**8. Fő Memória**

# Hardveres címvédelem

A hardveres címvédelem az operációs rendszer alapvető biztonsági funkciói közé tartozik, amely a felhasználói és rendszerszintű folyamatok közötti memóriaterületek elszeparálásával védi meg az adatokat és az erőforrásokat. Ennek a védelemnek egyik gyakori megvalósítása a bázis- és limit-regiszterek használata.

Bázis- és Limit-Regiszterek

Bázisregiszter: Tartalmazza a legkisebb jogos memóriacímet, amelyet a folyamat elérhet.

Limitregiszter: Meghatározza a jogos memóriacímek tartományának nagyságát.

Ezeket a regisztereket csak az operációs rendszer tudja beállítani különleges, privilégiumot igénylő utasításokkal. A CPU ezután minden felhasználói módban generált címet összehasonlít ezekkel a regiszterekkel, és ha a program próbál hozzáférni az operációs rendszer memóriájához vagy más felhasználók memóriájához, a rendszer csapdát generál, ami súlyos hibaként kezeli a hozzáférési kísérletet .

Memóriaterület Védelme

A memóriaterületek védelmét úgy érik el, hogy a CPU hardvere minden címet összehasonlít a regiszterekben tárolt értékekkel. Ha egy felhasználói program próbál hozzáférni a jogosulatlan memóriacímekhez, a rendszer leállítja a műveletet és csapdába esik, ami lehetővé teszi az operációs rendszer számára, hogy kezelje a hibát. Ez a megoldás megakadályozza a felhasználói programokat abban, hogy véletlenül vagy szándékosan módosítsák az operációs rendszer vagy más felhasználók kódját és adatszerkezeteit.

Ez a védelmi mechanizmus létfontosságú minden többfelhasználós rendszerben, ahol a folyamatok és az adatok elszeparálása elengedhetetlen a rendszer biztonságos működéséhez.

# Logikai vs. fizikai címtér, MMU

A logikai és a fizikai címtér különbsége, valamint az MMU (Memória-kezelő Egység) szerepe az operációs rendszerekben alapvető a memória hatékony kezelésében.

Logikai és Fizikai Címtér

Logikai Címtér: A CPU által generált címeket logikai címnek nevezzük. Ez a cím a program által generált címek halmazát jelenti, amelyeket a program futása során használnak.

Fizikai Címtér: A memóriaegység által látott címek fizikai címek. Ezek a címek azok, amelyeket a memória címregiszterébe töltünk, és ténylegesen a memória egy konkrét fizikai helyére mutatnak.

A fordítási időben és betöltési időben alkalmazott címösszekötési módszerek azonos logikai és fizikai címeket generálnak. Azonban az végrehajtási időben alkalmazott címösszekötés során eltérő logikai és fizikai címek jönnek létre. Ebben az esetben a logikai címet gyakran virtuális címként hivatkozzuk, amit a szövegben felváltva használunk .

MMU (Memória-kezelő Egység)

Az MMU egy hardvereszköz, amely a virtuális címeket fizikai címekké alakítja át futásidőben. Ez a fordítás többféle módszerrel megvalósítható, amelyek között választhatunk. Az egyik egyszerű megközelítés, amit bemutatnak, az úgynevezett alapregiszteres sémát használja, ahol az alapregiszter mostantól áthelyezési regiszterként ismert. Az áthelyezési regiszter értékét hozzáadják minden felhasználói folyamat által generált címhez, amikor a címet a memóriába küldik. Például, ha az alap 14000, akkor egy 0 helyre történő felhasználói hivatkozás dinamikusan 14000-re kerül áthelyezésre; egy 346-os hozzáférés 14346-ra lesz leképezve .

Ezek a funkciók biztosítják, hogy az operációs rendszerek hatékonyan kezelhessék a memóriát, biztosítva a folyamatok közötti memóriaizolációt és a memóriaterület hatékony kihasználását.

**c: Dinamikus linkelés, swapping**

A dinamikus linkelés és a swapping (memóriacserélés) két alapvető technika az operációs rendszerek memória-menedzsmentjében, amelyek jelentősen befolyásolják a rendszer teljesítményét és rugalmasságát.

Dinamikus Linkelés

A dinamikus linkelés lehetővé teszi, hogy a rendszerkönyvtárak csak futásidőben kapcsolódjanak a felhasználói programokhoz. Ez azt jelenti, hogy a programok nem tartalmazzák a használt könyvtárrutinok másolatait, hanem egy közös könyvtári kódot használnak, ami csökkenti a lemezterület és a fő memória használatát. A dinamikus linkelés során, amikor egy könyvtári hivatkozás futtatásra kerül, a rendszer először ellenőrzi, hogy a szükséges rutin jelen van-e a memóriában. Ha nem, a rutin betöltésre kerül. Ez a mechanizmus növeli a kódmegosztást és rugalmasságot teszi lehetővé, például hibajavításoknál az egyszerű frissítéseket anélkül, hogy újra kellene linkelni a programokat .

Swapping

A swapping egy memóriakezelési technika, amely lehetővé teszi a folyamatok ideiglenes ki- és becserélését a memóriából egy háttértárba, így növelve a multiprogramozás mértékét. A rendszer dinamikusan mozgathatja a folyamatokat a memória és a háttértár (például gyors lemez) között, attól függően, hogy éppen mely folyamatok igénylik a memória erőforrásait. A standard swapping magában foglalja a folyamatok mozgatását a fő memória és egy háttértár között. Ez a technika lehetővé teszi, hogy a rendszerben lévő összes folyamat fizikai címtere meghaladja a valós fizikai memória méretét .

Ezek a technikák kritikus szerepet játszanak az operációs rendszerek hatékonyságának és rugalmasságának növelésében, biztosítva, hogy a rendszerek optimálisan kezeljék a rendelkezésre álló memóriát, és alkalmazkodjanak a változó felhasználói igényekhez.

**d: Egybefüggő kiosztás, töredezettségek**

Az egybefüggő memóriaallokáció és a töredezettség két kulcsfontosságú fogalom az operációs rendszerek memória-kezelésében.

Egybefüggő Allokáció

Az egybefüggő allokáció esetén minden fájlnak egybefüggő blokkokon kell helyet foglalnia a lemezen. Ez a módszer egyszerűvé teszi a fájlhoz való hozzáférést, mivel a fájl blokkjai egymás után helyezkednek el, így a lemezfej kevés mozgatásával elérhetők. Azonban problémát jelenthet új fájlok számára helyet találni, különösen ha a lemezterületet már többször felosztották és felszabadították, ami az úgynevezett külső töredezettséghez vezet​​.

Töredezettségek

Külső töredezettség: Akkor fordul elő, amikor a szabad terület a lemezen kis, egymástól szeparált szegmensekre oszlik, amelyek egyedül nem elegendőek nagyobb fájlok tárolására. Ez a probléma az egybefüggő allokációval súlyosbodik, mivel nagy, egybefüggő területek megtalálása egyre nehezebbé válik az idő előrehaladtával​​.

Belső töredezettség: Ez akkor jelentkezik, amikor a rendszer több memóriát rendel egy folyamathoz, mint amennyire valójában szüksége van. Ez a helyzet különösen akkor fordul elő, ha a memória-allokáció fix blokkméretben történik​​.

Megoldások a Töredezettségre

Az egybefüggő allokációs problémák kezelésére több megoldás létezik:

Dinamikus allokáció: Ez lehetővé teszi a rendszer számára, hogy valós időben változtassa meg a lefoglalt terület méretét, például kiterjesztések (extents) hozzáadásával, amelyek további egybefüggő blokkokat foglalnak le​​.

Kompaktálás: Ez a memória tartalmának újrarendezését jelenti a szabad helyek egybefüggővé tételére, ami lehetővé teszi nagyobb méretű kérések kielégítését. Ez azonban költséges művelet lehet és általában csak offline módban végezhető el​​.

Ezek a megoldások segítenek kezelni a memóriaallokációs problémákat, bár minden megoldásnak megvannak a maga korlátai és költségei. A rendszermenedzsereknek gondosan kell mérlegelniük ezeket a tényezőket a rendszer teljesítményének és hatékonyságának maximalizálása érdekében.

**e: Szegmentálás, szegmenstábla**

A szegmentálás egy olyan memóriakezelési technika, amely lehetővé teszi a folyamatok logikai címtérnek nem egybefüggő fizikai címtérre való leképezését. Ezt úgy valósítja meg, hogy a programot logikailag értelmezhető egységekre, úgynevezett szegmensekre bontja. Mindegyik szegmens egy logikai címtér részeként van kezelve, és van egy azonosítója (szegmens-szám), valamint egy mérete (offset).

A szegmenstábla egy olyan adatstruktúra, amely a szegmensek báziscímét és limitjét tárolja, így segítve a logikai címek fizikai címekké való átalakítását. A logikai cím egy szegmens-számból és egy adott szegmens belüli eltolásból (offset) áll, amit a szegmenstábla használatával fizikai címmé alakítanak. Ha a logikai cím eltolása meghaladja a szegmens limitjét, az operációs rendszer csapdába esik, amit kezelni kell .

Ez a szegmentálási rendszer lehetővé teszi a memória rugalmasabb kezelését, ahol a programozók különálló logikai egységekben gondolkodhatnak, és a rendszer dinamikusan kezelheti a memória fizikai elosztását. A szegmentálás előnyei közé tartozik a könnyebb megosztás és védelem, valamint hogy támogatja a programok dinamikus növekedését a memóriában.

**f: Lapozás, címfordítás, laptábla megoldások**

A lapozás egy memóriakezelési technika, amely a logikai memóriát azonos méretű blokkokba, úgynevezett lapokba bontja, amelyek fizikai memóriába tetszőleges keretekbe (frame-ekbe) tölthetők be. A lapozás segítségével a logikai memória egymástól függetlenül kezelhető a fizikai memóriától, amely lehetővé teszi, hogy egy folyamat memóriája a fizikai címteren belül nem egybefüggő legyen.

A címfordítás a CPU által generált logikai címek fizikai címekké alakítását jelenti, amely a laptábla (page table) használatával történik. A laptábla minden oldalhoz tartozó fizikai cím alapcímet tartalmaz, amelyhez hozzáadódik a lapbeli eltolás (offset) az aktuális memóriacím előállításához.

A laptábla megoldások különböző módszereket foglalnak magukban a címfordítás és a memória hatékony kezelése érdekében:

Egyszintű laptábla: Minden logikai lap egyetlen laptáblában kap egy bejegyzést, amely közvetlenül a fizikai címre mutat.

Többszintű laptábla: A nagy logikai címtér kezelésére, több szinten keresztüli laptáblákat használnak, ahol az egyes szintek a következő szint laptáblájának címét tárolják, végül elérve a kívánt fizikai lap címét .

Ezen megoldások segítenek a memória rugalmas és hatékony kezelésében, illetve a rendszer teljesítményének optimalizálásában azáltal, hogy csak a szükséges lapok vannak betöltve, és a címfordítás gyorsan végezhető.

**g: Lapozás TLB-vel**

A lapozás és a Translation Lookaside Buffer (TLB) használata az operációs rendszerek fontos memóriakezelési technikája, amely jelentősen csökkenti a memóriához való hozzáférés idejét.

Lapozás

A lapozás során a logikai memória egységekre, ún. lapokra oszlik, melyek egy-egy kerethez (frame) rendelhetők a fizikai memóriában. Ez lehetővé teszi, hogy a logikai címek fizikai címekre leképezése rugalmas és hatékony legyen, függetlenül attól, hogy a fizikai memóriacímek egymástól nem egybefüggőek.

TLB Használata

A TLB egy speciális gyorsítótár, amely a lapozási tábla leggyakrabban használt részeit tárolja, így gyorsítva a címfordítási folyamatot. Amikor a processzor egy logikai címet generál, a TLB-ben keresi meg a megfelelő lap számot. Ha a lap száma megtalálható a TLB-ben (TLB hit), akkor azonnal elérhető a megfelelő keret száma, és a memóriahozzáférés gyorsan megtörténik. Ha a lap száma nem található meg a TLB-ben (TLB miss), akkor a rendszernek a lapozási táblához kell fordulnia a keret számának megszerzése érdekében .

TLB Miss és Kezelése

TLB miss esetén a rendszernek memóriareferenciát kell tennie a lapozási táblához, ami az adott lap keretszámának lekérdezését jelenti. Ha a keretszám megszerzése után újra felkerül a TLB-be, így a következő hivatkozásnál gyorsabb hozzáférést tesz lehetővé. Ha a TLB teljes, egy meglévő bejegyzést ki kell választani és helyettesíteni kell a friss adattal. A helyettesítési stratégiák közé tartozik a legkevésbé nemrégiben használt (LRU), körkörös, és véletlenszerű stratégiák .

A TLB és a lapozás kombinációja jelentősen növeli az operációs rendszerek memóriahatékonyságát, csökkenti a címfordítási időket és optimalizálja a rendszer teljesítményét.