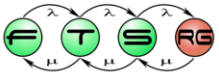
Intelligens rendszerfelügyelet

Mentés, archiválás

Tóth Dániel, Szatmári Zoltán



Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

Utolsó módosítás: 2011. 04. 21.

...


A fájlok megvannak, de ez nem elég, nem indul el. ☹
Nem tudjuk visszatölteni biztonsági mentésről?
Bugsy,
the database server
RIP
2005-2010

Az előző részek tartalmából

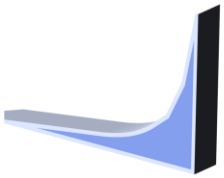
- Szolgáltatásbiztonság az IT rendszerekben
  - Alap technikák
  - *Analízis eszközök (FMEA, hibafa…)*
- Fürtök
  - …
  - *Replikációs módszerek*

Tartalom

- Hibatűrés az adattárolásban
  - Alap technikák
  - Mi mire való
- Mentés (Backup)
- Archiválás
- Adatmegsemmisítés

Hibatűrés az adattárolásban

- Az adat többet ér, mint az adathordozó
  - Az adathordozó pótolható, az adat többnyire nem
- Az adathordozók nem örökéletűek
  - Maguktól is elromolhatnak…
  - Elemi kár is elpusztíthatja
- És persze exponenciálisan nő a tárolandó adatmennyiség…



Jön A Meteor! ☺

RAID

- Már láttuk mérésen… Kérdés:
  - Van-e értelme RAID-nek desktop gépeken?
- Válasz:
  - Attól függ ☺
- El kell dönteni, hogy mit akarunk
  - Nagy teljesítmény?
    - RAID-0
  - Hibatűrés?
    - RAID-(1-6)
  - Biztosan?
  - Mennyi ideig állhat egy desktop gép?
  - Mik a jellegzetes hibamódok?
  - Mennyi az előfordulási valószínűségük? (hasból)
  - Ebből mennyi ellen véd a RAID?
  - Még mindig RAID kell nekünk?

RAID

- Kérdés:
  - Van-e értelme RAID-nek szerver gépeken?
- Válasz:
  - Attól függ ☺, de azért többnyire egyértelmű igen
- Miért?
  - Elsősorban hibatűrési céllal
  - De itt is el kell gondolkodni a kérdéseken: Mennyi ideig állhat a szerver gép?
  - Szerencsére minden IRF hallgató tudja, hogy ez egy rosszul feltett kérdés! Helyesen: mennyi ideig eshet ki a szolgáltatás?
  - Mik a jellegzetes hibamódok?
  - Mennyi az előfordulási valószínűségük? (hasból)
  - Ebből mennyi ellen véd a RAID?
  - Kell a RAID mellé más is nekünk?

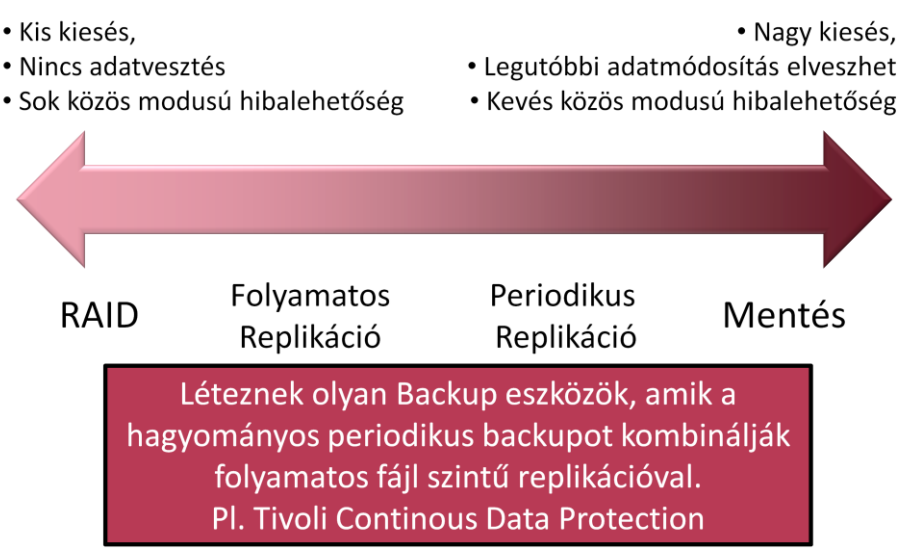
Hibatűrés az adattárolásban

Technika	Ez ellen véd	Az ellen nem véd	Garancia
RAID (1-6)	Adathordozó meghibásodás	Minden más (tápegység, elemi kár, vezérlő hiba, OS hiba, emberi hiba, alkalmazás hiba)	Folyamatos üzem meghibásodás esetén is
Replikáció (pl. DRBD)	Hardver többféle hibája, hálózati hiba	Emberi hiba, OS hiba, alkalmazás hiba	Típusfüggő, akár folyamatos üzem
Mentés	Mindenféle hardverhiba, akár elemi kár, emberi hiba is	Nagy emberi hiba	Nincs folyamatos üzemelés, visszaállítás kell!
Archiválás	?	?	?

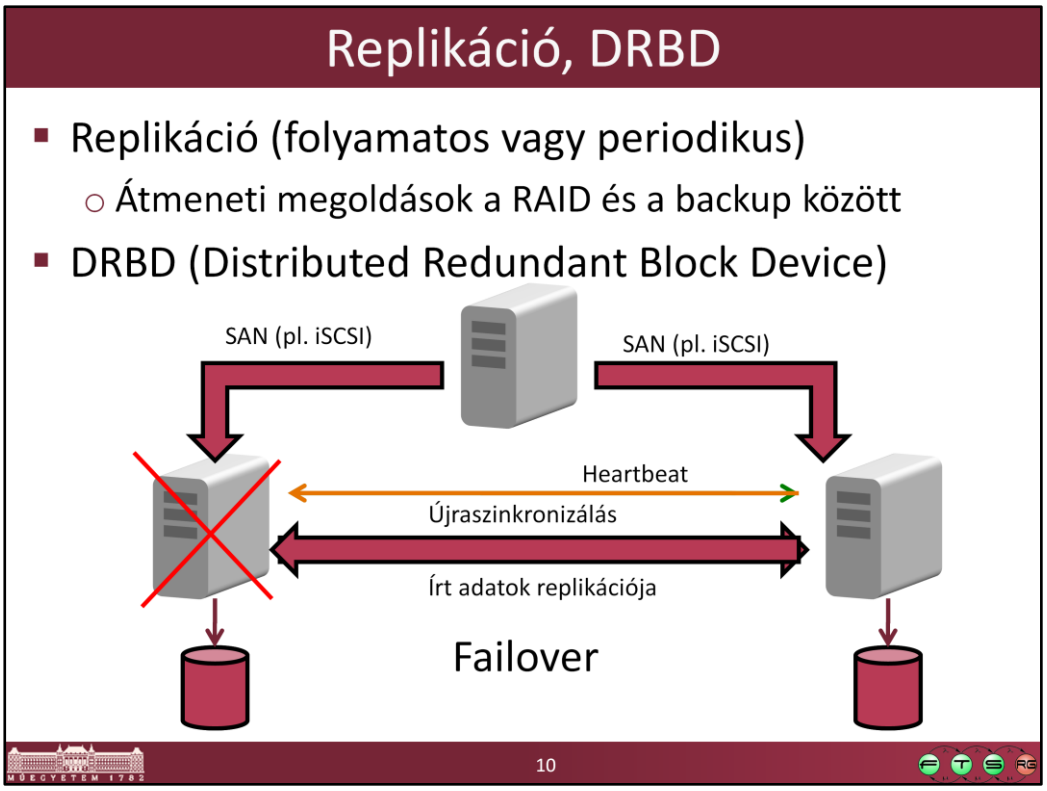
„Csalás!” ☺
Az archiválás NEM hibatűrési mechanizmus, nem összekeverendő a backuppal!
Az archiválás célja a már használaton kívüli, de megőrzendő adatok biztonságos tárolása

Gyors áttekintés

- Adattár hibatűrési technikák áttekintése:



Léteznek olyan Backup eszközök, amik a hagyományos periodikus backupot kombinálják folyamatos fájl szintű replikációval.
Pl. Tivoli Continous Data Protection



http://www.drbd.org
A DRBD véd sok közös módusú hiba ellen (hw, elemi kár), mert akár földrajzilag távol is lehetnek a csomópontok. Nem véd emberi, OS, alkalmazás hibája, illetve szándékos rongálás ellen (nem is tudja detektálni ezeket a hibákat, csak blokkos eszközt biztosít és replikálni fogja a hibás adatot is, ha ilyet kap). Hiba utáni helyreállásnál újraszinkronizál, a meghibásodás óta összegyűlt változásokat replikálja (és a gyakorlatban gyakran ez nem szokott működni…). Van periodikus replikációs üzemmód is, amikor nem folyamatosan, csak meghatározott időnként történik a szinkronban tartás. Az utóbbi a legutóbb módosult adatok elvesztésével jár, de korlátozottan védhet magasabb szintű hibák ellen is.

Biztonsági másolat készítése

- Mentés
  - Rendszeresen másolatot készítünk az adatokról, lehetőleg olyan médiumra, amit kevés az elsődleges rendszerével közös modusú hiba érint
  - A másolatokból visszamenőleg több eltérő időben készült példányt is őrizhetünk (ez véd akár emberi hibák ellen is!)
- Jellegzetes kérdések
  - Mire készítsük a mentést?
  - Milyen rendszeresen?
  - Miről készítsük a mentést?
  - Milyen módon? (kézzel, automatizáltan?)
  - Mennyi ideig őrizzük meg?

Mivel az IRÜ-ben elég részletesen ki van vesézve a mire készítsük a mentést kérdése, ezért ezt itt külön nem tárgyaljuk, legfeljebb szóban (szalag olcsó, főleg nagy méretben; optikai adattárolás már nem annyira, de még kis környezetben elmegy; merevlemez már meglepően olcsó, kis méretben akár szóba is jöhet biztonsági másolat tárolására, főleg a „green” változatok)

Biztonsági másolat készítése

- Milyen rendszeresen…
  - Ha sűrűn mentünk
    - Nagy terhelés a rendszernek
    - Sok mentett adat keletkezik
  - Ha ritkán mentünk
    - A mentések között túl sok védelem nélküli adat gyűlik össze
  - Mit tehetünk a hátrányok ellen?
    - Verziókezelés, csak a különbségek tárolása
    - Adat *deduplikáció*.

Biztonsági mentések historikus megőrzése

- Nem biztos, hogy az utolsó mentés helyes, vagy tartalmazza az elveszett adatot
  - Célszerű több mentést tárolni visszamenőleg
- Újra fontos kérdések merülnek fel:
  - Milyen gyakran mentsünk?
  - Mennyi idő múlva dobhatunk el egy mentést?

A mentés paraméterei tudatos rendszertervezés során alakulnak ki és a mentendő rendszerre jellemzőek!
„Backup policy”

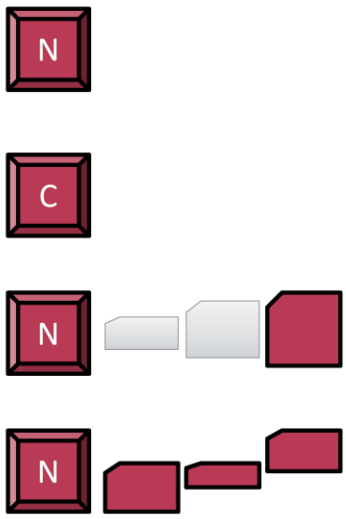
Különbségi mentés

- Példa: Windows Backup Service
- Kulcselem: „archive” attribútum bit a fájlokon
  (igen, ez félrevezető elnevezés, újabb verziókban már „to be backed up” flag)
- Backup típusok:
  - **Normal** – minden fájlt ment, törli az archive bitet
  - **Copy** – minden fájlt ment, nem törli az archive-ot
  - **Differential** – csak az archive bittel jelölt fájlokat menti, nem törli az archive bitet róluk
  - **Incremental** – csak az archive bittel jelölt fájlokat, törli az archive bitet

Különbségi mentés

- Helyreállítási alapesetek:

Normal	• Legutóbbi normal
Copy	• Legutóbbi copy
Differential	• Legutóbbi normal • + legutóbbi differential
Incremental	• Legutóbbi normal • + azóta összes incremental



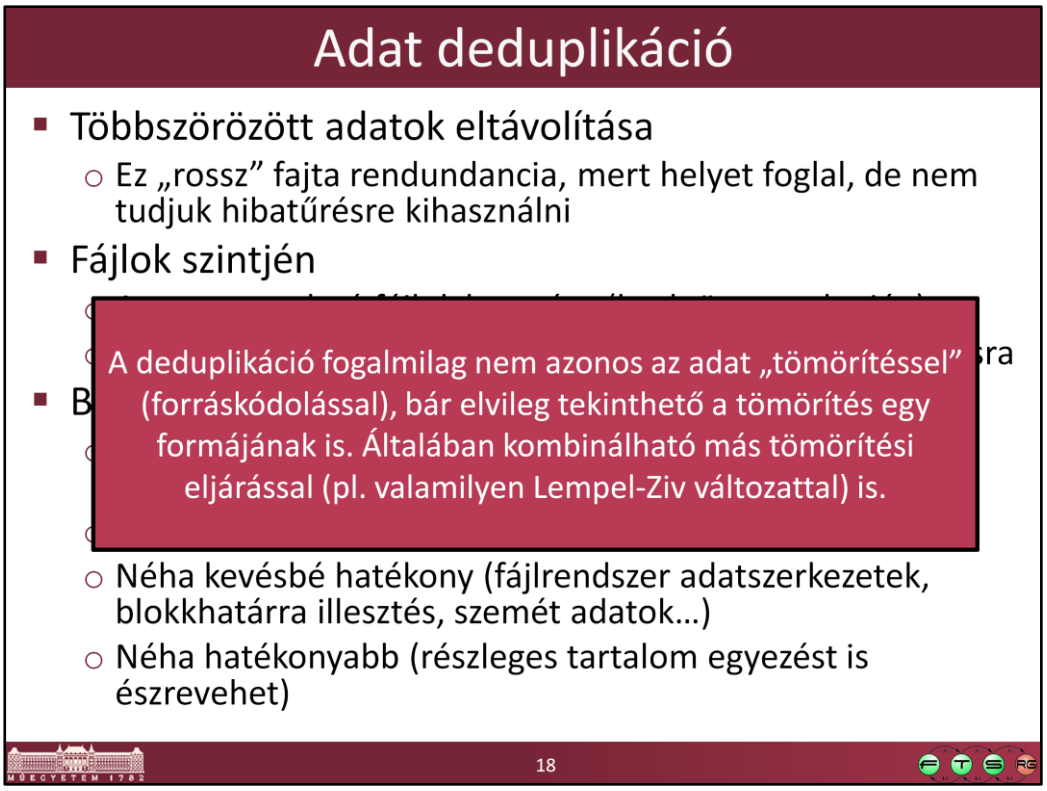
Jobb oldalon kiemelve: mi kell a teljes rendszer legutóbb mentett állapotának helyreállításához.

DEMO Windows Backup Service

- Különbségi mentés lehetőségek kiválasztása
- Volume Shadow copy service (mire jó?)

Adat deduplikáció

- Többszörözött adatok eltávolítása
  - Ez „rossz” fajta rendundancia, mert helyet foglal, de nem tudjuk hibatűrésre kihasználni
- Fájlok szintjén
  - Azonos tartalmú fájlok keresése (hash összeg alapján)
  - Ismétlődő fájlok lecserélésre az első példányra hivatkozásra
- Blokkos eszköz szintjén
  - Blokkok, vagy nagyobb allokációs egységek szintjén hash összeg alapján
  - Újabb SAN eszközökben már hardveres támogatással
  - Néha kevésbé hatékony (fájlrendszer adatszerkezetek, blokkhatárra illesztés, szemét adatok…)
  - Néha hatékonyabb (részleges tartalom egyezést is észrevehet)



Blokkos eszköz szintjén az teszi lehetővé, hogy pl. egy LVM rendszerben van egy B-fa alapú indirekciós réteg a fizikai és logikai címtartományok között, amiben (csak ebben az esetben kivételesen) megengedett, hogy egy logikai blokkhoz több fizikai blokk tartozzon.

Tömörítés: érdemes megmelíteni az lrzip (long range zip) projektet, ami deduplikációs előfeldolgozóval egészít ki lzma, bzip2 és hasonló elterjedt tömörítőket.

DEMO Rsync / Dirvish, rSnapshot

- Egy elemi eszköz és erre épülő technológiák
- Deduplikáció fájlok szintjén
  - Unix VFS hard link funkcióra épül (a változatlan fájlok csak hard linkek az eredeti példányra)
  - A fájlrendszer megoldja a szemétszedést (akkor törlődik az adat, ha a hard link referencia számláló 0-ra fut)

Mit szeretnénk menteni?

- Teljes operációs rendszert
- Partíciót / Fájlrendszert
- Kizárólag fájlokat
- Speciális szolgáltatások adatait, beállításait

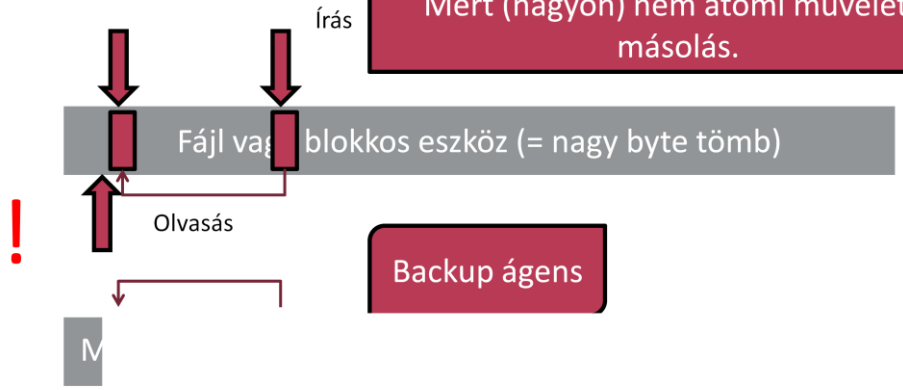
Hogyan készítsünk mentést?

- Fájlrendszerről
  - Egyszerű… NOT!
  - Lockolt fájlokkal gond lehet (főleg windows alatt)
  - Ráadásul az alkalmazásnak jó oka lehet rá, hogy lockot tart a fájlon
  - A fő kérdés: ha valamilyen átmeneti, módosítás közbeni állapotot állítunk vissza, akkor az alkalmazás képes lesz-e abból helyreállni?
  - -> Alkalmazás specifikus backup lehetőségek is kellenek

Hogyan készítsünk mentést?

- Blokkos eszközről
  - Ugyanaz a problém

A már átmásolt részen történő módosítás már nem kerül mentésre. Inkonzisztenssé válik a másolt példány!
Miért?
Mert (nagyon) nem atomi művelet a másolás.



Segít itt a naplózás? Nem!

Megoldás

- Pillanatkép (Snapshot) funkció
  - Logikai kötetkezelők támogatják
  - Néhány fájlrendszer is (pl. ZFS)
- A snapshot készítés atomi művelet
  - Másolás csak menet közben történik (copy on write)

Snapshot != mentés
Nem készül biztonsági másolat! A snapshot egy támogató technológia konzisztens, atomi másolat létrehozásához.

A snapshot egy olyan blokkos eszköznek vagy fájlnak látszik, amiben változatlanul maradt minden. Valójában csak azoknak a blokkoknak az eredetijét tárolja, amik felül lettek írva az eredeti eszközön, a változatlan blokkokat közvetlenül az eredeti példányról olvassa.
Elég ez a konzisztens backuphoz, amiből az alkalmazás fel tud állni? Nem. Csak akkor elég, ha az alkalmazás belül még naplózott adatstruktúrát is használ.

Alkalmazás-specifikus mentés

- Példa1 MS-SQL szerver beépített backup funkciója
  - `BACKUP DATABASE yourdb TO DISK='C:\temp\yourdb.bak' WITH INIT`
  - `RESTORE DATABASE yourdb FROM DISK='C:\temp\yourdb.bak'`
- Példa2 OpenLDAP
  - slapcat, ldif formátumban dumpolja a teljes adatbázis tartalmat)
- Példa3 MySQL
  - mysqldump
- …

DEMO Adatbázis mentés bash környezetben

```
#!/bin/bash
for db in `mysql -h host -u user -p pass
                        -e  'show databases' `;
do
mysqldump -h host -u user –p pass
   --add-drop-database
   --add-drop-table
   --create-options $db | gzip > $db.sql.gz
done
```

Ki kezdeményez?

- Push típusú mentés
  - A mentendő rendszer kezdeményezi a biztonsági mentést
  - Mindenhova „intelligenciát” kell telepíteni
- Pull típusú mentés
  - Központi rendszer kezdeményez
  - Hozzáférést kell biztosítani a mentendő rendszer fájlaihoz
- Vegyes megoldás

Biztonsági mentés teljesítménye

- Mivel mérhető a biztonsági mentés teljesítménye?
  - Egyszeri mentés lefutási ideje
  - Visszaállítás időigénye (Nem kritikus, ha ritkán van rá szükség)
  - Visszaállítható időtáv
  - Visszaállítható verziók száma
  - Elfoglalt tárhely aránya az adatmennyiséghez képest

Esettanulmány

- Komplex, vegyes komponensekből felépülő rendszer mentésének megtervezése
  - Windows és Linux csomópontok vegyesen
  - Speciális szolgáltatások

Összefoglalás

- Feltételek konzisztens mentés készítéséhez egy működő rendszerről:
  - A futó alkalmazás vagy fájlrendszer belül naplózó adatszerkezetet használjon
    (ez erősen alapkövetelmény, nélküle egy egyszerű „kemény” leállítást sem élne túl!)
  - Az adatról a másolat atomi műveletként készüljön, akár snapshot funkció segítségével.

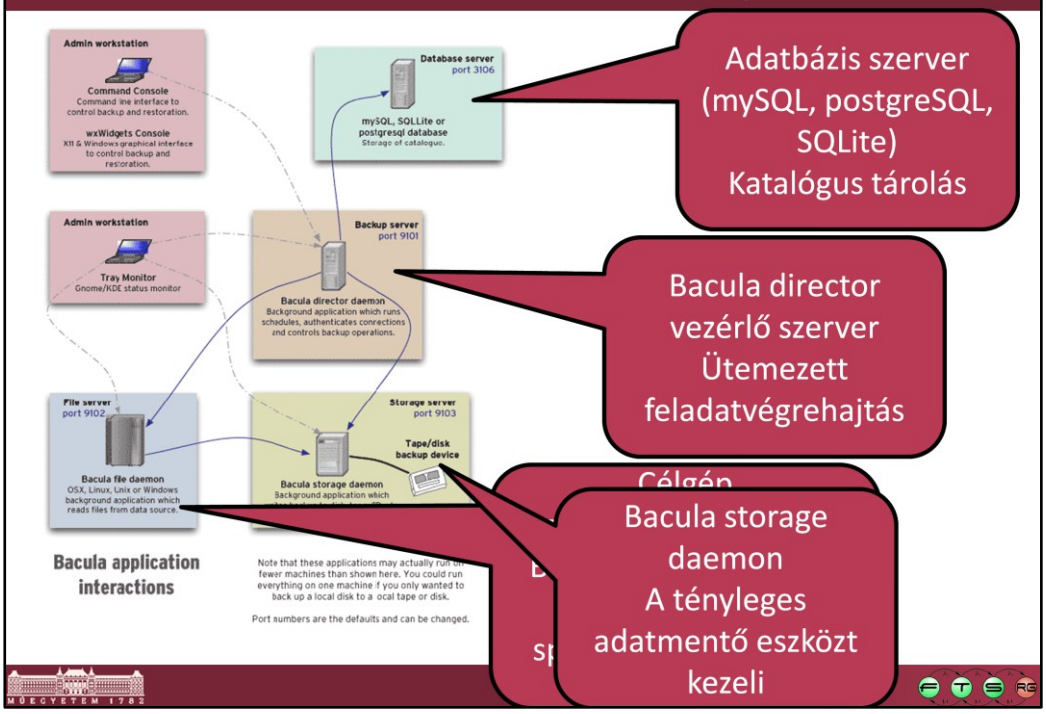
Archiválás

- Még egyszer: **nem mentési technológia**
- Nagy mennyiségű, ritkán szükséges adat **„másik” helyen való tárolása**
- Mentéssel ellentétben itt **egy példány**ban létezik az adat továbbra is

Adat katalogizálás

- Főleg az archiválás, kisebb részben a Backup problémája
- Hova mentettünk mit?
  - A NASA pl. az Apollo 11 eredeti rádióforgalmat rögzítő mágnesszalagjairól nem tudja, hogy hol vannak
- Egyszerű megoldás: Fájlnév katalógus
- Bonyolultabb megoldások: metaadatok, tartalom alapján keresés

Példa: Bacula backup



Adatmegsemmisítés

- Egy kis disztroj ☺
- Adatvédelmi előírások miatt szükséges lehet, hogy egy adatból garantáltan ne maradjon fenn példány
- Fájlok törlésekor nem törlődik az adat ténylegesen
  - Fájlrendszer elhagy szemetet
  - Biztonsági másolatok is megmaradnak
- Tudni kell, hogy hol vannak backup/archív példányok belőle (katalógus)
- Felül kell írni az fizikai eszközön a helyét
  - 0-kkal?
  - Elég egyszer?

Bizonyos esetekben a 0-kkal felülírás kevés (az adathordozóról „analóg” módszerekkel még helyreállítható valami gyenge jel), ilyenkor véletlen adatokkal írják felül, több menetben is. Ilyen adatmegsemmisítést főleg katonai vagy egyéb állami törvények írhatnak elő.

Összefoglalás

- Hibatűrés az adattárolásban
  - Alap technikák
  - Mi mire való
- Mentés
  - Konzisztencia biztosítása
  - Deduplikáció
  - Katalogizálás
- Archiválás
- Adatmegsemmisítés