

## Előadás\_#12.

### 1. Az NT alapú operációs rendszerek memóriakezelése

A Windows feladatkezelő (a Teljesítmény fülön) a memória állapotáról is ad információt a felhasználó számára, a következők szerint. A Memória elnevezésű területen ezt láthatjuk:

- Windows 10 esetén a legfelső sorban található az összes érték, azaz a számítógépre telepített RAM mennyisége gigabájtban. A következő négy érték grafikusán is ábrázolásra kerül a „Memóriamegoszlás” részen. A kurzort a megfelelő terület fölé mozdítva jelenik meg a részletes információ.
- A „Használatban” érték a folyamatok, az illesztőprogramok illetve az operációs rendszer által használt memória mennyiségét jelzi. A CPU-k teljesítményének növekedése tette lehetővé azt, hogy ezen memória mennyiség egy részét a rendszer tömörített formában is képes legyen kezelni, amivel így fizikai memória helyet tud megtakarítani.
- A „Módosított” érték annak a memória területnek a méretét jelzi, amit háttértárra kell áthelyezni.
- A „Készenléti” az a memóriamennyiség, mely aktívan nem használt, gyorsítótárazott adatokat és programkódokat tartalmaz. A Windows 10 sajnos hajlamos az összes rendelkezésre álló memóriát ide besorolni, ami a legjobb szándék ellenére is lassuláshoz vezet.
- A „Szabad” érték a jelenleg használaton kívüli, hasznos információt nem tartalmazó memóriát jelöli, melyet a rendszer elsőként oszt ki újra, ha a folyamatoknak, illesztőprogramoknak és az operációs rendszernek további memóriára van szüksége.
- A „Gyorsítótárazott” érték a jelenleg elfoglalt, de kihasználatlan memórialapok méretét jelöli.
- A „Rendelkezésre álló” érték az, amely elérhetővé tehető a folyamatok, az illesztőprogramok és az operációs rendszer általi használatra.

Ezen kívül érdemes megnézni a virtuális memória állapotát is. Gyakorlatilag a szabadnak jelzett memória területnél nagyobb memóriaigényű program is elindítható, hiszen egyrészt az osztott memóriahasználat miatt lehet közös memóriaterülete más programokkal, másrészt a foglalt, de használatban nem lévő memória egy része akár át is helyezhető háttértárra. Ha a program, amit el akarok indítani, sok megosztott memóriát használ, akkor lehet, hogy sokkal kevesebbet foglal el más programokkal együtt futtatva, mint akkor, ha csak egyedül fut.

A memória maximális méretének fizikai okokból (azaz az architektúrából következő) illetve üzleti megfontolásokból fakadó különböző korlátai vannak. Ezen korlátok részleteivel az Operációs Rendszerek tárgy első gyakorlata foglalkozott.

A 32-bites (x86) illetve a 64-bites (x64) rendszereket megvizsgálva szembeötlő különbség a fizikai memória mérete. Természetesen a 32-bites rendszerek egy részénél a Microsoft bevetett memória méret növelő eszközöket. Ilyen eszköz például a Physical Address Extension (PAE). A PAE egyes 32-bites processzorokban meglévő támogatás, aminek a lényege az, hogy ez esetben 32 címbit helyett 36 címbit van használatban. Amelyik 32-bites operációs rendszer ezt ki tudja használni, az képes akár 64GB fizikai memóriát is kezelni. Figyelni kell azonban arra, hogy a folyamatok címtére továbbra is 32-bites marad, így csak az operációs rendszer tud több fizikai memórialapot kiosztani a folyamatoknak.

Jellemzően a 32-bites rendszereken (ahol nincs PAE), 4 GB fizikai memória esetén is néha kevesebbet lát az operációs rendszer, mert a memóriaterület felső részére I/O eszközöket szokás berakni (pl. ha van egy 512 MB memóriagényű "alaplap" videokártyánk, akkor nyilván csak 3,5 GB-ot lát szabadon az operációs rendszer).

Kliens operációs rendszerek esetében nem használatos a PAE támogatás, mert a kliensekben lévő eszközök meghajtói jellemzően nem, vagy nem megfelelően kezelik le a 4GB-nál több fizikai memóriát.

A 64-bites NT alapú operációs rendszerek kompatibilisek továbbá az IA-64-el (Intel Architecture 64), egy az Intel és a Hewlett-Packard által közösen kifejlesztett utasításkészlet-architektúrával is (Instruction Set Architecture, ISA; az utasításkészlet-architektúra az utasításokon kívül a felépítés több más elemét is meghatározza, pl. az adatformátumokat, regisztereket, címezési módokat, a memória felépítését, a megszakítás- és csapdakezelést, valamint a kapcsolatot a külvilág felé – I/O). Az IA64 architektúra jelenleg csak az Itanium és az Itanium 2 processzorokban került megvalósításra.

A Windows NT alapú operációs rendszerek memóriakezelőjének egyik alapvető feladata, hogy a folyamatok virtuális címtérének címeit hozzárendelje a fizikai címekhez, így valósítva meg a logikai- és fizikai címek transzformációját. A másik alapvető feladata a virtuális memóriakezelés megvalósítása. A gyakorlatban ez azt jelenti, hogy a memória telítődésekor bizonyos (pl. a régen nem használt) memóriaterületeket a háttértárra menti. Természetesen a folyamat kétirányú, azaz ha olyan memórialapra történik hivatkozás, amely a háttértáron található, akkor memóriakezelő a keresett lapot visszamozgatja az operatív memóriába. A mozgatás alapegysége minden esetben a memória lap (Page).

A fent említett alapszolgáltatáson felül a Windows NT alapú operációs rendszerek memóriakezelője a rendszer hatékonyságát megnövelő többlétszolgáltatásokat is nyújt a felhasználóknak. A legfontosabb ilyen szolgáltatás a fájlok memóriaként történő elérése (Memory Mapped Files), ami lehetővé teszi a fájlok osztott elérését, illetve a Copy-on-Write mechanizmus használatát.

A Windows NT 32-bites memóriakezelője x86 címtartományban alapesetben két egyenlő részre osztja a memóriát, azaz a 4GB-ot egy alsó és egy felső 2 – 2GB-ra. Az alsó 2GB az operációs rendszeré, a felső 2GB pedig az alkalmazásoké.

Alapesetben egy 32-bites rendszerben egy felhasználói folyamat maximum 2GB-os címtartományt használhat fel a saját kódjának és adatának tárolására. A boot.ini-ben a /3GB kapcsoló aktiválásával az arány 3GB – 1GB-ra változik. Ez eleve egy kényszer megoldás volt, azon cél érdekében, hogy egy folyamat – például egy nagy adatbázis-kezelő – a 64-bites rendszer nélkül is képes legyen 2GB feletti memóriahasználatra. Vista óta ezt 4GT-nek hívják (4-Gigabyte Tuning).

64-bites memóriakezelés esetén a címtér (ami egyébként 48-bites) maximuma jelentősen megnő, hiszen 6.657GB válik címezhetővé a rendszer folyamatainak számára, és 8.192GB a felhasználói folyamatok számára. Nyilván ehhez a megfelelő mennyiségű memóriára is szükség lenne, azaz az x64 egy lehetőség, amit megfelelő HW környezetben ki tudunk használni, hiszen jelenleg a maximális fizikai memóriaméret 2.048GB, azaz a rendszerben van még bőven tartalék.

A memóriakezelés alapelvei:

- Virtuális tárkezelés

- Lapszervezés (4kB / 2MB méretű lapok, 2/3/4 szintű).
- Lapozófájl használata.

Kezdetben az igény szerinti lapozás volt jellemző, de ez nem bizonyult kellően hatékonynak.

Clustering: Az igény szerinti lapozás kiegészítése azzal, hogy a kért lap környékén lévő pár lapot is behozta (lokalitás elv).

Prefetch: XP óta van egy Prefetcher, ami rögzíti, hogy a programok induláskor miket szoktak igényelni, és az azokhoz tartozó lapokat előre behozza. A gondot az okozza, hogy a rendszer Prefetch nélkül először pár lapot az egyik fájlból, majd pár lapot a másik dll-ből tölt be (ahogy éppen az utasítások jönnek a program kódban), és így a merevlemezen a fejnek össze-vissza kell mozognia. Ezért egy adott program esetében a Prefetcher az első 10 másodperc hozzáféréseit megjegyzi és következő programindításkor ezek szerint jár el, hiszen adott program

esetében mindig ugyanazokat a lapokat kell betölteni. A lapok ismeretében a betöltés lehet aszinkron módú is.

Vista óta további Prefetch jellegű szolgáltatások is vannak:

Superfetch: 8 szintű prioritást rendel a memórialapokhoz, követi a betöltött lapok "életciklusát", így képezve egy stand-by listát, amire felkerülnek azok a lapok, amelyek nincsenek a memóriában, de előreláthatóan hamarosan kelleni fognak.

ReadyBoost: Frappánsan hívhatjuk úgy, hogy "a szegény ember SSD-je", hiszen funkcióját USB Flash alapú eszköz használatával éri el. Célja a merevlemez hosszú elérési idejének csökkentése, ReadyBoost tehát nem a memória bővítésére szolgál, hanem a lassú lemezműveletek felgyorsítására. Lényege hogy egy program számára a HDD-ről használandó adatok előre áttöltődnek a Flash meghajtóra, ahonnan a rendszer gyorsabban éri el azokat.

- Hatékonyság
  - Igény szerinti lapozás + Clustering + Prefetch.
  - Memória megosztás, Copy-on-Write.
  - Fájl cache-elés memóriában (Memory Mapped File).
- Biztonság
  - Minden folyamatnak külön címtartomány.
  - Elérés leírókon keresztül (hozzáférési token).

A memóriakezelő felhasználói interfész

A memóriakezelő szolgáltatásai a Win32 API-n keresztül érhetők el, szolgáltatásai a következők:

- Virtuális memória allokáció, illetve felszabadítás.
- Osztott elérésű (Shared) memória létrehozása.
- A file-ok osztott elérésű memóriához hasonló elérése.
- Virtuális memória kezelés  
(pl.: információkérés, tárolás memóriába, kiírás háttértárra).
- Memória védelmi funkciók.
- Kernel szintű funkciók (elsősorban a Device Driver-ek) támogatása  
(pl.: fix fizikai memóriaterület használata).

## A folyamatok memóriefoglalása

A memóriefoglalás mindig két lépésben történik. Előnye, hogy mindig csak annyi memória kerül lefoglalásra, amennyire a folyamatnak ténylegesen szüksége van.

- Reserve

A virtuális címtartomány lefoglalása, a szándék kifejezése, nem jelent fizikai memóriefoglalást. Jelzi az operációs rendszer számára, hogy mennyi memória lesz, vagy lehet szüksége. Alapértelmezett foglalási méret x86 és x64 esetében is 1MB, 4kB-os lapok formájában.

- Commit

A virtuális memória tényleges lefoglalása. Alapértelmezett foglalási méret x86 és x64 esetében is 4kB, azaz egy lapnyi. A Commit művelet a Reserve művelettel lefoglalt területen belül hajtódik végre, igény szerinti lappal. Az a folyamat, amelyik a memória foglalását kezdeményezte csak a Commit művelet után veheti a memória lapokat használatba.

A két folyamat hívása történhet egy függvénnyel is, de fizikailag ebben az esetben is két lépésben fog végrehajtódni.

A két lépcsőben történő foglalás előnye a hatékonyabb működés:

- Reserve csak címtartomány foglalást végez.  
(egy bejegyzés az operációs rendszer foglalási táblázatában)
- Tényleges erőforrás használat csak a Commit után indul.
- A folyamatok futtatásához használt fizikai memóriaigény csökkenthető.
- Lehetséges előre lefoglalni egybefüggő címtartományokat a rendszer terhelése nélkül.

## A szálak user módú veremtárának (User Stack) memória foglalása

Alapkövetelmény, hogy a user módú veremtár egybefüggően kerüljön kijelölésre, azaz címtartománya folyamatos kiosztású legyen.

Egy szál indulásakor a Reserve jelzi az 1MB igényt, de a Commit fizikailag csak 2 lapot, azaz 2x4kB-ot fog lefoglalni. Az első lapban lesz a user módú veremtár indulási címe, míg a másik lap azt a célt szolgálja, hogy induláskor, pontosabban az első lap teljes kitöltése után további lapot, lapokat lehessen újabb Commit művelettel lefoglalni.

Amennyiben nem a Commit/Reserve rendszerben, az nem két- hanem egy lépésben történne a foglalás, akkor minden egyes szál indulásakor le kellene foglalni egy 1MB-os területet, melynek jelentős részét a szál valószínűleg nem is használná.

## Osztott elérésű memória (Shared Memory)

Az osztott elérésű memória optimális memóriagazdálkodást tesz lehetővé, hiszen amennyiben két vagy több folyamat azonos adatokat használ, akkor ezen adatoknak elég fizikailag egyszer megjelenni a memóriában. Azaz a folyamatok adott virtuális címtartományához azonos fizikai memórialap kerül kiosztásra.

Például ha a rendszerben két folyamat is éppen C programot fordít. Az osztott memória használata lehetővé teszi, hogy a C fordító csak egy példányban foglaljon helyet a memóriában.

A kezelés alapeleme egy speciális objektum, az úgynevezett szekció objektum (Section Object). A szekció objektum jellemzői:

- neve
- helye, azaz hogy melyik virtuális memóriaterületben található
- az objektum indulásakor hozzá társított fájl vagy fájl részlet
- ACL (Access Control List) -ben meghatározott elérhetőségi paraméterei

## A memória védelme

A memória védelme alatt azt értjük, hogy – az osztott elérésű memórián kívül – a se a folyamatok, se az operációs rendszer ne legyen képes módosítani egy másik folyamat címtartományába tartozó memóriaterületeket. Ez egy roppant fontos mechanizmus, ami a Windows NT alapú operációs rendszerek megbízhatóságát biztosítja.

A memóriavédelem megvalósítási szintjei:

- Kernel módban futó modulok adatstruktúrái a felhasználói módú folyamatok számára elérhetetlenek.
- Minden folyamat önálló logikai memória területet, címtartományt használ, amit a memóriakezelőn keresztül ér el, ami biztosítja a hardware-rel támogatott logikai-fizikai transzformációt, és a kizárólagos hozzáférést a memóriaterülethez.
- Processzorfüggő memória védelmi attribútumok  
r, w, x, (írás, olvasás, végrehajtás)  
A kódszegmensek memóriaterületei jellemzően csak olvashatók.
- ACL (Access Control List) az elérést szabályozó védelmi lista, amely arról rendelkezik, hogy az adott objektumot melyik folyamatok érhetik el.

## A memória foglalása

A Windows NT alapú operációs rendszerek a memóriát mindig memórialaponként értelmezik és kezelik. A memóriakezelő feladata az, hogy kevés szabad fizikai memória esetén az arra kiválasztott lapokat a lapozás (Paging) művelettel a háttértárra mentse.

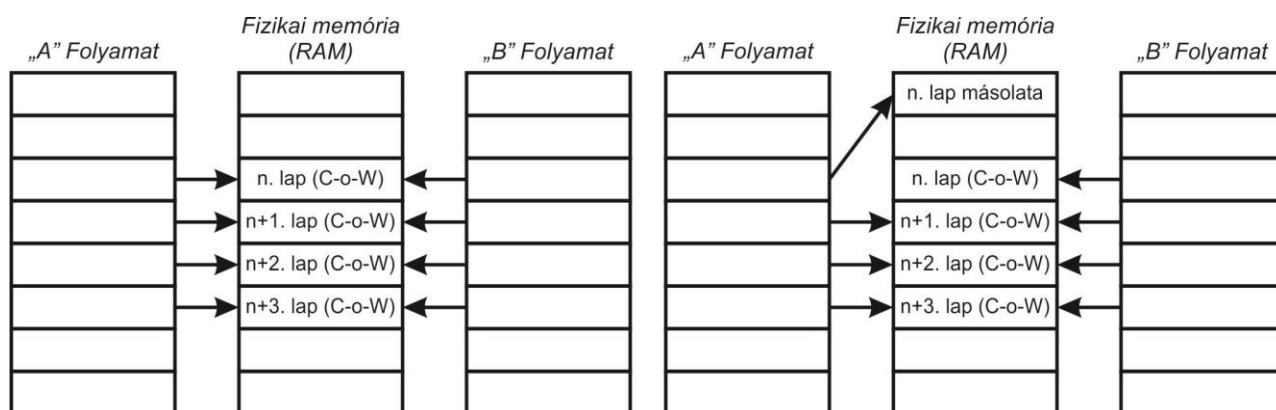
- Kernel komponensek esetén nem mindig megengedett a lapozás
  - kernel szinten megvalósított kölcsönös kizárás esetén
  - driver-ek által használt memóriaterületek fixen a memóriában maradnak, még akkor is, ha csak ritkán használjuk őket
- A kernel komponensek memória foglalása kétféle képpen történhet
  - lapozott memória tárból (Paged Memory Pool)
  - nem lapozott memória tárból (NonPaged Memory Pool)

A nem lapozott memória tár a memória azon védett részét jelenti, ami semmiképpen sem lapozódhat ki az operatív tárból, azaz a fizikai memóriából.

## Copy-on-Write mechanizmus (másolás írás esetén)

A Copy-on-Write mechanizmuson alapuló memóriahasználat lényege a következő. Amennyiben egy folyamat egy olyan – írható és olvasható, más folyamatok által is potenciálisan használt – szekció objektumot kezd használni, akkor a rendszer alapértelmezésben csak átírja a memórialapokhoz tartozó Copy-on-Write (C-o-W) flag-et, de még nem készít a folyamat számára lokális másolatot.

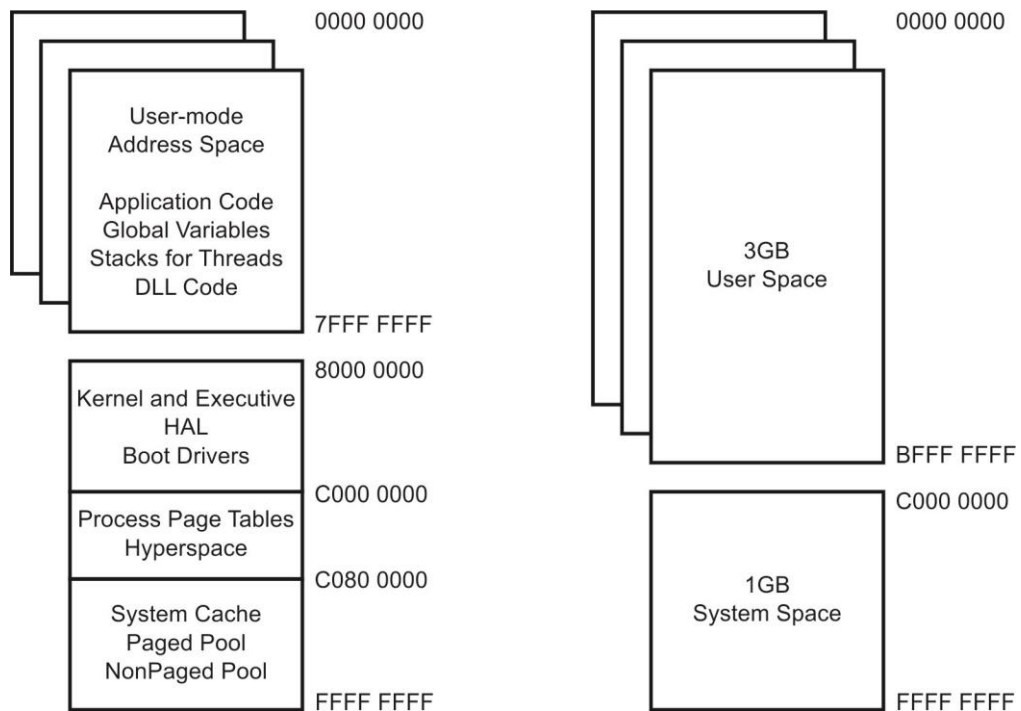
Tényleges másolás a rendszerben tehát csak azután történik, amikor az érintett szekció objektum bármelyik használója kezdeményez egy memória írási műveletet. Ekkor a rendszer mielőtt megváltoztatná a memóriatartalmat, a kérdéses memórialapról egy privát másolatot készít az írást kezdeményező folyamat számára. A másolat tehát azért és akkor készül az egész szóban forgó memória lapról, amikor az a cél, hogy a memória megváltoztatása csakis a kérdéses (az írást kezdeményező) folyamat számára legyen látható.



A Copy-on-Write mechanizmus a lusta kiértékelés (Lazy Evaluation) elvét követi. A lusta kiértékelés – mint elv – azt a végrehajtási módot jelenti, amikor a rendszerben a sok erőforrást igénylő (azaz költséges) műveletek egészen addig elhalasztásra kerülnek, amíg azok végrehajtása már ténylegesen elkerülhetetlené nem válik.

A logikai címtér felépítése

A 32-bites Windows NT alapú operációs rendszerek alapértelmezett 2GB+2GB címtérét a bal oldali ábra, a /3GB kapcsoló bekapcsolása utáni 1GB+3GB állapotot a jobb oldali ábra mutatja.



- Az alkalmazások címtartománya:
  - legfelső címtartomány a rendszer dinamikusan foglalt elemei
- Operációs rendszer címtartománya:
  - Kernel, HAL, Executive, Boot Drivers
  - A folyamatok memórialapjainak táblái
  - System Cache, Memória címtranszformációs táblázatok