    - Archiválás, Véletlen adattörlés, Diszkmeghibásodás
78. *Sorolja fel a 4 mentési módszert:*

1. **Teljes mentés**
   - Minden nap a teljes diszktartalmat mentjük. Rossz a szalagkihasználtság, lassú, a változatlan adatok is sokszor mentésre kerülnek, viszont egy szalagról helyreállítható.
2. **Inkrementális mentés**
   - A ciklus első napján teljes mentés, utána minden nap az **előző mentés óta** történt **változásokat** mentjük le. Ez kis adatmennyiséget eredményez, de a visszaállítási idő hosszú, és rossz a szalagkihasználtság.
3. **Differenciális mentés**
   - A ciklus első napján teljes mentés, utána minden nap csak az **előző <u>teljes</u> mentés** óta történt változtatásokat menjtük le - ez nagyobb, egyre növekvő napi adatmennyiséget eredményez.
   - DE: rövidebb a visszaállítási idő (max. 2 szalag); több szalag
4. **Progresszív mentés**
   - Teljes mentés csak egyszer (a ciklus első napján), utána csak inkrementális mentés, DE mellette az adott napi fájlstruktúrát is elmentjük! Ez kicsivel (!) több mentést eredményez, mint az inkrementális mentésnél.
   - Így helyreállításkor visszakereshető, hogy egy fájlnak melyik az aktuális állapota.
   - Jelentős időnyereség többször módosított, ill. törölt fájlok helyreállításakor.

Inkrementális / Differenciális mentés problémája

Day 1	Day 2	Day 3	Day 4	Day 5
File A	File A renamed to File F	File F	File F	File F deleted
File B	File B deleted			
File C	File C renamed to File G	File G	File G	File G
File D	File D moved to new location	File D (new location)	File D deleted	
File E	File E	File E	File E	File E

Files from Day 1 FULL backup		Files from Day 3 / INCREMENTAL / DIFFERENTIAL backup		Hard Drive after a restore to Day 3
File A		File F		File A – **wrong** File F
File B	+		=	File B – **wrong**
File C		File G		File C – **wrong** File G
File D		File D (new location)		File D – **wrong** File D (new location)
File E				File E

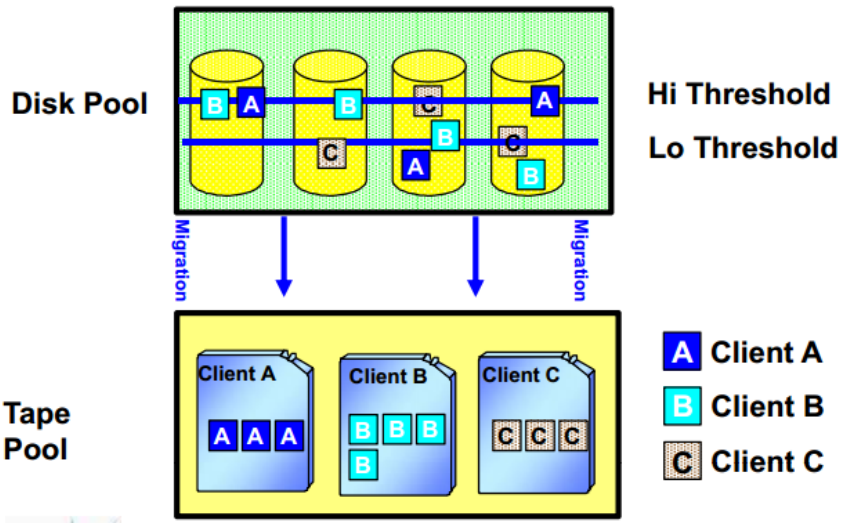
Progresszív mentés előnye

Day 1	Day 2	Day 3	Day 4	Day 5
File A	File A renamed to File F	File F	File F	File F deleted
File B	File B deleted			
File C	File C renamed to File G	File G	File G	File G
File D	File D moved to new location	File D (new location)	File D deleted	
File E	File E	File E	File E	File E

Required files from Day 1 FULL backup		Required files from Day 2 & Day 3 INCREMENTAL backups		Hard Drive after a restore to Day 3
	+	File F	=	File F
		File G		File G
		File D (new location)		File D (new location)
File E				File E

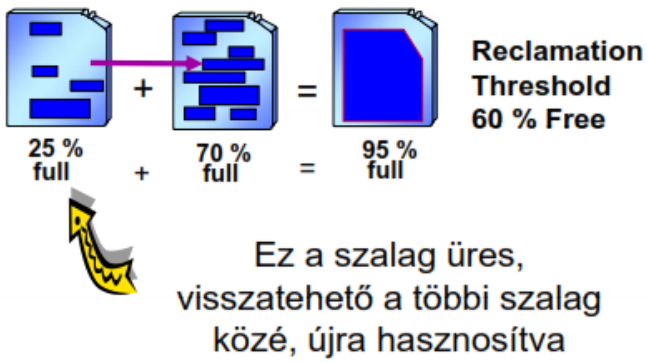
-

79. *Mit nevezünk kollokációnak (collocation)?*
    - Az **egy klienshez vagy klienscsoporthoz tartozó adatokat** egy szalagra vagy szalagcsoportra másolja. Csökkenti az adott visszaállítás során a szalagbefűzéseket, és rövidebb visszaállítási idő biztosítható.



-

80. *Mit nevezünk szalag-visszanyerésnek (tape reclamation)?*
    - A felhasználó által definiálható küszöbérték elérésekor az adatokat egy új szalagra másoljuk át. Ez a másolás időzíthető, kontrollálható

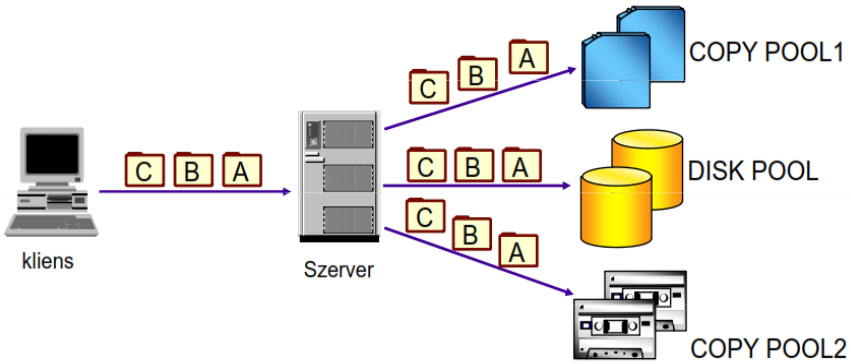


-

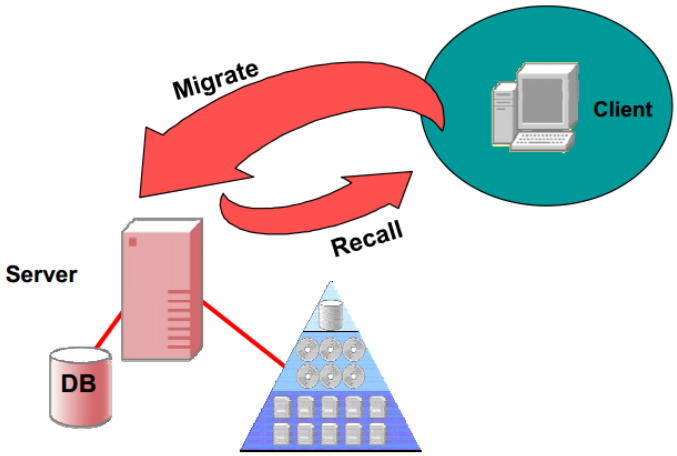
81. *Mi a LAN-free mentés és visszaállítás, mire jó ez?*
    - A kliens végzi a tényleges mentést, ő mozgatja az adatokat diszkről a szalagra

vagy SAN-on lévő diszkre, a szerver csak menedzseli a belső tárterületet~~, illetve ezt a folyamatot~~.

- Előnye, hogy a LAN hálózaton alapvetően csak a **metaadatok mozognak**, így a LAN-t nem terheli a mentési adatforgalom (csökkenti a LAN-túlterhelődés veszélyét a LAN-forgalom csökkentésével, és a mentési ablak időzítése rugalmasabb lesz), skálázható (mentőszerver-erőforrásokat is felszabadít), valamint megnöveli a tárolóeszközök kihasználtságát, és lehetővé teszi az ütemezett és házirend-alapú működést, a megosztott SAN-erőforrások terhelésének optimalizálására.

82. *Mi a párhuzamos mentés lényege, mikor hasznos?*
    - A párhuzamos mentés lényege az, hogy a cél storage pool-on kívül akár több másik copy storage pool definiálható, és ezekre szimultán (együttesen) történik az írás.
    - A cél storage poolok különbözőek lehetnek (diszk, merevlemez).
    - 

83. *Mit jelent a Zero down-time mentés, és mi az előnye?*
    - A tükrözött kötet, vagy snapshot tartalmazza az adott pillanatbeli másolatot, a mentés pedig az így készült másolatról készül.
    - Előnye, hogy nincs szükség az alkalmazás jelentősebb leállítására.
84. *Mi az a Disaster Recovery Manager, mi a feladata?*
    - Feladata a rendszer által támogatott katasztrófatervezés (up-to-date) és visszaállítás. Pontos útmutatók készítése vészhelyzet esetére, visszaállítási scriptek (katasztrófa utáni helyreállítási feladatok). Integrálódik a mentőrendszerrel.
85. *Mi a hierarchikus tárolókezelés szerepe?*
    - Megelőzni a “betelt a diszk”-jellegű problémákat a passzív adatok migrációjával (kliensről migrálunk bizonyos adatokat a szerverre). A kliensen csak egy leírót hagy, ha mégis kell az adat: automatikus aktiválás, visszatöltés. Házirend-alapú: pl. mióta nem használt. A mentőrendszerrel integrált.



86. *Soroljon fel 3 speciális archiválási követelményt.*
    1. **Előre definiált megőrzési idő**
        - Szerződéskötés pl 4 évre szólóan, addig megőrzés
    2. **Esemény alapú megőrzés**
        - Például életbiztosításnál, a biztosított halála után 70 évig
    3. **Törléstiltás, -engedélyezés**
        - Bizonyos állományok esetében a törlés felfüggesztve – pl. egy bírósági eljárás végéig
87. *Sorolja fel a mentés tervezési menetének 4 lépését.*
    1. **Vállalati stratégia (Corporate Guidelines)**
        - Ez az egész szervezetre vonatkozik, nagyvonalakban határozza meg a mentési tervet. Jogi minimumokat, mentési célokat, szempontokat, mentendő adatok típusát határozza meg, a mentés megvalósításának részleteivel nem foglalkozik.
    2. **Szolgáltatási szint meghatározása (SLA)**
        - Az adott telephelyen mik az elvárt és a biztosítandó szolgáltatási szintek.
        - Tipikusan használók egyeztetésével történik, pl. mentések típusa, adatok megőrzésének ideje, elvárt helyreállítási idők az egyes típusokra, a mentések gyakorisága (milyen mentés milyen gyakran legyen), az adatok megőrzésének ideje, a mentési ablak(ok) a különböző típusú mentésekhez
        - Konkrét példa: A használók az utolsó 6 hónap – 3 év bármelyik fájljának 1 hónapos pontossággal való visszaállítását kérhetik.
    3. **Mentési politika (Backup and Restore Policy)**
        - A mentési politika az a politika, amely az SLA-ban leírt követelményeket teljesíti
        - pl.: napi mentés, az SLA-ban meghatározott tárolási idők, annak az eldöntése, hogy legalább hány naponta legyen teljes mentés (a többi differenciális/inkrementális)
    4. **Mentési ütemterv (Backup Schedule)**
        - Ez konkrétan leírja, hogy mikor, melyik host melyik partícióját kell menteni
        - sokszor nincs külön leírva, hanem a mentőszoftver konfigurációjában rögzítik.
        - Az SLA és a mentési politika általános és ritkán változik.
88. *Mit nevezünk 80/20-as szabálynak?*
    - A hozzáférések 80%-a az adatok 20%-ának ismételt elérésére irányul.

/**********/

SZÁMOLÓS PÉLDA (ezeket kimásoltam a diákból, mert érthetőek úgy, ahogy vannak)

Egy szerverkörnyezetben 2TB adatmennyiséget kell menteni.
• Inkrementális mentést használunk.
• A változás mértéke kb. 10%/nap.

a. ***Határozza meg, hogy hetes mentési ciklus, és napi mentések esetén mekkora adatmennyiséget kell menteni az első 4 hétben!***
   - 1. nap: **teljes mentés** → 2 TB
   - **Inkrementum** → 2 TB*0,1 = 0,2 TB naponta
   - 1 hét → 2 TB + 6*0,2 TB = 3,2 TB
   - **4 hét → 3,2 * 4 = 12,8 TB**

b. ***Mekkora lesz a szükséges mentési időablak az egyes napokon, ha egy mentőeszköz effektív írási teljesítménye 100 GB/h?***
   - vasárnap (full backup) → 2TB / 100GB/h = **20 óra** (!!)
   - többi napon → 0,2 TB / 100 GB/h = **2 óra**

c. ***Hány mentőeszköz szükséges, hogy a mentési ablak 8 óránál ne legyen több?***
   - Legrosszabb: vasárnap → 20 óra, így (20/8=2,5→) **3 mentőeszköz** kell

d. ***Hány média szükséges a mentéshez, ha feltételezzük, hogy minden mentés új médiára kerül, és egy média maximális kapacitása 500 GB?***
   - Vasárnap: 2 TB / 500GB = 4 média
   - Hétköznap: 0,2 TB (= 200GB) = 1 média
   - Összesen: 4 + 6*1 = 10 média / hét
   - 40 média / 4 hét

e. **Egy adott időpont visszaállításához maximum hány média visszatöltésére van szükség?**
   - Legrosszabb: szombat
   - Visszaállítás: 1 full + 6 inkrementum
   - 4+6*1 = 10 média kell

/**********/

88. *Miért előnyös a centralizáció?*
   - Csökkenteni tudjuk a berendezések költségét
   - Csökkenteni tudjuk a szalagcserék költségét

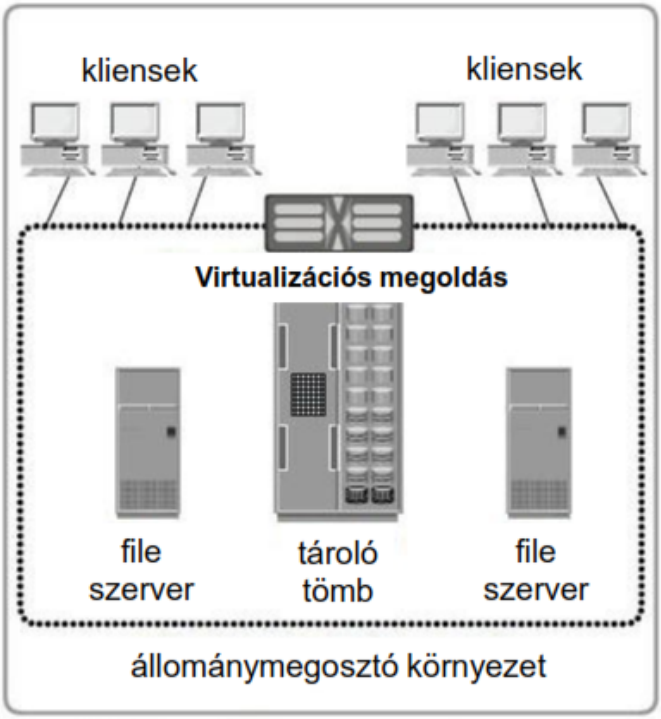
89. *Soroljon fel 6 okot, amiért sikertelen lehet a mentés*
   1. Nem megfelelő mentőrendszer - tervezési probléma
   2. Nem megfelelő menedzsment, emberi tévedések
   3. Nem elegendő kapacitás, mentési idő
   4. Hardverhibák
   5. Médiahibák
   6. Hálózati hibák

90. *Soroljon fel 6 okot a visszaállítás sikertelenségére*
   1. Tervezés, tesztelés gyakorlati hiánya
   2. Olvasatlan vagy üres szalagok
   3. Korrupt adatok

4. Nem teljes szalagállomány
5. Szoftver- vagy eszközhibák
6. Kapacitásproblémák

IRU_2013_felho, kulcsszavak: virtualizáció, felhő IT, SaaS, PaaS, IaaS, UCI, OCCI

91. *Mit nevezünk virtualizációnak?*
    - Virtualizáció: az a képesség, hogy egy fizikai rendszeren több(féle) operációs rendszer futtatható (és megosztják a rendelkezésre álló erőforrásokat)
92. *Mi a felhő IT?*
    - Szolgáltatások kívánságok szerint az erőforrások le/felskálázásával
93. *Vázolja fel szavakkal a virtuálisszerver-koncepciót!*
    - Logikailag elválasztja a szerverszoftvert a hardvertől. Egy virtuális szervert egy vagy több host is megvalósíthat, és fordítva: egy host több virtuális szervert is magába foglalhat. A virtuális kiszolgálókat (is) funkció szerint szokás hivatkozni (levelező-, adatbázis-, fájlszerver , stb.).
94. *Mik a virtuálisszerver-koncepció előnyei, és mik a hátrányai?*
    - Előnyök:
        1. redundancia
        2. közös erőforrás-gazdálkodás
        3. fizikai erőforrások
        4. új szerver gyors telepítése
        5. magas rendelkezésre állás
        6. leállás nélkül konfigurálható!!!
    - Hátrányok:
        1. bonyolultabb tervezés
        2. drágább konfiguráció, mint a hagyományos
95. *Mik a Cloud IT főbb tulajdonságai:*
    - Felhasználás alapján fizetett IT erőforrás (pay per use) igénybevételi modell. Hálózati hozzáférés egy megosztott IT-erőforrás-készlethez (pl. szerverek, tárolók, alkalmazások, szolgáltatások), amit (a szükséges verzióban) gyorsan lehet biztosítani, kevés szolgáltatói interakcióval.
96. *Mi a fájlszintű virtualizáció előnye a virtualizáció nélkülivel szemben?*
    - Nem kell tudnia a kliensnek/klienseknek, hogy hol van a keresett fájl fizikailag a szerveren.

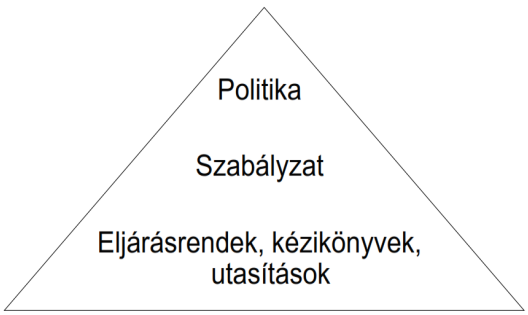


97. *Mi a Saas, Paas, Iaas?*
    1. **SaaS** (Software as a Service): szoftverszolgáltatási modell, amiben a felhasználó alkalmazáslicencet kap, igény szerinti (on demand) szolgáltatásként.
    2. **PaaS** (Platform as a service): IT-platform & megoldási csomag szolgáltatásként.
    3. **IaaS** (Infrastructure as a Service): IT-infrastruktúra mint szolgáltatás (tipikusan platformvirtualizációs környezet).
98. *Mi az a pay-per-use modell?*
    - Létező, kényelmes és igény-szerinti hálózati hozzáférés engedélyezésére konfigurálható IT-erőforrások (hálózatok, szerverek, tárolók, alkalmazások és szolgáltatások) megosztott készletéhez, amelyek könnyen létesíthetők és változtathatók minimális menedzsment-erőfeszítéssel vagy szolgáltatói interakcióval.
99. *A felhő IT 3 fajtája?*
    1. **Privát** felhő
        - Dedikált IT infrastruktúra egy bizonyos szervezet számára, nem osztozik mással. Drágább, biztonságosabb, mint a nyilvános felhő IT. Az adott szervezet telephelyén, vagy egy felhőből dedikálva.
    2. **Publikus** felhő
        - Az IT-infrastruktúrát egy szolgáltató a saját telephelyein működteti. Az ügyfél nem tudja, nem befolyásolja, hogy hol. Az infrastruktúrán tetszőleges ügyfelek osztoznak.
    3. **Hibrid** felhő
        - Az előző két modell optimális kombinációjaként, a hibrid felhő a privát felhő nyilvános elemekkel történő kiegészítése, kiterjesztése. Pl. egy vállalat alapvetően arra használja a **privát** felhőt, hogy megossza a fizikai és virtuális erőforrásait a hálózatán keresztül, de a **public** cloud igénybevételével akár ki is terjeszthetí ezeket az erőforrásokat, amikor éppen arra szüksége van. Továbbá a vállalat eldöntheti, hogy a sokszor többezer alkalmazás közül, melyeket szeretnék a

privát, és melyeket a **nyilvános** felhőn keresztül igénybe venni. Pl. a pénzügyi szoftvereinket a saját tűzfalunkon belül tarthatjuk, míg a kollaborációs szoftvereket a nyilvános felhőből lehet igénybe venni.

100. *Soroljon fel 2 Cloud IT szabványosítási törekvést.*
    - UCI, OCCI
101. *Mi az OCCI?*
    - Open Cloud Computing Interface, lényegében egy API a különböző felhő IT menedzsment feladatokhoz.

102. *Mit értünk **Információvédelem alatt?***
    - Az információ **bizalmasságának, sértetlenségének és rendelkezésre állásának** biztosítása. Az információvédelem nem más, mint az információval kapcsolatmáros biztonsági kockázatok folyamatos menedzselése.
103. *Mi a bizalmasság?*
    - Annak biztosítása, hogy az információ csak az arra **felhatalmazottak** számára elérhető.
104. *Mi a sértetlenség?*
    - Az információk és a feldolgozási módszerek teljességének és pontosságának megőrzése.
105. *Mi a "szabályozás piramis" 3 eleme?*



1. **Politika**
    - Hosszú távra szól
    - általános irányelvek, felelősségi körök
    - Legfelső szintű vezetői jóváhagyást igényel
2. **Szabályzat**
    - Középtávra
    - Legfelső szintű jóváhagyást igényel
3. **Eljárások, kézikönyvek, utasítások**
    - Rövid távra készülnek
    - Technológiai irányultságú intézkedések
    - Informatikai jóváhagyást igényel

106. *Milyen információbiztonsági szerepek vannak?*
1. **Információgazda**
    - adat-, hálózat-, rendszergazda
    - általában üzletági vezetők
    - Teljes felelősséggel tartoznak az adatért, hálózatért, rendszerért
2. **Információkezelő**
    - Neki delegálja az információgazda a napi teendőket
3. **Információhasználó**
    - bárki, aki az adatokat használja
    - cégen belül, vagy külső ügyfél

107. Sorolja fel az üzemeltetés biztonsági feladatokat.

1. **Védelem rosszindulatú kódok ellen**
   - spam, mobil kódok elleni védelem
   - vírusvédelmi szoftver:
     - vírusmegelőzés
     - vírusmentesítés
   - központilag vezérelt
   - felhasználók ne tudják semlegesíteni
2. **Adatmentés és -megőrzés**
   - Adatoknak a kritikussági szintnek megfelelő mentése
   - Adatgazda feladata
   - megőrzési idő után az adatok szakszerű megsemmisítése
   - fajtái:
     - teljes mentés
     - inkrementális mentés
     - differenciális mentés
3. **Naplózás**
   - Olyan attribútumokat naplózunk, amely biztonsági események észlelésénél keletkeznek, pl esemény, dátum stb...
   - NEM jelszót és hasonlókat!!!!!
   - Monitorozás, riasztáshoz szükségesek
4. **Biztonsági frissítések**
   - védelmi rések betömése
   - lehet manuális vagy automatizált
   - kritikus a gyorsaság → zero day attack (levlistáról)
   - Erre muszáj reagálnom. Bemásolom:
     - 19. Mit nevezünk "zero day attack"-nak? "A nulladik napi támadás (zero-day vagy zero-hour támadás) egy biztonsági fenyegetés, ami valamely számítógépes alkalmazás olyan sebezhetőségét használja ki, ami még nem került publikálásra, a szoftver fejlesztője nem tud róla, vagy nem érhető még el azt foltozó biztonsági javítás." - Wikipedia
     - Na ezt nekem nem fogadták el, ugyanis elvileg a fejlesztő már tud róla, csak még nem került publikálásra (ezért 0-day, 0 napja lett volna a fejlesztőnek a javítást közzétenni a támadás előtt), tehát majdnem jó a wikis megfogalmazás, csak pontot éppen nem ér.
5. **Adathordozók kezelése**
   - biztonságos szállítás, tárolás (HVAC, UPS)
   - ne szivárogjon adat, pl.:USB-ről stb..
   - biztonságos megsemmisítés
6. **Logikai hozzáférések kezelése**
   - Authorizáció → mihez van jogosultsága az adott usernek
   - Authentikáció → user azonosítása
     - alapelvek:
       i. Need-to-know → csak annyit tudjon a user, amennyi szükséges a feladata elvégzéséhez
       ii. csak ahhoz férjen hozzá a user, amihez jogosultsága van → minimális jogosultság
       iii. feladatok elhatárolása

7. **Kriptográfiai megoldások**
    - Bizalmassági szempontok
    - Integritásvédelmi szempontok

108. *E-mail-biztonság hogyan oldható meg?*
    - Spamszűrés
    - SMTP, POP protokoll, nem volt szempont régen az email-biztonság.

109. *Hálózatbiztonsági megoldások?*
    - Tűzfalak, behatolás-észlelő és -megakadályozó rendszerek (IDS/IPS).
    - Honeypot → védtelen eszköznek mutatja magát, megtámadják a vírusok és rosszindulatú SW-ek, mi pedig megismerhetjük a viselkedését, patternjét, így védekezhetünk a valódi eszközökben.

110. *Szerverek biztonsági megoldások?*
    - dedikált szerverek
    - távoli adminisztráció csak titkosított kapcsolaton keresztül
    - naplógyűjtés, riasztás
    - Patch-menedzselés
    - fizikai elhelyezés

111. *Definiálja az incidens fogalmát.*
    - Minden olyan esemény, ami negatívan befolyásolja az információs rendszerek biztonságát.

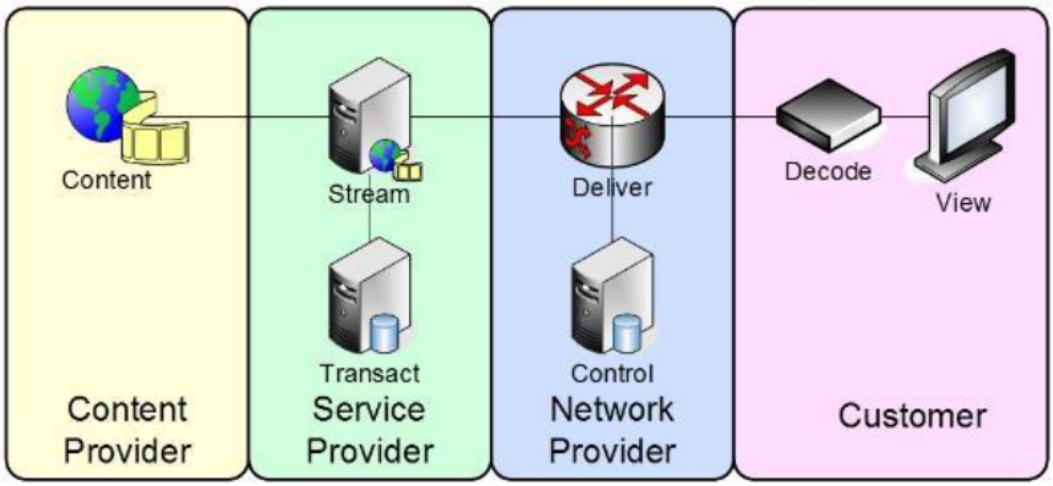
112. *Mit jelent a CSIRT?*
    - **C**omputer **S**ecurity **I**ncident **R**esponse **T**eam - **I**nformációbiztonsági **I**ncidens**e**lhárító **C**sapat , feladatuk az incidens feltárása, hibaanalízis, a nem sérült, kritkus rendszerek működésének megóvása, adatok gyors visszatöltése+NEGATÍV visszhang elkerülése → a rossz kommunikáció nagyobb bajokat képes okozni, mint maga az incidens!
    - Csapat összetétele:
        1. CSIRT vezető
        2. ügyfélszolgálati munkatárs
        3. jogi osztály munkatársa
        4. Felső vezető
        5. Rendszer és hálózati adminisztrátor
        6. PR munkatárs
        7. HR munkatárs
        8. Épületbiztonsági munkatárs

113. *Definiálja az eszkaláció fogalmát.*
    - Ha az incidens nem oldható meg egy előre rögzített időtartományon belül, akkor több szakértelem vagy hatáskör bevonása szükséges. Fajtái:
        1. Funkcionális eszkaláció → képzettebb szakember bevonása
        2. Hierarchikus eszkaláció → Felsőbb rétegek bevonása

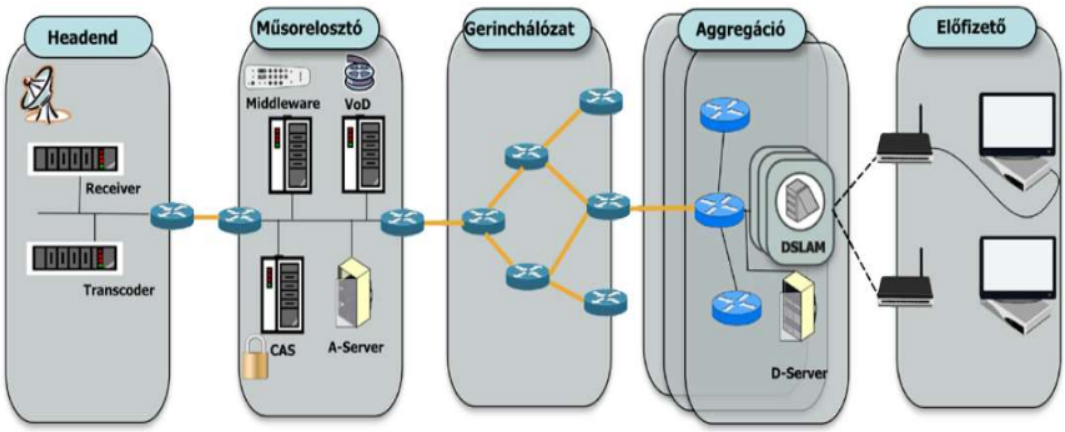
114. *Milyen részekből épül fel a bizonyítékgyűjtés és -kezelés?*
        1. **Részletes napló vezetése**
            - időbélyegek
            - bizonyítékok azonosítása (IP cím, MAC cím, stb..)
            - helyszín, ahol a bizonyítékot tárolták

2. **Bizonyítékgyűjtés**
    - teljes disk image készítése
    - kettesével dolgozni (tanú, tévedés mérséklése)
    - Minden bizonyítékot aláírni és dátummal ellátni
3. **Bizonyítékkezelés**
    - Bizonyíték megőrzése a tárgyalás lezárásáig, hozzáférés szigorú kontrollálása

115. *Milyen információ-szétosztási közegek vannak?*
    - Vezeték nélküli műsorszórás
    - Vezetékes műsorszórás
    - Internet
116. *Milyen szereplői vannak az IPTV-nek?*
        1. Content Provider
        2. Service Provider
        3. Network Provider
        4. Customer
    - áttekintő ábra:

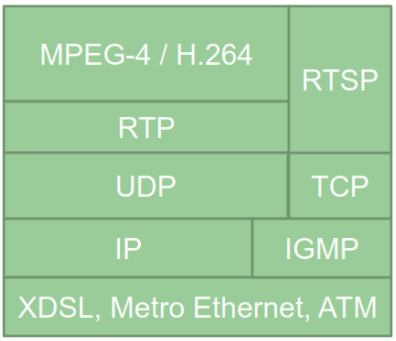


    - szakirányos médiatechnológia tárgyban részletesebb ábra:

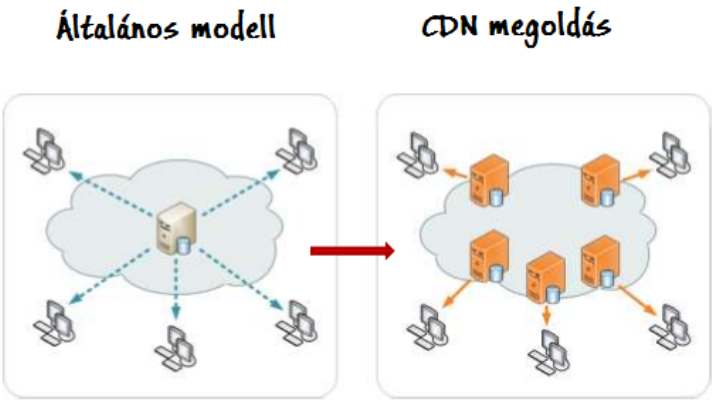


117. *Milyen QoE szempontok vannak IPTV esetén?*
    - Rendelkezésre állás
    - Késleltetés
    - “kockásodás”
    - Csatornaváltási sebesség
118. *Soroljon fel legalább 3 IPTV-protokollt.*
        1. UDP (User Datagram Protocol)
        2. RTP (Real Time Transport Protocol)
        3. TCP (Transmission Control Protocol)

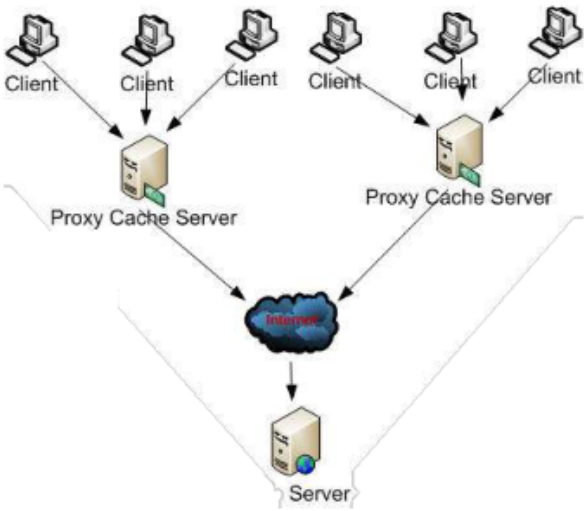
4. RTSP (Real Time Streaming Protocol)
5. IP (Internet Protocol)
6. IGMP (Internet Group Management Protocol)



119. *Milyen üzenetekkel történik az IPTV csatornaváltás, melyik protokoll érte a felelős?*
    - Az IGMP protokoll érte a felelős. Minden csatorna külön multicast-csoport (nem kell minden csoportbelinek egyenként elküldeni a csomagot, nem úgy, mint a unicastnál), a JOIN üzenettel (IGMP-beli) tudunk feliratkozni a multicast-csoportra, és a LEAVE üzenettel leiratkozni. Csatornaváltáskor az éppen nézett csatorna multicast-csoportjának küldünk egy LEAVE üzenetet, az újnak pedig egy JOIN-t.
120. *Mit jelent a CDN?*
    - Content Delivery Network,elsődleges célja gyorsítani a felhasználói forgalmat, és csökkenteni a hálózati forgalmat. Megvalósítása történhet szerverfarmmal, ami azt jelenti, hogy ugyanazt a tartalmat több szerveren is tároljuk, hogy kéréskor melyikről szolgálunk ki, az terhelés- és teljesítményalapú.

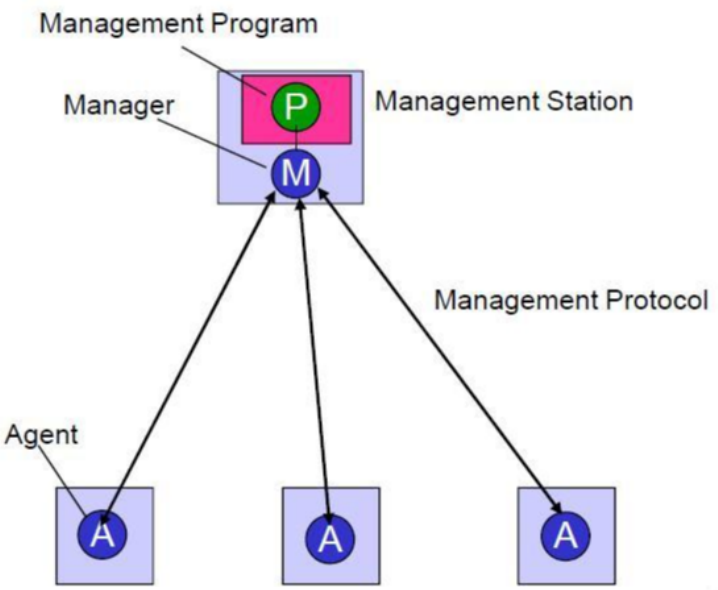


121. *Minek a rövidítése az ISP?*
    - Internet Service Provider
122. *Mi az a Caching proxy és mire használják?*
    - A caching proxy elsősorban az ISP-k internetes sávszélesség-igényének csökkentésére szolgál, vagyis elsődlegesen az ISP-előfizetők kiszolgálási késleltetését hivatott csökkenteni. A lényege, hogy vannak Proxy Cache szerverek, amik különböző technológiai megoldásokkal (csúszóablakos stratégia, prefix caching...) eltárolják a nemrég lekért csomagokat, vagy részeiket és így gyorsítják a kiszolgálást újbóli elérés esetén.



123. *Mi az az OTT?*
    - Over the Top TV, egy nemlineáris tartalomelérés biztosítása a cél, az OTT elnevezés a szolgáltatási modellre is utal egyben.
    - (Nemlineáris médiafogyasztás: Azt érjük el, töltjük le, nézzük/hallgatjuk, amit kiválasztunk, akkor, amikor akarjuk)

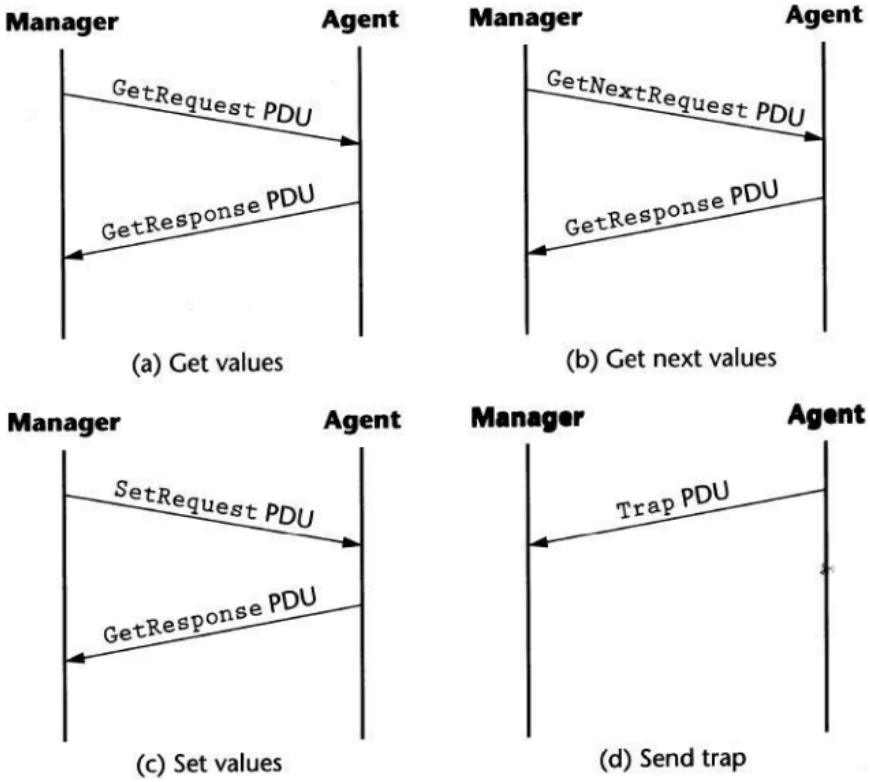
124. *Mi az az SNMP, mik a keretrendszer elemei?*
    - SNMP = Simple Network Management Protocol (egyszerű hálózatmenedzsment protokoll). A protokoll a **hálózatra kötött eszközök vezérlését, adatainak lekérdezését** szolgálja. Kialakításakor a cél az volt, hogy lehetőséget adjon számítógép-hálózatok távoli menedzselésére, meghatározott adatok lekérdezésével és beállításával, valamint lehetővé tegye a hálózat eseményeinek monitorozását.
    - Kliens-szerver felépítésű. A menedzselhető eszközön fut egy SNMP daemon, amely többnyire az UDP 161 és 162-es portokon figyel a kérésekre. A kéréseket a menedzselő állomás küldi, ez leggyakrabban egy számítógép, amely előtt a hálózat rendszergazdája ül.
    - Bővíthető, amit a MIB (Management Information Base ~ menedzsment információk csoportja) révén valósul meg.
    - Elemei:
        1. Management Station (Felügyeleti/menedzselő állomás):
            - Management Program (maga a program)
            - Manager (menedzser, aki kommunikál az ügynökkel)
        2. Management Protocol (Felügyeleti protokoll)
        3. Agent (ügynök, aki többnyire passzív, általában csak akkor szólal meg, ha a menedzser felkéri valamely feladat végrehajtására)



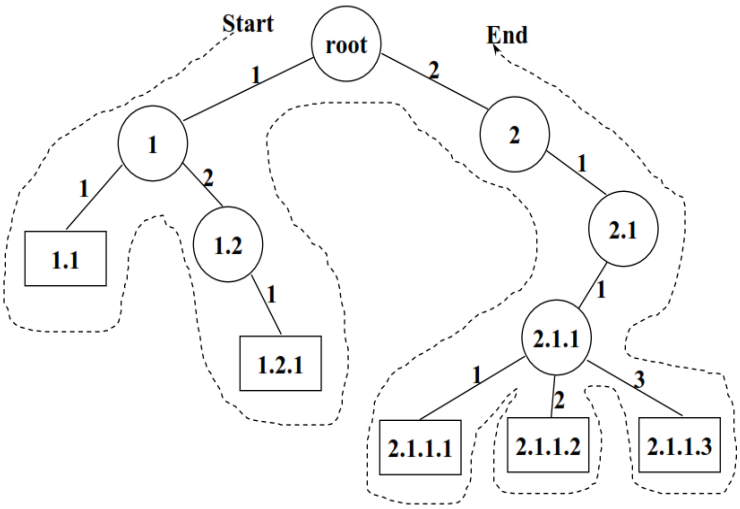
125. *Sorolja fel az SNMP-eljárásokat és üzenetszekvenciáit.*
    1. **Get-Request**
        - Egy vagy több értéket kér a Manager a Management Agent MIB-től
    2. **Get-Next-Request**
        - A lexikografikus leírásban a következő Object Identifiert kéri el a MIB-fán, úgy hogy megadja a jelenlegi Object Identifiert. Tehát a következő információ lekérése: ennek segítségével végig lehet lépkedni az információkon.
    3. **Get-Response**

- Válasz az értékkérésre

4. **Set-Request**
   - Beállítja az adott értéket (vagy feladatot) a menedzselt eszköz MIB-jében (egy objektumnak értékadás)
5. **Trap**
   - Egy kéretlen üzenet a menedzselt eszköztől, amit egy ott beállított esemény triggerelt; figyelmeztető üzenet lehet, pl. valami elért egy beállított értéket.



126. *<u>Mire szolgál egy MIB fa lexikografikus sorrendezése?</u>*
    - A lexikografikus sorrendezés a MIB-objektumok soros hozzáférésére szolgál. Hasznos olyan MIB-ek feltárásához, aminek a struktúrája nem ismert az NMS (Network Management System) számára.
    - Ha adott a MIB fa-struktúrája, az objektum-azonosító (OID, ObjectID) meghatározható a gyökértől az objektumig haladó úton.
    - Más néven preorder traversal (gyökér, balra, jobbra), vagy depth-first search (mélységi bejárás).



-

127. *Mi a MIB?*
    - Management Information Bases, objektumok gyűjteménye, adott menedzselt cél érdekében csoportosítva. Hierarchikus felépítésű (fa struktúrájú), bővíthető adatbázis.
    - Semmilyen explicit parancs nem adható ki az objektummal kapcsolatban, csak a levélelemeihez van hozzáférés
    - Némelyik objektum csak olvasható (pl. forgalomszámláló), némelyik értékét meg is lehet változtatni (read-only, read-write, write-only)
    - csak az alábbi szabványos adattípusok fordulnak elő benne: INTEGER, OCTET STRING, OBJECT IDENTIFIER, NULL, SEQUENCE, SEQUENCE OF
128. *Mi az az SMI?*
    - SMI = SNMP Management Information modell/struktúra iránymutatásokat ad, hogyan lehet MIB-eket, objektum-típusokat és objektum-azonosítókat definiálni.
        - Ezek ASN.1-ben (Abstract Syntax Notation 1) íródnak
            - ASN.1-ben definiáltak a szabályok is a menedzsment-információ kódolására vonatkozóan is (pl. hogyan kódolódik oktet-sztringgé)

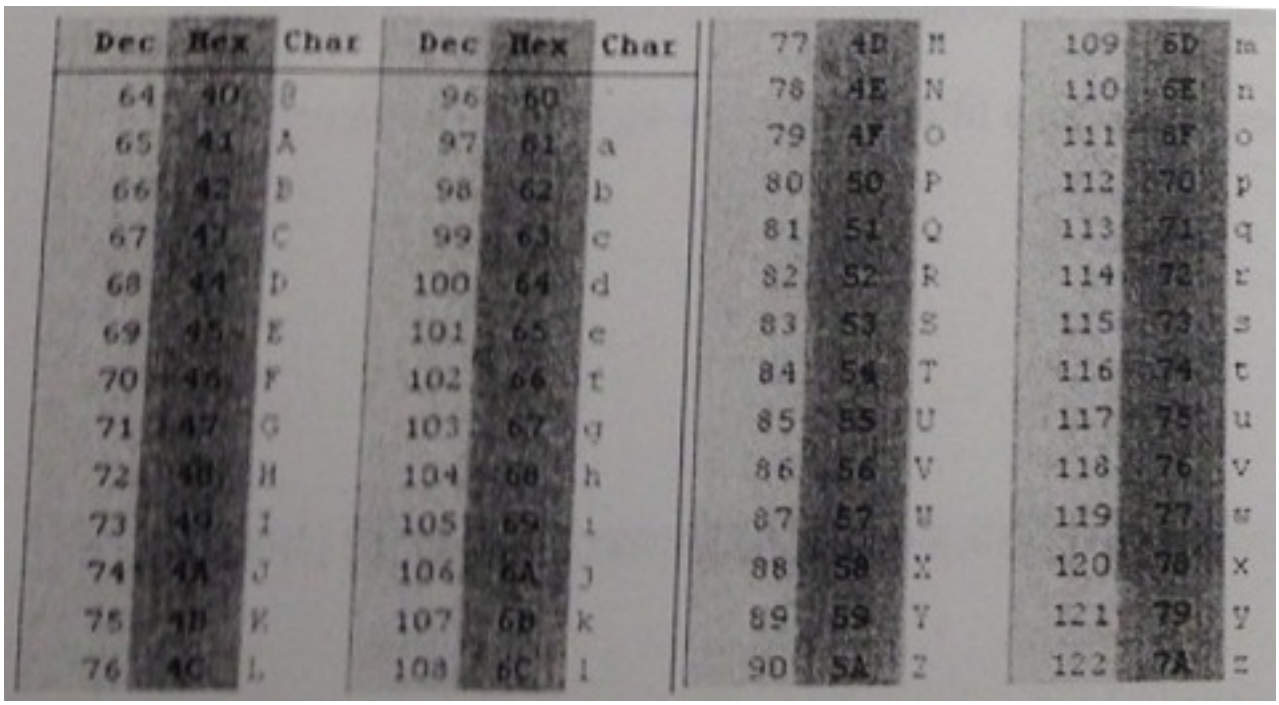
ASN.1-FELADAT

Kódolja ezt a szöveget ASN.1-gyel: "Think Good Talk Good Act Good" (Basic Encoding Rules), mint 6 darab sorozat sorozata ("sequence of 6 sequences") (Tag code: H'10, Universal: 00, Constructed: 1). Minden szó egy oktett-sztring-sorozatként legyen reprezentálva (Tag code: 04, Universal: 00, Primitive: 0), a szóközt nem kell kódolni.

http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters

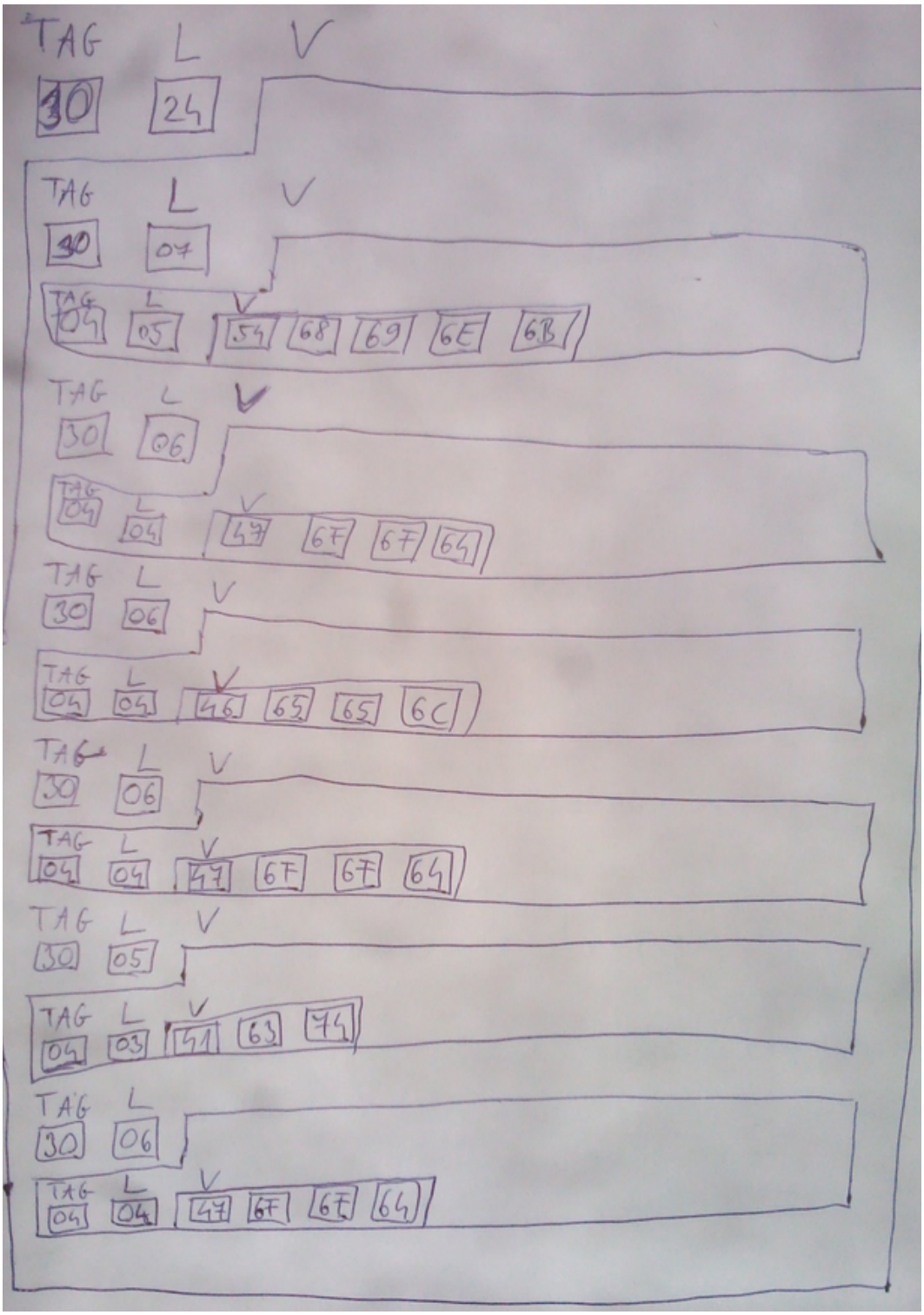
Dec	Hex	Glyph	Dec	Hex	Glyph	Dec	Hex	Glyph	Dec	Hex	Glyph
65	**41**	A	80	**50**	P	95	**5F**	_	110	**6E**	n
66	**42**	B	81	**51**	Q	96	**60**	`	111	**6F**	o
67	**43**	C	82	**52**	R	97	**61**	a	112	**70**	p

68	**44**	D	83	**53**	S	98	**62**	b	113	**71**	q
69	**45**	E	84	**54**	T	99	**63**	c	114	**72**	r
70	**46**	F	85	**55**	U	100	**64**	d	115	**73**	s
71	**47**	G	86	**56**	V	101	**65**	e	116	**74**	t
72	**48**	H	87	**57**	W	102	**66**	f	117	**75**	u
73	**49**	I	88	**58**	X	103	**67**	g	118	**76**	v
74	**4A**	J	89	**59**	Y	104	**68**	h	119	**77**	w
75	**4B**	K	90	**5A**	Z	105	**69**	i	120	**78**	x
76	**4C**	L	91	**5B**	[	106	**6A**	j	121	**79**	y
77	**4D**	M	92	**5C**	\	107	**6B**	k	122	**7A**	z
78	**4E**	N	93	**5D**	]	108	**6C**	l			
79	**4F**	O	94	**5E**	^	109	**6D**	m			

- 

- *(Bagyinszki Bence megoldása levlistáról:) Nem olyan nagy ördöngösség ez az ASN.1-kódolás, annyi az egész, hogy **mindig raksz egy TAG és egy LENGTH elemet (hexa) a tényleges adat elé**. A **TAG** jelzi az adat típusát, a **LENGTH** meg nyilván a hosszát.*
- *A TAG-ekre vannak kódok, pl. 30->SEQUENCE, 04->OCTET STRING, 02->Integer, 01->BOOLEAN, stb*
    - *04 az OCTET STRING tag kódja, minden szó elé kell írnunk, és utána azt a számot, ahány betűből áll*
    - 30 a sequence, mert
        - 00 - Universal, 1 - Constructed és decimálisan 16 a SEQUENCE, ami binárisan egymás mellett `00|1|10000` - ami decimálisban 48, hexában pedig **30**

- *A feladatban kellett egy sequence, ami octet stringeket (6 db-ot) tartalmazott. Az octet stringek belsejében az ASCII-kódokat kellett használni, ez alapján a megoldás (“**Think Good Feel Good Act Good**”):*
    - *30 24 (Sequence, 36 hosszú-> ezt legkönnyebb utólag megszámolni)*
    - *04 05 54 68 69 6E 6B (Octett string, 5 hosszú, "T", "h", "i", "n", "k")*
    - *04 04 47 6F 6F 64 (Octett string, 4 hosszú, "G", "o", "o", "d")*
    - *05 6C (Octett string, 4 hosszú, "F", "e", "e", "l")*
    - *04 04 47 6F 64 04 46 65 6F 64 (Octett string, 4 hosszú, "G", "o", "o", "d")*
    - *04 03 41 63 74 (Octett string, 3 hosszú, "A", "c", "t")*
    - *04 04 47 6F 6F 64 (Octett string, 4 hosszú, "G", "o", "o", "d")*
- *A vizsgán nem biztos, hogy ez a szöveg volt, de a lényeg az, hogy alapvetően így kell megoldani. Remélem, így érthető. :)*
- De akkor már az egyik barátom megoldását is beraknám ide, aki a szakirányán ilyennel már találkozott:
    - 30 24 (Sequence, 24 (hex) hosszú -> mert az "al-sequencek" méreteit kell összeadni
        - 07+06+06+06+05+06 = 36 (dec) = 24(hex))
    - 30 07 ('Think' szóhoz a sequence)
        - 04 05 54 68 69 6E 6B (Octett string, 5 hosszú, "T", "h", "i", "n", "k")
    - 30 06 ('good' szóhoz a sequence)
        - 04 04 47 6F 6F 64 (Octett string, 4 hosszú, "g", "o", "o", "d")
    - 30 06 ('Feel' szóhoz a sequence)
        - 04 04 46 65 65 6C (Octett string, 4 hosszú, "F", "e", "e", "l")
    - 30 06 ('good' szóhoz a sequence)
        - 04 04 47 6F 6F 64 (Octett string, 4 hosszú, "g", "o", "o", "d")
    - 30 05 ('Act' szóhoz a sequence)
        - 04 03 41 63 74 (Octett string, 3 hosszú, "A", "c", "t")
    - 30 06 ('good' szóhoz a sequence)
        - 04 04 47 6F 6F 64 (Octett string, 4 hosszú, "g", "o", "o", "d")
- A fenti számpárosok mind 8 biten kódolt hexadecimális számokat jelölnek!!!
- **A következő oldalon látható hozzá az ábra!!!!**



[3]

129. *Sorolja fel az SNMP 3 biztonsági szolgáltatását.*
    1. **Authentication**
        - Az Agent limitálni szeretné a MIB-hozzáférést authentikálatlan menedzsereknek.
    2. **Access**
        - Az Agent különféle privilégiumokat akar kiosztani különböző menedzsereknek.
    3. **Proxy-szolgáltatás**
        - Special Access + Authentication

[3] **ÁBRA a 128-as feladat ASN-es megoldásához**

130. *Mi az SNMP community, és milyen fajtái vannak?*
    - Ez egy kapcsolat egy „agent” és egy csoportmanager között – itt az authentikáció, hozzáférés és proxy lehetőségek is definiáltak közöttük.
131. *Milyen hozzáférési kategóriái vannak (Access Policy) az SNMP communitynek?*
    1. **SNMP MIB View**
        - objektumok részhalmaza a MIB-en belül
        - különféle MIB nézetek lehetnek definiálva a communityk számára
    2. **SNMP Access Mode**
        - Egy hozzáférési mód {READ-ONLY, READ-WRITE} definiálható a community-nek
        -

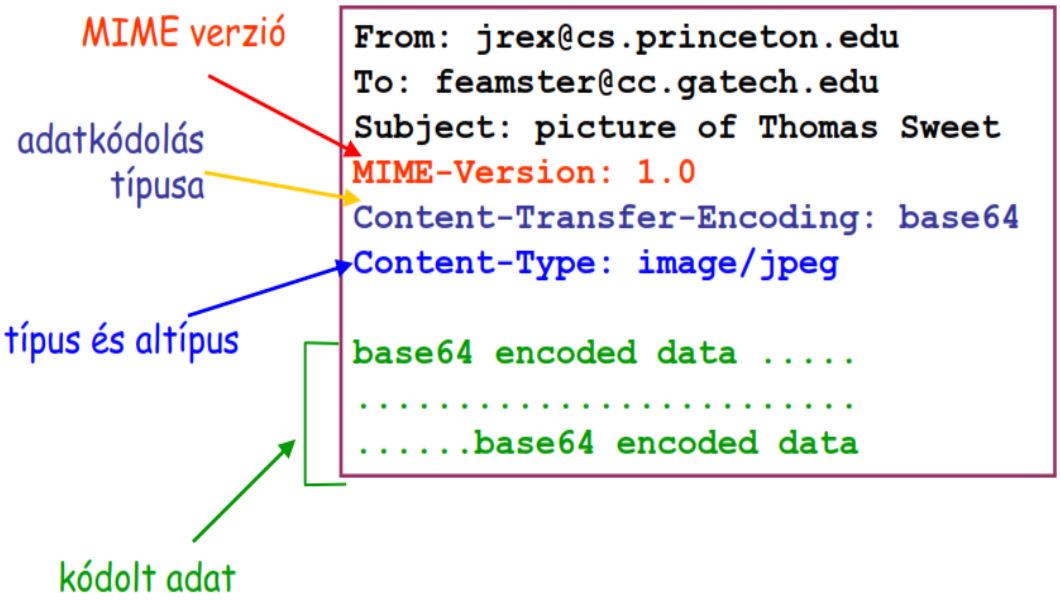
MIB ACCESS Category	SNMP Access Mode	
	READ-ONLY	READ-WRITE
read-only	get és trap eljárásokhoz	
read-write	get és trap eljárásokhoz	get, set és trap eljárásokhoz
write-only	get és trap eljárásokhoz – de az érték implenetációfüggő	get, set és trap eljárásokhoz – de az érték implenetációfüggő get és trap esetben
not accessible	nem használható	

    3. **SNMP Community Profile**
        - MIB view és egy access mode kombinációja

IRU_IT_Szolgaltatasok_13 diasor, kulcsszavak: e-mail, MIME, POP, IMAP, SMTP, RAS, RAW, PCL

132. *Mitől szolgáltatás egy szolgáltatás?*
    - Megtervezés
    - Beüzemelés
    - Fejlesztés
    - Monitorozás
    - Karbantartás
    - /Támogatás/
133. *Soroljon fel alapszolgáltatásokat.*
    - E-mail
    - Authentikáció
    - Távoli elérés
    - Nyomtatás
    - DNS
134. *Mik a jól karbantartható szolgáltatás jellemzői?*
    - Egyszerű
    - Kevés függőséget tartalmaz
    - redundáns HW
    - Szabványos SW és HW
135. *Mi a különbség a vastag- és vékonykliens-szolgáltatások között?*
    1. Vastagkliens-szolgáltatás: Nagyrészt a gazda gépén fut
    2. Vékonykliens-szolgáltatás: Nagyrészt a szervergépen fut
136. *Mit jelent az SPF és mi okozhatja?*
    - SPF = Single Point of Failure, például protokoll gateway-ek használata idézheti elő.
137. *Sorolja fel az e-mail-küldés lépéseit.*
    1. Üzenettovábbítás → ahogyan az e-mail szerverről szerverre jut
    2. Kézbesítés → amikor az e-mail a fogadó mailbox-ába kerül
    3. Üzenetlisták feloldása → Amikor a listacímre küldött levél megsokszorozódik, és így kerül továbbításra
138. *Milyen részekből épül fel egy e-mail?*
    1. Fejléc → Címzés, tárgymegjelölés
    2. Törzs
139. *Milyen megszorítások adódnak e-mail küldésekor és mi rájuk a megoldás?*
    1. Nem szöveges adat küldése → MEGOLDÁS: konvertálás Base64-es kódolással. Nem ASCII-ból ASCII-konvertálás
    2. Több adategység küldése → MEGOLDÁS: több emailt egy üzenetbe csomagolunk, elválasztásuk pl. stringgel
140. *Mi a MIME, és mikből áll?*
    - MIME = Multipurpose Internet Mail Extension, hozzáadott fejlécek a törzs leírásához
    - Részei:
        1. MIME-Version

2. Content-Type
3. Content-Transfer-Encoding



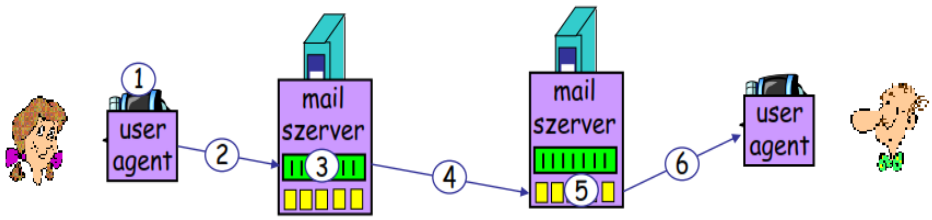
-

141. *Mik az e-mail-címek részei?*
    - Helyi mailbox és domain név. Pl. Tibiatya@humbakfalva.com, itt Tibiatya mailbox és humbakfalva.com a domain
142. *Mi az az SMTP és mi a feladata?*
    - SMTP = Simple Mail Transfer Protocol. A mail üzeneteket szerverek sorozata szállítja, a szerverek a bejövő üzeneteket üzenet sorbaállítják, hop by hop módszer.
    - Minden “hop” beírja az azonosítót a fejlécbe.
    - Az SMTP egy kliens-szerver protokoll, kliens a küldő szerver, szerver a fogadó szerver. (küldés kezdésekor a kliens a tényleges kliens, aki a mailt küldi, a szerver az első fogadó mailszerver).
    - Megbízható adattovábbítás TCP protokoll (on port 25) felett.
    - Az SMTP egy “PUSH” protokoll, ugyanis csak benyomja a következő szerver mailboxába, amint megkapta a “response” választ a “command” üzenetre.
    - Továbbítás 3 fázisa:
        1. Handshaking (kézrázás)
        2. Üzenettovábbítás
        3. Lezárás

-

Eset:
Anna üzenetet küld Bélának

1) Anna az UA-t használja, hogy üzenetet írjon ide: beela@nagyceg.com
2) Anna UA-je üzenetet küld a mail szerverének; az üzenet bekerül egy sorba
3) A kliens oldali SMTP egy TCP kapcsolatot nyit Béla mail szerverével
4) Az SMTP kliens átküldi Anna üzenetét a TCP kapcsolaton
5) Béla mail szervere berakja az üzenetet Béla postaládájába (mailbox)
6) Béla aktiválja az UA-ját az üzenet elolvasásához



-

143. *Mi a POP, mik a korlátai?*
    - POP = Post Office Protocol. Célja, hogy időszakosan is el tudjuk érni a mailjeinket, letölthessük és tetszés szerint manipulálhassuk őket, ha nincs csatlakozva.
    - Tipikus user-agent interakció POP szerverrel:
        1. Kapcsolódás a szerverhez
        2. e-mailek leszedése
        3. Az üzenetek “új”-ként való tárolása a PC-n
        4. kapcsolat lezárása a szerverrel
    - POP korlátai:
        - Nem könnyen kezel többszörös mailboxokat
        - nem az üzenetek szerveren való tárolására tervezték
        - a mailboxhoz a többszörös klienshozzáférés nehéz
        - nagy hálózati sávszélességet igényel
144. *Mi az IMAP?*
    - IMAP = Interactive Mail Access Protocol.
    - Connected” és „Disconnected” módok támogatása
    - Egyszerre több kliens is csatlakozhat a mailboxra; detektálja a más kliensek által a mailboxon történt változtatásokat
    - Hozzáférés az üzenetek MIME részeihez & részleges letöltés
    - A kliens tud létrehozni, átnevezni, és törölni mailboxot
    - A kliens tud egyik folderből másikba áthelyezni üzenetet
    - Az üzenet letöltése előtt a szerveren lehet keresést indítani rá
145. *Jellemezze a webes mailt.*
    - User agent: hagyományos Web browser
    - A felhasználó HTTP-n kommunikál a szerverrel

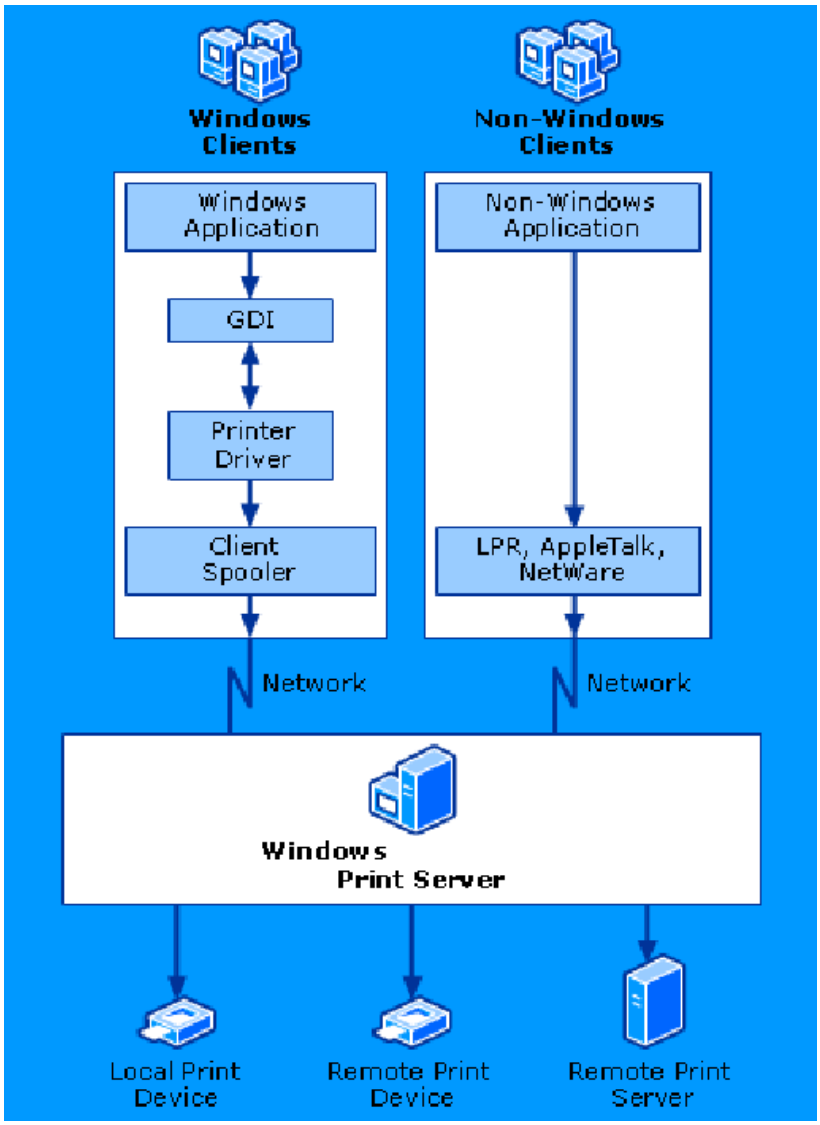
- A weboldalak a folderek tartalmát jelenítik meg
- A szöveget egy „form”-ba írjuk, majd „submit” a szervernek
- “POST”-kérés és -adatfeltöltés a szerverhez
- A Szerver SMTP-vel küldi az üzenetet más szerverhez
- Könnyű az anonymous e-mail (pl. spam) küldése

146. *Vállalati e-mail fontos jellemzői*
    1. **Privacy Policy:**
        - A hely e-mail policy-jével mindenki legyen tisztában (...és fogadja el...)
        - A vezetőség dönthet úgy, hogy privát levelek küldését céges címről nem támogatja.
    2. **Namespace-ek**
        - Az e-mail cím a vállalat namespace-énak egyik legláthatóbb része
        - Ugyanaz legyen a belső és a külső e-mail cím
        - Legyen standard címformátum – például → first.last
    3. **Megbízhatóság**
        - Az e-mail alapeszköz. Mindig Jól Működjön
        - Melegtartalék az egész rendszerről
        - Ha nem lehetséges a hot swap (mivel eléggé költséges) → készenléti terv, begyakorolt lépések a hibából való helyreállásra
    4. **Egyszerűség**
        - Korlátozzuk a szükséges gépek számát
        - Kerüljük a protokoll- vagy formátum-gateway-ek használatát → mert ezek SPF
    5. **Automatizáltság**
        - Az e-mail account létrehozása legyen az általános account-készítési folyamat része
        - A búcsúzó kollegák e-mailjeit nem forwardoljuk; listákról töröljük
        - Accountok másolása szerverek között
    6. **Monitorozás**
        - Hálózat: ping (ICMP echo üzenetet küld)
        - TCP 25-ös port elérhető?
        - Visszapattanó üzenetek – diagnosztikai info
        - Naplóállományok (pl. üzenet-mennyiségek, előrejelzéshez)
    7. **Skálázhatóság**
        - Növekvő felhasználói bázisnak
        - Forgalmi börsztök kezelése
        - Óriási, akkumulálódó adattömeg tárolása
        - Mail spool használata segít
        - Üzenetméret-korlátozás is segít (időszakosan)
    8. **Security**
        - Ellenőrzés a szervereken ÉS a felhasználói gépeken is
        - A tűzfallal együtt: vállalati biztonsági stratégia

147. *Mi a RAS? Mondjon példákat rá!*
- RAS = Remote Access Service. A külső alkalmazások eléréséhez tűzfallyukasztás → a tűzfalat az alkalmazás protokolljához rendelt porton átjárhatóvá kell tenni
- Hozzáférésre jogosultak léphetnek be a vállalati hálózatba, alapos tervezést igényel

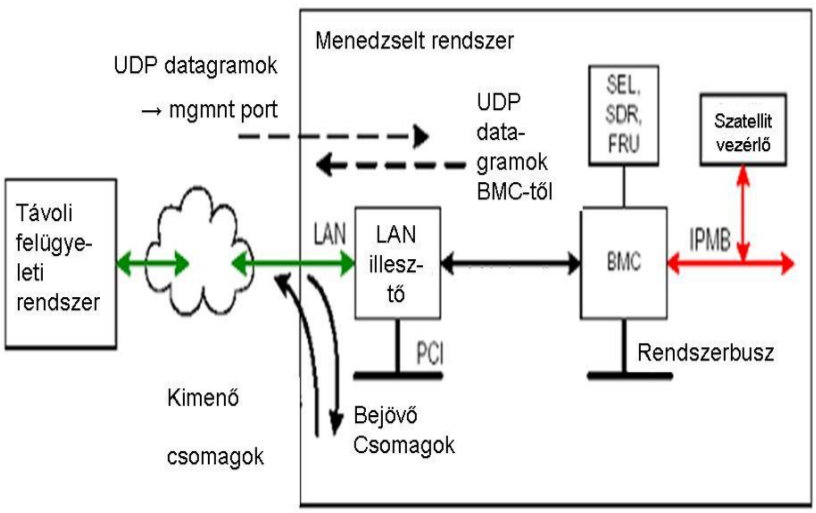
- Példák: Remote Desktop (Windows-alapú), VNC (bármilyen OS alatti gépre), TeamViewer, Join.Me, Mikogo

148. *Nyomtatás Windows-szerveren.*



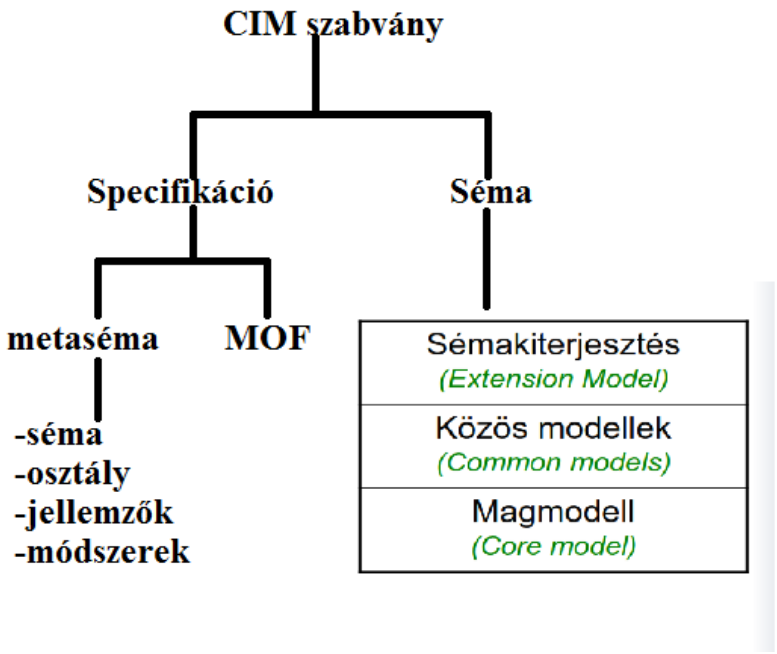
149. *Soroljon fel hálózati nyomtatási protokollokat.*
    1. klienstől a szerverig:
        - SMB – Server Message Block
        - LPR (LPD) - Line Printer Remote
        - IPP – Internetwork Printing Protocol
    2. szervertől a nyomtatóig:
        - LPR
        - IPP

150. *Mit nevezünk RAW-nak?*
    - „Nyers”, a nyomtató által emészthető formátum, PCL (Printer Command Language) és PostScriptben írva

151. *Mi az IPMI?*
    - IPMI = Intelligent Platform Management Interface
    - A számítógépek hardver- és firmware-eszközei számára határoz meg közös interfészeket, amelyekkel a rendszer-adminisztrátor a rendszerek állapotát ellenőrzés alatt tarthatja és menedzselheti.
    - The Intelligent Platform Management Interface (IPMI) is a standardized computer system interface used by system administrators for out-of-band management of computer systems and monitoring of their operation.
152. *Mik az IPMI jellemzői?*
    - OS-független, sőt OS hiányában is képes a rendszer adminisztrátor a szükséges adatokat lekérdezni külön soros vonalon, vagy serial over LAN-on keresztül.
    - A szabvány előírja a riasztási mechanizmust is. SNMP protokollon keresztül!
    - Az IPMI (az állapot adatok közel valós idejű figyelése révén) problémák megelőzésére is alkalmas. Javítja a rendszerek biztonságát is.
153. *Mikből épül fel az IPMI?*
    - Egy fő és több mellékvezérlőből állhat. Fővezérlő: BMC (Baseboard Management Controller) és a mellékvezérlők = szatellitek
    - Nagyobb rendszerben a mellékvezérlők az IPMB (Intelligent Platform Management Board/Bus) interfésszel kapcsolódnak a BMC-hez. A BMC a mellékvezérlőket képes más BMC-khez csatolni az IPMC (Intelligent Platform Management Chassis) segítségével.
    - Az egészet a RMCP (Remote Management Control Protocol) felügyeli, amit az IPMI definiál.
    - 

154. *Mit jelent az, hogy az IPMI nem ügynök-alapú megoldás?*
    - A nem ügynök-alapú megoldás azt jelenti, hogy nem szoftverügynökökön alapul a megoldás, hanem vezérlőkön.
155. *Milyen tárolói vannak az IPMI-nek?*
    - FRU (Field Replacable Unit), ami tárolja a cserélhető eszközök leltárát, valamint az SDR (Sensor Data Records) tárolóban találhatóak az eszközben működő érzékelők adatai.

156. *Mi az a CIM és "ki" hozta létre?*
    - CIM = Common Information Modell
    - Hierarchikus, objektum-orientált menedzsment információs modell.
    - Definiálja egy információtechnológiai környezetben üzemeltetett eszközök objektum-alapú reprezentációját.
    - CMI is an interface between content providers and service providers, which does not directly involve the end user. The scope of the standard covers the entire off-deck content management lifecycle but does not include implementation or behavior beyond the API. Therefore it can accommodate a broad range of services and service policies.
    - A DMTF (Distribute Management Task Force) fejlesztette ki, úgy, ahogy a DMI-t is.
157. *Mik a CIM jellemzői?*
    - Lehetővé teszi a különböző gyártótól származó berendezések üzemeltetési adatainak kicserélését, az aktív vezérlést, beavatkozást.
    - Objektumorientált menedzsment modell, UML nyelven van leírva. Ebben a leírásban a modell a menedzselt elemeket (HW eszközök, SW-ek) külön CIM osztályokként definiálja, a köztük lévő kapcsolatokat pedig CIM kapcsolatokként.
    - A menedzsmentadatok számára szabványos keretet biztosít.
    - A CIM a rendszerelemek állapotának nem csak a lekérdezését, hanem a menedzselt elemek manipulálását is lehetővé teszi.
158. *Sorolja fel a CIM-infrastruktúra elemeit.*
    - metaséma
    - szintaxis
    - szabályok
    - formátum (MOF - Managed Object Format)
159. *Egy CIM-profilt mi azonosít?*
    - A neve, a felhasználó szervezet neve és verziószáma.
160. *Miből áll a CIM-szabvány?*
    1. Specifikáció
        - Definiálja a más menedzsment-modellekkel való integráció részleteit.
    2. Séma
        - Az aktuális modell leírását tartalmazza.
    - **Előzetesen is összefoglalva:**



○

161. *Jellemezze a CIM specifikációt.*
    - Metasémákat, a metaséma elemeit, és minden egyes elemhez a szabályokat tartalmazza.
    - A metaséma a modell formális leírása, tartalmazza: modell kifejezését, modell használati folyamatát, modell szemantikáját
    - Definiálja továbbbá a MOF-ot, amely definiálja az osztályokat és az eseteket.
162. *Sorolja fel a CIM metaséma elemeit:*
    1. **Sémák (Schemas)**
        - A sémaosztályok csoportja ugyanazon csoporttulajdonossal.
        - A sémát adminisztrációs céllal használjuk
    2. **Osztályok (Classes)**
        - Az osztály egy menedzselt objektum jellemzőit fogja össze
    3. **Jellemzők (Properties)**
        - Az osztály jellegének kifejezésére szolgáló érték, egyedinek kell lennie az osztályon belül.
    4. **Módszerek (Methods)**
        - A módszer egy meghívható művelet, az osztályára értelmezett, és azon belül egyedinek kell lennie. Egy osztályban nulla vagy több módszer lehet.
163. *Mi a CIM MOF?*
    - MOF = Managed Object Format, szöveges leírása az osztályoknak, a kapcsolatoknak, jellemzőknek, referenciáknak, módszereknek és esetdeklarációknak, a hozzájuk rendelt minősítőkkel együtt. Megjegyzések is lehetnek benne.
    - Unicode vagy UTF-8 kódolású lehet.
164. *Jellemezze a CIM-sémát.*
    - A menedzsmentsémák az építőelemei a platformok és alkalmazások menedzselésének: például az eszköz konfigurálásának, a teljesítmény hangolásának, a változáskövetésnek.
165. *Milyen modellekből épül fel a CIM-séma?*

1. **Sémakiterjesztés (Extension Model)**
   - A sémakiterjesztés azért szükséges, mert a rendszerelemek jellemzően termék- illetve gyártó-specifikusak. A sémakiterjesztéssel tetszőleges tulajdonságok és viselkedések leírhatóak.
   - Új tulajdonság vagy metódus hozzáadása létező séma létező osztályához.
2. **Közös modell (Common Model)**
   - Egy közös modell egy adott technológia vagy implementáció esetére vonatkozik. Például: a hálózati eszközökre, a rendszeren futtatott operációs rendszerekre, stb.
   -
3. **Magmodell (Core Model)**
   - A magmodell osztályok, a kapcsolatok, jellemzők

     - Applications Alkalmazások
     - Event Eseménykezelés
     - Network Hálózatok menedzselése
     - Support Terméktámogatás
     - Database Adatbáziskezelés.
     - Interop A webalapú vállalati menedzsment (WBEM)
     - Physical A fizikai eszközkészlet kezelése
       - Pl. a különböző bővítőkártyák és kábelezések leírásai

     és módszerek készlete, azoké, amelyek a menedzselés valamennyi területére vonatkoznak.
   - A magmodell definiálja a menedzselt környezet alapvető osztályait és asszociációit.
   - Minden osztály a CIM_ManagedElement osztály leszármazottja.
   - A magmodell a menedzselt rendszer „alapszótára”.

166. *<u>Mi a DMI?</u>*
   - DMI = Desktop Management Interface, felhasználói végberendezések és szerverek alkatrészeinek kezeléséhez (menedzsmentjéhez) nyújt szabványos keretet.
   - A DMI a Rendszer Menedzsment BIOS (SMBIOS) része, ez teszi szabványossá a számítógép-konfigurációról szóló adatok hozzáférését a felhasználók vagy erre feljogosított alkalmazások számára.
   - A DMI tehát szabványos módon kérheti le a BIOS-ból a számítógép alkatrészeiről, felépítéséről az adatokat.
167. *<u>Mi a WBEM?</u>*
   - WBEM = Web Based Enterprise Management, A WBEM rendszermenedzsment technológiák olyan készlete, ami az elosztott IT környezet menedzselésének egységesítésére szolgál.
   - Alapjai: CIM standardok és Internet technológiák (CIM infrastruktúra és séma, CIM-XML, CIM over HTTP, WS-Management, SNMP)
   - A WBEM a CIM web-alapú implementációja, beleértve a protokollokat, amelyekkel detektálhatók és elérhetők más CIM implementációk.
   - Tartalma:
     1. protokol(lok)
     2. lekérdezési eljárások
     3. felismerési mechanizmusok

4. leképezések

168. *Sorolja fel a WBEM-technológia kulcselemeit.*
    - alkalmazások távmendzsmentje
    - egy alkalmazás több esetének egyetlen egységként való menedzselése
    - standard interfész különböző alkalmazások távmenedzseléséhez
    - az alkalmazás menedzsment leválasztása a kliensről
169. *Foglalja össze a WBEM-architektúrát röviden.*
    - Az operátor felhasználói felületen éri el a menedzsmentrendszert, azt, hogy milyen a felhasználói felület, a WBEM nem köti meg. → következésképpen a a felhasználói felületet változathatjuk anélkül, hogy a többi elemet módosítanánk.
    - A WBEM kliens, hogy megtalálja a WBEM szervert, a menedzselni kívánt eszköz számára létrehozza a kérést tartalmazó XML üzenetet, amit HTTP- vagy HTTPS-protokollon továbbít.
    - A kliens kizárólag a modellel kommunikál, a modell pedig a valóságos HW-rel vagy SW-rel!!! A kommunikációt a szolgáltatók kezelik le.
170. *Mit szükséges egy eszközfejlesztőnek (vagy szolgáltatónak) elkészítenie ahhoz, hogy eszköze vagy szolgáltatása szabványosan menedzselhető legyen?*
    - A modellt
    - a “szolgáltatókat”
171. *Soroljon fel 3 ismert WBEM implementációt.*
    - Solaris WBEM Services, IBM TIvoli Monitoring, Open Pegasus, Purgos, SMI-S
172. *Sorolja fel az üzemeltetési politika 5 elemét.*
    1. Rendszerüzemeltetési etika
        - Egy szervezetben az etikai elvárásokat feladatkörökhöz illeszkedően a napi teendők tükrében érthető módon célszerű megfogalmazni. pl.: szakmai viselkedési kódex, felhasználói viselkedési kódex
    2. Névtér-politika
        - A névtér (namespace) bizonyos típusú elemek (pl. személynevek, földrajzi nevek, műszaki kifejezések, stb.) felsorolása és összefüggéseinek megadása egy rögzített szabályokon alapuló tároló elrendezésben
        - A névtér-elemek vonatkozhatnak valóságos tárgyakra, élőlényekre és elvont fogalmakra is.
        - A névterekre vonatkozóan határozott, rögzített, írott egyértelmű politika kell.
    3. A rendkívüli helyzetek teendői
    4. Változáskezelés
    5. IT biztonsági politika
173. *Sorolja fel, milyen névterek vannak! (ez elég pongyola)*
    - absztrakt névtér
    - konkrét névtér
    - egyszerű névtér
        - A névtér elemeinek egy és csak egy értelmezése lehet.
    - hierarchikus névtér
        - Konténereket is tartalmaz valamilyen elrendezésben.

174. *Sorolja fel a névtérpolitika kiterjedéseit.* (ez is bullshit megfogalmazás, sorry eddig tartott az energiám :D)
    - Elnevezési politika
    - Élettartam p.
    - Láthatósági p.
    - Konzisztencia p.
    - Újrahasználati p.
    - Védelmi p.
175. *Sorolja fel a névválasztás főbb módszereit.*
    - **Formális**
        - Kötött, szigorú szabályok szerint adunk neveket, pl.: gépnév: pc + 4 számjegy, login név: vezeték első hat jegye + keresztnév kezdőbetűje + n jegyű azonosítószám
    - **Téma szerinti**
        - A különböző típusú nevek különböző téma köré csoportosulnak, pl. szerverek csillagok, printerek bolygók stb.
    - **Funkcionális**
        - Felhasználói szerepek (admin, titkár, vendég)
        - A gép által betöltött szerep (DNS, cpuserver12, web001)
    - **“Nincs szabály”-módszer**
        - Mindenki úgy nevez el valamit, ahogy ő gondolja, az ütközések feloldása elsőbbségi alapon történik.
176. *Mi a névtérséma?*
    - Egy sématípus megadása, kizárólag neveket és definíciókat tartalmaz.
177. *Mi az névtér alkalmazás profil?*
    - Egy sématípus leírása, az alkalmazási környezetben használt nevek leírása.
    - Szemantikus definíciót is tartalmazhat.

178. *Alapvető műveletek:*
    - cd :könyvtárak közötti navigációhoz
    - mkdir :könyvtár létrehozása
    - rmdir :könyvtár törlése
    - ls :könyvtár tartalma
    - echo $PATH :végrehajtható parancsok helyei → könyvtárai!!! legalábbis megtekintésen így fogadták el, mind1, h a diában is ez van szó szerint
        - pl.: /usr/local/bin/:/usr/bin:/bin/
    - export :környezeti változó beállítása pl. PATH=$PATH:/new/directory/path
    - man :help a parancsok használatához
    - history :eddig használt parancsok listája
    - & :parancs után írva: fusson a háttérben ls & -> [1] 23142
    - fg :futó job visszahozása előtérbe fg 23142
179. *Szövegszerkesztés:*
    - cat :teljes file szövegének összeállítása (kiírás)
    - tac :utolsó sor legelöl (hasznos pl. log file esetén)
    - head :a file kezdő sorai (default:10)
    - tail :a file záró sorai (default:10)
    - more :szöveg kiírása oldalanként a terminálra
    - less :szöveg kiírása: a felhasználó mászkálhat benne
    - grep :szövegben keresés szabályok szerint regxp – regular expressions
    - (g)awk :szövegben keresés/manipuláció
    - sed :szöveg egyszeri futás alatti szerkesztése
    - vi :klasszik szövegszerkesztő
    - emacs :legendás szövegszerkesztő
180. *Account:*
    - felhasználónév azonosítja,
    - jelszó védi
        - Password file: username:password:uid:gid:gecos:homedir:shelljelszó
    - Két féle account:
        1. Root
        2. User
    - Parancsok:
        - `su`: root átvált az adott userre/ré (paraméter nélkül rootra)
        - `adduser`: felhasználó hozzáadása
        - `passwd` : jelszó megváltoztatása
        - `userdel`: felhasználó törlése
        - `/etc/passwd`: elem törlése
181. *Hozzáférés:*
    - Multi-user környezet!
    - Felhasználók nem férnek hozzá egymás file-jaihoz

- Speciális file-okhoz csak a root fér hozzá
- Csoportok – groups
    - a felhasználók több csoporthoz tartozhatnak
    - könnyebb/rugalmasabb hozzáférés-kezelés
- Hozzáférés: felhasználók / csoportok / egyéb
-

182. *Ownership-változtatás:*
    - Ownership (birtoklás) változtatás: chown pl.: chown username file_or_dir
    - Csoport-birtoklás változtatás: chgrp pl.: chgrp groupname file_or_dir
    - Kombinált csoport és felhasználónév: chgrp name.name file_or_dir
183. *Hozzáférés-változtatás:*
    - chmod parancs
    - r, w, x: read, write, execute
    - Root: megváltoztathatja bármely file/directory hozzáférés-szabályait
    - A root mellett csak a birtokos (user) tudja változtatni