

Fábián Zoltán – Hálózatok elmélet

Operációs rendszerek

Memóriakezelés

Memóriakezelés

i Fizikai memória

§ Félvezetőkől előállított memóriamodulok

§ RAM - (Random Access Memory) - R/W – írható, olvasható, pl DRAM, SDRAM,

- A dinamikusan frissítendőek : Nagyon rövid időnként a memóriakezelő kiolvassa a memória tartalmát, majd visszaírja a korábbi állapotot, ha nincs frissítés, akkor elveszik az információ.
- Olcsó áramkörök

§ ROM – Read Only Memory – Csak olvasható. Lassabb, mint a RAM. Egyes fajtái UV fénnel vagy Elektromos árammal törölhetők és újraprogramozhatók (EPROM, EEPROM)

§ Flash memória – Külön áramforrás nélkül is megtartja állapotát.

- Nem lehet akárhányszor újraírni!
- Lassabb, mint a DRAM.
- Drágább, mint a DRAM

i Virtuális memória

§ Olyan memóriakezelés, amikor a fizikai memória méreténél nagyobb memóriáról tudnak a rendszerben futó alkalmazások. Általában a plusz helyet a háttértáron kialakított terület biztosítja

§ Pl. Windows: pagefile, Linux:page partíció

i Címzési módok: szegmens:offset

§ 32 bit szegmens és 32 bites offset esetén 2^{64} a megcímezhető memória mérete ~ 4 millió * 4 Millió byte

§ Átlapolt címzés – ugyanazon a címen több elektronika is elérhető felváltva

§ Lehet a szegmenshatár kisebb, mint a szegmens mérete => ugyanannak a fizikai címnek több címe is lehet.

§ Lehetnek változó szegmensméretek is

i Asszociatív memória – Olyan címző áramkör van a processzor és a fizikai memória között, hogy az adatok mozgathatnak nem az adat mozog, hanem a címe változik. A processzor a kívánt sorrendben látja az adatokat. Sokkal gyorsabb, a módosítás, drágábbak az ilyen memóriák

Memóriakezelés

i Memóriakezelő dolga

- § Virtuális vagy fizikai memória foglalás, felszabadítás
- § Osztott (shared) memória létrehozása
- § Fájlok osztott memóriához hasonló elérése (pipe)
- § Memóriavédelem
- § Kernel támogatás

i A memóriefoglalás folyamata

- § Egy lépéses foglalás – A címtartomány és a virtuális memóriát egy lépésben foglalja – pazarló
- § Két lépés (reserve – Címtartomány foglalása, commit – Virtuális memória foglalása) – gazdaságosabb. A commit csak akkor fut le, ha ténylegesen szükség van a memóriára
- § Laponként foglal területet a memóriakezelő. A lapok mérete 2-4 kb.

Memóriagazdálkodás – Fizikai tárkezelés

- ! Rögzített címzés – a szegmens és offszet is rögzített
- ! Áthelyezhető címzés – a cím szegmens része a bázisregiszterben van
- ! Fizikai tárkezelés
 - § Overlay – Állandó rész a memóriában, cserélődő modulok a háttértáron
 - § Tárcsere
 - Swapping – teljes memóriát cseréljük
 - Állandó partíciók – a partíciókat cseréljük (belső elaprózódás) – egy partíció – egy alkalmazás. A partícióban van szabad terület
 - Rugalmas partíciók – külső elaprózódás (sok apró üres memóriaszelet van a partíciók között) => garbage collection
 - Paging – A memóriát kis méretű lapokban kezeljük. A memórialapok és a folyamatok kapcsolatát laptáblák határozzák meg.

Memóriagazdálkodás - Virtuális tárkezelés

- ! A folyamatok kódjának éppen csak a futó része és környezete legyen a fizikai memóriában
- ! Kevés háttértár művelet kell => lokalitási elv => az alkalmazások 95%-ban sorfolytonosan helyezkednek el a háttértáron és a memóriában is
- ! A fizikai és a virtuális tár lapokra van osztva.
 - § Laptábla tárolja az egyes fizikai lapok helyét és tulajdonságait
 - § A laptáblában a helyen kívül az is benne van, hogy bent van-e a fizikai memóriában a lap.
 - Laphiba - ha a kívánt lap nincs a fizikai memóriában => azt be kell tölteni (esetleg betöltjük a környezetét is!)
 - Előtte ki kell menteni valamelyik korábban használt lap tartalmát . Ezt végzik a lapcsere algoritmusok.
 - Piszkos lapot nem kell menteni (nem írtunk bele)
 - Cél: a lehető legkevesebb laphiba legyen működés közben

Memóriagazdálkodás 3

! Lapkiosztási elvek

- § Egyenletes, arányos, prioritásos
- § Globális és lokális lapkiosztási elv

! Vergődés

- § Ha túl sok a laphiba, akkor a rendszer a legtöbb időt a memóriakezeléssel tölti
- § Pl. Kevés fizikai memória egy szerverben – a válaszidők drámaian megnőnek

! Lapcsere stratégiák

- § OPTimális (legkésőbb lesz szükség)
 - § FIFO – First In first Out
 - § LRU - Last Recently Used
 - § Second Chance – Második esély (Mielőtt kivennénk egy lapot a fizikai memóriából előtte még félretesszük egy kicsit, hátha szükség lesz rá)
- ! Programozó is tud javítani a laphibák számán megfelelő programozási technikákkal