

12. előadás

Állományok

Egyszerű állományok

Adatszerkezetek és algoritmusok előadás
2018. május 8.

Kósa Márk, Pánovics János és Szathmáry László
Debreceni Egyetem
Informatikai Kar

Állományok

Kósa Márk
Pánovics János
Szathmáry László



Alapfogalmak

Állománykezelési
műveletek

Állományszervezési
módok

Egyszerű
állományszerkezetek

Szeriális állomány
Szekvenciális állomány
Direkt állomány
Random állomány



Másodlagos tároló (külső tároló, háttértár, periféria tároló):

- tárolóeszköz, amelyen adatokat és programokat tárolunk
- adatcsatornán keresztül kapcsolódik a központi feldolgozóegységhez
- a rajta tárolt adatok közvetlenül nem használhatók fel, csak a belső tárba való bejuttatásuk után

Azért van rá szükség, mert a belső táruk kapacitása viszonylag kicsi, és csak időleges adat- és programtárolásra szolgál.

A modellezés során először az állomány absztrakt szerkezetét, a **logikai állományt** alakítjuk ki, majd ezt képezzük le a háttértárra, s így kapjuk a **fizikai állományt**.

Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszervezetek

Szeriális állomány

Szekvenciális állomány

Direkt állomány

Random állomány

Alapfogalmak

- **Adattétel, adatelem**: a legkisebb, a felhasználó szempontjából még önálló jelentéssel bíró adat. Jellemzői: név, típus és hossz.
- **Adatcsoport**: adattételek logikailag egymáshoz kapcsolódó együttese. Típusai: vektor típusú, ismétlődő, összetett.
- **Logikai rekord**: adattételek és adatcsoportok rögzített sorrendű, logikailag összetartozó együttese.
- **Logikai rekordazonosító, elsődleges kulcs**: az az adatelem vagy adatcsoport, amely minden logikai rekordban különböző értéket vesz fel.
- **Másodlagos kulcs**: az az adatelem vagy adatcsoport, amelynek értékei nem feltétlenül egyediek az egyes rekordokban, de amelyek a rekordok feldolgozásának alapjául szolgálnak.
- **Logikai fájl, logikai (adat)állomány**: valamilyen feldolgozási cél, tartalom vagy forma szerint összetartozó logikai rekordok névvel ellátott együttese.



Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszerkezetek

- Szeriális állomány
- Szekvenciális állomány
- Direkt állomány
- Random állomány

Rekordformátumok

- **fix**: a logikai rekordok azonos szerkezetűek és hosszúságúak
- **változó**: a logikai rekordok szerkezete azonos, hosszuk azonban eltérhet
- **határozatlan**: a logikai rekordok szerkezete is különbözhet

A logikai állományok szerkezete lehet:

- **struktúra nélküli**: a logikai rekordok sorrendje tetszőleges
- **asszociatív**: a logikai rekordok valamilyen egyértelmű, diszjunkt csoportokba sorolhatók
- **szekvenciális**: a logikai rekordok között valamilyen szempont szerint sorrendiséget (rendezettséget) értelmezünk
- **hierarchikus**: a logikai rekordok hierarchiát alkotnak, minden rekordnak legfeljebb egy megelőzője és tetszőleges számú rákövetkezője lehet



Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszervezetek

Szeriális állomány

Szekvenciális állomány

Direkt állomány

Random állomány

Alapfogalmak

- **Mező:** a fizikai állomány legkisebb alkotórésze, a logikai adattételnek (adatelemnek) tartalmilag megfelelő fizikai tárolási egység. Jellemzői: név, típus, hossz (bájtban) és ábrázolási mód.
- **Mezőcsoport:** a logikai adatszoport fizikai megfelelője.
- **Fizikai rekord, blokk:** a tárolt adatok olyan egysége, amelyet a számítógép egyetlen író-olvasó utasítással képes kiírni egy tárolóterületre vagy beolvasni egy tárolóterületről. Kapcsolata a logikai rekorddal vagy rekordokkal:
 - egy fizikai rekord egy logikai rekordot tartalmaz
 - egy fizikai rekord több logikai rekordot tartalmaz (**blokkolás**)
 - egy logikai rekord csak több fizikai rekordban fér el (**szegmentálás**)
- **Fizikai állomány:** egy (vagy több) logikai állomány logikai rekordjait tartalmazó fizikai rekordok (blokkok) önálló névvel ellátott sorozata.



Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszervezetek

Szeriális állomány

Szekvenciális állomány

Direkt állomány

Random állomány



1. *Fix rekordformátum:*

Az összes rekord hossza azonos: állományra jellemző információ.

- *Nincs sem blokkolás, sem szegmentálás:* Nincs szükség további információra.
- *Blokkolás:* Szükséges egy plusz információ arról, hogy hány logikai rekord kerül egy blokkba (**blokkolási tényező**), ami ugyancsak az állományra jellemző tulajdonság.
- *Szegmentálás:* Szükséges egy plusz információ arról, hogy hány blokkra tördeljük a logikai rekordokat, ami szintén az állományra jellemző tulajdonság.

Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszerkezetek

Szeriális állomány

Szekvenciális állomány

Direkt állomány

Random állomány



2. Változó rekordformátum:

A rekordok mérete különbözhet.

- *Nincs sem blokkolás, sem szegmentálás:* Rögzített blokkméret esetén a blokk méretét akkorának kell választani, hogy a leghosszabb rekord is beleférjen. Ebben az esetben a blokk ki nem használt része lehet „szemét”, vagy feltölthetjük egy speciális értékkel. Ha a blokkméret nem rögzített, akkor a blokk mérete megegyezhet a logikai rekord méretével, és ezt a méretet minden blokk elején tárolni kell. Mindkét megoldás esetén tárolnunk kell a változó hosszúságú adatelemek hosszát, illetve ismétlődő adatcsoportok esetén az adatelemek számát. Ezeket vagy a kérdéses adatelemek/adatcsoportok előtt, vagy a blokk elején, a **blokkfejlécben** tároljuk.

Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszerkezetek

Szeriális állomány

Szekvenciális állomány

Direkt állomány

Random állomány

2. Változó rekordformátum:

- **Blokkolás:** Minden logikai rekord előtt tárolni kell a rekord méretét. Rögzített blokkméret esetén az érkező rekordokat sorban pakoljuk a blokkba, és ha már nem fér több, akkor a következő blokkban folytatjuk. Az előző blokk ki nem használt része vagy „szemét” marad, vagy feltöltjük egy speciális értékkel. Ha a blokkméret nem rögzített, akkor megadunk egy maximális blokkméretet, és ha az érkező rekord nem fér el az aktuális blokkban, akkor új blokkot kezdünk. Ilyenkor nincs feleslegesen lefoglalt terület, viszont tárolni kell a blokkok elején azok hosszát.
- **Szegmentálás:** A blokkoláshoz hasonlóan sok plusz információ tárolásával kell dolgozni.

3. Határozatlan rekordformátum:

Minden egyes rekordban meg kell adni, hogy mely adatalemeket és adatcsoportokat tároljuk, vagy azt, hogy melyeket nem. Általában nincs sem blokkolás, sem szegmentálás, mert körülményes azok megvalósítása.



A műveletek alapja a logikai rekord.

- 1 Létrehozás
- 2 Bővítés
- 3 Törlés
 - logikai
 - fizikai
- 4 Csere
- 5 Elérés
 - soros
 - szekvenciális
 - közvetlen
- 6 Keresés
- 7 Rendezés
- 8 Feldolgozás
- 9 Újrászervezés



- Szeriális állomány
- Szekvenciális állomány
- Direkt állomány
- Random állomány



Egyszerű állományszerkezet

A fizikai állomány csak a logikai állomány adatait tartalmazza.

Összetett állományszerkezet

A fizikai állomány a logikai adataikon kívül szerkezhordozó adatokat is tartalmaz.

Alapfogalmak

Állománykezelési
műveletek

Állományszervezési
módok

Egyszerű
állományszerkezetek

- Szeriális állomány
- Szekvenciális állomány
- Direkt állomány
- Random állomány

Egyszerű állományszerkezetek osztályozási szempontjai

Egy egyszerű állomány fizikai szerkezetét azok a szabályok határozzák meg, amelyek szerint a logikai rekordok elhelyezkednek a fizikai állományban.

A szabályok két dologra vonatkoznak:

- a logikai adatszerkezet elemei közötti kapcsolatokra,
- a logikai rekordazonosító és a logikai rekord által elfoglalt fizikai tárolóterület címe közötti kapcsolatra.

Ezek alapján az **egyszerű állományszerkezetek** négy alapvető fajtája:

kapcsolat állomány- szerkezet	a logikai rekordazonosítók között	a rekordazonosítók és tárolási címeik között
szériális	nincs	nincs
szekvenciális	van	nincs
direkt	van	van
random	nincs	van



Szeriális állomány

Egy **szeriális** állományban sem a rekordazonosítók, sem pedig a rekordazonosítók és tárolási címeik között nincs összefüggés (kapcsolat). Gyakran még rekordazonosítók sincsenek.

Szeriális állománnyal végezhető műveletek

- **Létrehozás**: üres tevékenység, nincs szerkezet.
- **Bővítés**: az utolsó rekord után.
- **Törlés**: alapvetően logikai, fizikai törlés csak speciális esetben.
- **Csere**: csak speciális esetben realizálható, általában újraszervezést igényel.
- **Elérés**: soros.
- **Keresés**: teljes keresés.
- **Rendezés**: nincs.
- **Feldolgozás**: a rekordok fizikai sorrendje szerint.
- **Újraszervezés**: csere esetén, illetve ha a logikailag törölt rekordokat fizikailag is törölni szeretnénk.



Szekvenciális állomány

A **szekvenciális** állományban a logikai rekordok az elsődleges kulcs értéke szerint növekvő vagy csökkenő sorrendben helyezkednek el, ezáltal a rekordok fizikai és logikai sorrendje megegyezik.

Szekvenciális állománnyal végezhető műveletek

- **Létrehozás:** egy lépésben (ha a rekordok érkezési sorrendje az elsődleges kulcs alapján rendezett) vagy két lépésben (egy szeriális állomány rendezésével).
- **Bővítés:** csak újraszervezéssel.
- **Törlés:** alapvetően logikai, fizikai törlés csak újraszervezéssel.
- **Csere:** csak speciális esetben realizálható, általában újraszervezést igényel.
- **Elérés:** a soros és a szekvenciális elérés most egybeesik.
- **Keresés:** lineáris, speciális esetben bináris.
- **Rendezés:** nincs értelme.
- **Feldolgozás:** a keresés alapján. Lineáris keresést használva egyszerre több rekord is feldolgozható, ha a keresett rekordokat is rendezzük az elsődleges kulcsuk szerint.
- **Újraszervezés:** bővítés és csere esetén, illetve ha a logikailag törölt rekordokat fizikailag is törölni szeretnénk.



A direkt és random állományok esetén egy **hash függvényt** használunk az egyes rekordok háttértáron elfoglalt helyének meghatározására, amely a logikai rekordazonosítóhoz (az elsődleges kulcshoz) rendel egy lemezcímet. Ha ez a hash függvény kölcsönösen egyértelmű, akkor **direkt** állományról, ha csak egyértelmű, **random** állományról beszélünk.

Ha a gyakorlatban előforduló rekordazonosítók nem egyenletes eloszlást követnek, akkor a kölcsönösen egyértelmű megfeleltetés érdekében helyet foglalhatunk néhány olyan rekord számára is, amelyek valójában sohasem fognak előfordulni az állományban. Ennek következtében a tárolóterületen felhasználatlan tárolási egységek alakulnak ki.

A kölcsönösen egyértelmű hash függvény a logikai rekordazonosítók növekvő sorrendjét kialakítva állítja elő az $1., 2., \dots, n.$ címeket. A hash függvénynek e tulajdonsága teszi lehetővé, hogy a direkt állományokat szekvenciálisan is feldolgozhassuk.



Direkt állomány

A direkt állomány fizikailag egyetlen területből, az **adatterületből** áll, és csak címezhető háttértárolón hozható létre.

Direkt állománnyal végezhető műveletek

- **Létrehozás:** két lépésben. Az első lépésben meghatározzuk a kezelendő rekordok számát és a hash függvényt, majd kialakítjuk az állomány vázát; a második lépésben feltöltjük a vázat rekordokkal. A létrehozás után maradhatnak üres rekordhelyek (**álrekordok**).
- **Bővítés:** szűkebb értelemben nincs. Tágabb értelemben az új rekordokat a hash függvény által meghatározott címre helyezzük.
- **Törlés:** csak logikai (törlőbájtal vagy a kulcs cseréjével).
- **Csere:** a logikai rekordazonosítón kívül bármelyik adatelem értéke cserélhető.
- **Elérés:** közvetlen, szekvenciális (amely egybeesik a sorossal) és szakaszos szekvenciális.
- **Keresés, rendezés:** nem értelmezett.
- **Feldolgozás:** az elérés alapján, legtöbbször közvetlenül.
- **Újraszervezés:** csak akkor merülhet fel, ha változik az elméletileg számba vehető rekordazonosítók tartománya.





A túlcsoordult rekordok elhelyezése történhet:

- az elsődleges adatterületen
- független túlcsoordulási területen

Szinonimakezelési technikák

- nyílt címzés
- láncolás
- szabad helyek nyilvántartása
- Vyssotzky-módszer
- független túlcsoordulási terület használata
- osztott szabad helyek módszere

Alapfogalmak

Állománykezelési műveletek

Állományszervezési módok

Egyszerű állományszerkezetek

Szeriális állomány

Szekvenciális állomány

Direkt állomány

Random állomány

Random állománnyal végezhető műveletek

- **Létrehozás:** két lépésben. Az első lépésben meghatározzuk a kezelendő rekordok számát, a hash függvényt és a szinonimakezelési technikát, majd kialakítjuk az állomány vázát; a második lépésben feltöltjük a vázat rekordokkal. Ha a túlcsondult rekordokat az elsődleges adatterületen helyezzük el, akkor egyrészt célszerű kb. 10%-kal több helyet foglalni, mint a gyakorlati lehetőségek száma, másrészt célszerű először a nem túlcsondult rekordokat elhelyezni az állományban. A létrehozás után maradhatnak üres rekordhelyek (**álrekordok**).
- **Bővítés:** ha a hash függvény által meghatározott tárterület már foglalt, akkor a szinonimakezelési technika által előírt helyre helyezzük a rekordot.
- **Törlés:** csak logikai (törlőbájtal vagy a kulcs cseréjével).
- **Csere:** a logikai rekordazonosítón kívül bármelyik adatelem értéke cserélhető. Ha a rekord új értékének mérete meghaladja az eredeti értékének méretét, akkor a rekordot túlcsondult rekordként kell kezelni, tehát gyakorlatilag egy törlést és egy bővítést kell végrehajtani.
- **Elérés:** közvetlen és cím szerinti szekvenciális.
- **Keresés, rendezés:** nem értelmezett.
- **Feldolgozás:** az elérés alapján.
- **Újraszervezés:** gyakran felmerül, ha nagy a rekordok fluktuációja (sok a törlés és a bővítés), mivel ilyenkor nő a túlcsondult rekordok száma.

