**1.**

**A. Ön egy olyan vállalat informatikusa, amely több-felhasználós operációs rendszerre fejleszt**​  **alkalmazásokat. Az alkalmazás memóriaigénye miatt Önnek át kell alakítani a fizikai memória kiosztását. Ismertesse, hogy milyen memória definíciós eljárásokat ismer, és magyarázza el a statikus és dinamikus memória allokáció közötti különbségeket!**

**Információtartalom vázlata:**

– A memória definiálás lehetséges módszerei

– A memória allokálás technikái (bit térkép, memória ellenőrző blokk)

– Allokációs stratégiák dinamikus allokációhoz

– A virtuális memória-kezelés elve.

– A virtuális memória-kezelés alapproblémái (Betöltési, Elhelyezési, Helyettesítési és Visszatöltési problémák)

# **A memória definiálás lehetséges módszerei:**

A fizikai memória felosztásának két módja van (az OR hajtja végre):

* Statikus memória definíció (SRAM): Az allokálható blokkok méretét és számát az OR generálásakor rögzítjük. A blokkok száma soha nem változik (partíciók).
* Dinamikus memória definíció (DRAM): Az OR a blokkok számát és méretét a futásidőben, egy igény alapján határozza meg. (Régiók, szétdaraboltság problémája.)

# **A memória allokálás technikái (bit térkép, memória ellenőrző blokk):**

Attól függően, hogy egy feldolgozás alatt álló folyamat futás közben foglalhat-e le újabb területeket, két stratégia lehetséges:

* Statikus allokáció (a program betöltéskor, indításakor)
* Dinamikus allokáció (futás közben is lehetőség újabb területek lefoglalására, ha szükséges)

A dinamikus allokáció hatékonyságát két tényező befolyásolhatja:

* a jól megválasztott adatstruktúra kevesebb helyigényű és gyorsabb adminisztrációt eredményez:
  + bit térképet akkor használunk, ha a memória egyenlő méretű blokkokra van osztva.

Egy blokkhoz egy bitet rendelünk a foglaltság jelzésére.

* + Memória ellenőrző blokkokkal változó hosszúságú blokkokat kezelünk. Ez lehetőséget ad több memóriablokk összekapcsolására is. (Mindig tartalmazza a típust, a hosszat és a pointert.) Két típus:

-FMCB = Free Memory Control Block

-AMCB = Allocated Memory Control Block.

* az allokálást hatékonyan végző allokációs algoritmus jobb helykihasználást eredményez

# **Allokációs stratégiák dinamikus allokációhoz**:

Első Illesztés (First Fit): Az első FMCB-től indulva megvizsgál minden szabad blokkot és az első olyan blokkot választja, amely megfelel az igényelt méretnek. Gyors, de egy idő után összegyűlnek a kicsi „maradvány” blokkok, amikkel már nem tudunk mit kezdeni.

Következő Illesztés (Next Fit): Az előző egyszerűsített változata, csak itt új keresést mindig ott kezdünk az FMCB blokkok között, ahol az előzőt abbahagytuk. Egy kicsit rosszabbul viselkedik, mint az FF.

Legjobb illesztés (Best Fit): Az előző kettő nem veszi figyelembe a blokk tényleges méretét. Ez az algoritmus bejárja az összes blokkot, megjegyzi azt, amelyikhez a legjobban illeszkedik az igény. (azaz a legkevesebb felhasználatlan terület marad)

Legrosszabb illesztés (Worst Fit): Megfordítja az előző szabályt. Ha nagyobb szabad hely marad, nagyobb az esély, hogy később fel tudjuk használni.

**A virtuális memória kezelés elve:**

A virtuális memóriakezelés azt az illúziót nyújtja a felhasználónak, mintha a virtuálisan címezhető terület mindig a rendelkezésre állna fizikailag is.

Ezért a programokban specifikált virtuális címeket az OR a hivatkozás pillanatában át tudja alakítani olyan fizikai címekké, amelyek a főtárban rendelkezésre állnak.

Így a virtuális címeket a fordítás során úgy tudjuk megválasztani, ahogy az a fordítónak kényelmes, a fizikai címek pedig mindig az OR-hez illeszkednek.

A módszer előnyei:

● A virtuális és a fizikai címek elkülönülnek egymástól: bárhogyan változnak meg a fizikai címezhetőség határai, a virtuális címek nem változnak.

● A memóriavédelem könnyebben szervezhető.

# **A virtuális memória-kezelés alapproblémái (Betöltési, Elhelyezési, Helyettesítési és Visszatöltési problémák): é......p**

oNégy alap probléma

* Betöltés

o Mikor kell egy lapot betölteni?

* Elhelyezés

o Hova kell a lap képét betölteni?

* Helyettesítés

o Cél: olyan lapot távolítsunk el, amelyet nem kell visszatölteni (feltehetőleg) stratégiák:

▪ legrégebben használt

▪ legritkábban használt

▪ legrégebben memóriában lévő (FIFO)

▪ kombinált módszerek

* Módosított lapok kezelése

o nem módosított lapok

o módosított lapok

▪ Időigényesebb a lap cseréje! Ezért csak akkor választunk ebből a halmazból, ha a másik üres

„Az aktuális - és az ezek közvetlen közelében elhelyezkedő – címekre a közeljövőben nagy valószínűséggel ismét hivatkozunk.” – Lokalitás elve: Bélády László – 1966

**B. Ön részt vesz programozóként egy üzleti portál fejlesztésében. A portál egyik szolgáltatása az**​  **online értékesítés, amelyhez egy adatbázist kell kialakítani. A részletesen meghatározott követelmények és funkcionális elvárások alapján el kell készítenie a logikai adatmodellt. Fejtse ki az adatbázisok tervezésének módszereit és eszközeit!**

**Információtartalom vázlata:**

– Adatbázis-kezelés alapfogalmai

– Adatbázis kezelő rendszerek feladatai

– Adatmodell elemei és jellemzőik

– Egyed-Kapcsolat diagram

**Adatbázis-kezelés alapfogalmai:**

* **Adat:** értelmezhető, de nem értelmezett ismeret (észlelhető, érzékelhető)
* **Információ:** értelmezett. “hasznos” adat pl: a kreditérték (valakinek adat, másnak információ)

Példa: egy kurzus kreditértéke 2, aki tudja, mi ez, annak információ, aki nem annak csak adat

* **Meta adat:** adat0,
* okhoz tartozó adatok, jellemzően a strukturált tárolás, értelmezéshez szükségesek, Példa: az egyes adatok azonosítására szolgáló név
* **Adatbázis:** logikailag összetartozó adatok
* **Adatbázis-rendszer:** az adatbázist, a hozzá tartozó egyéb adatokat, sokszor a kezelő informatikai rendszert (hardver, szoftver) együtt jelentő kifejezés
* **Séma:** az adatbázis szerkezetét, az összefüggéseket írja le
* **Konzisztencia:** az adatok közötti összefüggések helyessége, pl. egy adat többszöri előfordulásakor ugyanúgy legyen leírva
* **Adatintegritás:** az adatok helyesek, konzisztensek
* **Hivatkozási integritás:** csak az adatbázisban létező adatra lehet hivatkozni
* **Redundancia:** ugyanazon adatok ismétlődése az adatbázisban

**Adatbázis kezelő rendszerek feladatai:**

* **Adatbázis létrehozás, séma definiálása - Adatdefiníciós nyelv (DDL)**
* **Adatok tárolása, kezelése, lekérdezés - Adatmanipulációs nyelv (DML)**
* **Adatbiztonság:**

o nagy mennyiségű adat biztonságos tárolása

o az adatintegritás biztosítása

o hozzáférési szabályok definiálása,

o ne kerüljön be hibás adat és ne módosulhasson hibásra (ehhez általában a séma megadásakor definiálhatók szabályok, de erre törekedni kell az adatbázist felhasználónak pl. programnak is)

* **Adatok megosztása:** egyidejű hozzáférés biztosítása több felhasználónak úgy, hogy ezek egymást ne befolyásolják

**Adatmodell elemei és jellemzőik:**

* **Egyed:** Elkülöníthető egységek / dolgok / objektumok

Példa: Hallgató

* **Egyedtípus:** az egyedek absztrakciója
* **Egyedhalmaz:** a konkrét egyedek halmaza
* **Egyed-előfordulás:** egy darab konkrét egyed
* **Tulajdonság (attribútum)**: Az egyedeket leíró tulajdonságok

Példa: hallgató neve

* Tulajdonságtípus: absztrakt tulajdonság
* Tulajdonságérték: konkrét érték

**Speciális tulajdonságok:**

* **összetett tulajdonság:** amelynek több rész tulajdonsága van példa: lakcím, részei a település, irányítószám, stb.
* **többértékű tulajdonság:** egy tulajdonságnak több értéke lehet példa: keresztnév, van akinek több keresztneve van
* **Kapcsolat:** Egyedek közötti viszonyt írja le.

Típusai: multiplicitás, a résztvevő egyedek száma szerint

**Multiplicitás szerint:**

* **Egy-egy (1-1) kapcsolat:** Minden egyed-előforduláshoz legfeljebb egy pár egyed-előfordulás tartozik a másik egyedből és viszont.

Példa: a Hallgató és a Felhasználó között van egy tulajdonos kapcsolat, hiszen egy hallgatónak egy felhasználója van a rendszerben és egy felhasználó egy hallgatóhoz tartozik.

* **Egy-sok (1-N vagy egy több) kapcsolat:** Az egyik egyedben minden egyed-előforduláshoz több pár egyedelőfordulás tartozik a másik egyedből, de viszont egy egyhez csak egy pár lehet.

Példa: a Kurzus és a Terem között lehet egy „helyszíne” kapcsolat, hiszen egy kurzust egy teremben tartanak, de egy teremben több kurzus is lehet.

* **Sok-sok (N-M vagy több-több) kapcsolat:** Minden egyed-előforduláshoz mindkét egyedben több pár egyedelőfordulás tartozhat a másik egyedből

Példa: a Hallgató és a Képzés közötti „szakos” kapcsolat, hiszen egy hallgató több képzésre is járhat, de egy képzésre több hallgató is járhat.

**Résztvevő egyedek száma szerint**:

* **Bináris kapcsolat:** két egyed vesz részt a kapcsolatban (leggyakoribb)

(Példa: a Hallgató és a Képzés közötti „szakos” kapcsolat)

* **Sokágú kapcsolat:** kettőnél több egyed vesz részt a kapcsolatban

(Példa: a Hallgató, a VizsgaAlkalom és a Kurzus egyedek közötti „vizsgázik” kapcsolat)

* **Rekurzív kapcsolat:** egy egyed többször vesz részt a kapcsolatban

(Példa: a Kurzus saját magával lehet „előfeltétele” kapcsolatban, mivel egy kurzusnak egy másik kurzus lehet az előfeltétele)

* **Kapcsolati szerep:** Az egyednek definiálható szerep a kapcsolatban. Általában egyértelmű a szerep, ilyenkor nem részletezzük. Rekurzív kapcsolat esetén jellemzően szükség van rá.

(Példa: Kurzusnak az „előfeltétel” kapcsolatnál melyik kurzus az előfeltétele a másiknak)

* **Kapcsolat attribútuma:** A kapcsolathoz is tartozhat tulajdonság (attribútum) Példa: a Kurzus és Terem közötti „helyszíne” kapcsolatnak attribútuma az időpont

**Megszorítások:**

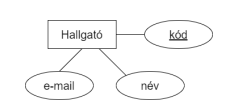
**Kulcs:**

* Egy egyed kulcsa a tulajdonságainak olyan halmaza, amely egyértelműen azonosítja bármely egyed-előfordulást.
* Ha nincs kulcsa az egyednek gyenge egyed, ekkor kell egy másik egyed amely kapcsolatban áll --> meghatározó kapcsolat
* Nem üres: Az attribútum értékét mindig kötelező megadni.
* Egyenértékűség: egy tulajdonság értéke egyszer fordulhat elő(nem kötelező).
* Tartomány: egy érték egy meghatározott értelmezési tartományból vehet fel értéket.
* Hivatkozási épség: egy kapcsolatnál egy egyed-előfordulás csak olyan egyed előfordulásra mutathat, ami létezik.
* Egyéb megszorítás: az adott attribútumra speciális szabály. Egy vagy több adatra is vonatkozhat.

**Egyed-Kapcsolat diagram:**

Jelölések, adatmodellt lehet grafikusan ábrázolni. (EK diagram)

* **Egyed:** téglalap, beleírva a neve 
* **Attribútum:** ellipszis, beleírva a neve, hozzákötve az egyedhez egy vonallal
* **Kulcs:** aláhúzva az attribútum(ok) neve



* **Kapcsolat:** rombusz, beleírva a neve, vonallal hozzákövet a kapcsolódó egyedekhez
* **Kapcsolat típusa:** nyíllal van jelölve, a nyíl az egyértelmű hivatkozás felé mutat, azaz mindig az “egy oldalon” van nyíl. Emiatt az 1-1 kapcsolatnál mindkét oldalon van, 1-N az egyik oldalon, az N-M kapcsolatnál egyik oldalon sincs nyíl.
* **A kapcsolat attribútuma:** ugyanúgy szerepel, de a kapcsolathoz van hozzákötve

**Gyenge egyed:** dupla falú téglalap, a hozzá tartozó meghatározó kapcsolat szintén dupla falú rombusz.

