

# *Operációs rendszerek*

Windows NT



## A Windows NT

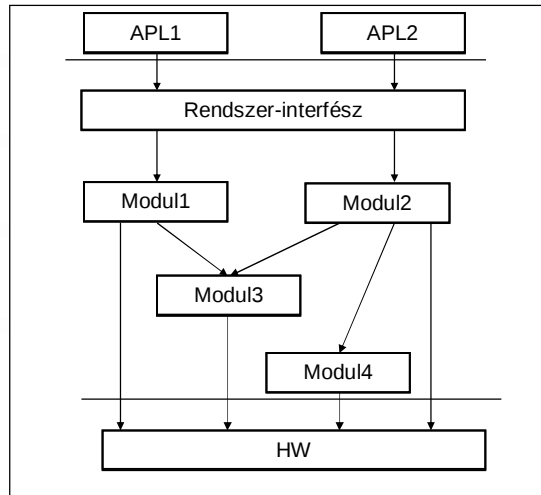
- Felépítésében is új operációs rendszer:  
*New Technology (NT)*
- 32-bites Windows-os rendszerek felváltása
- Windows 2000: *NT alapú*

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Operációs rendszerek felépítése

- Monolitikus kernel



hétfő, 2005. december 5.

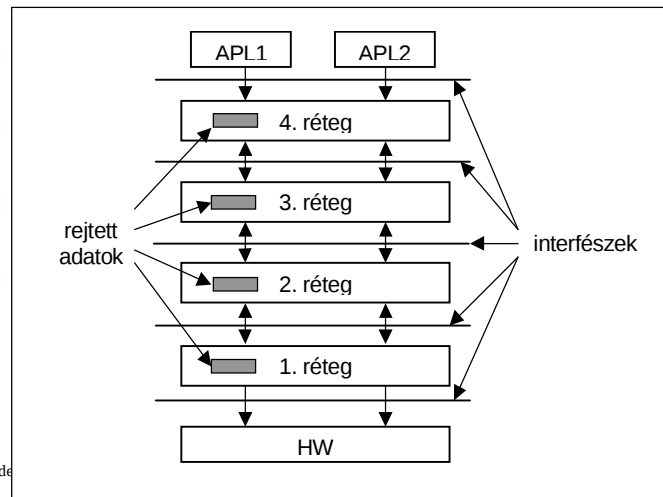
## Monolitikus kernel

- Közvetlen kommunikáció modulok között.
- "Közösen" használt adatok.
- Nincs szabványos érintkezési felülete (interfésze) a moduloknak.
- Nem megbízható:  
egy hibás modul hibája továbbterjedhet a többi modul felé.
- Nehezen karbantartható, fejleszthető.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Réteg szerkezetű operációs rendszer I.



hétfő, 2005. de

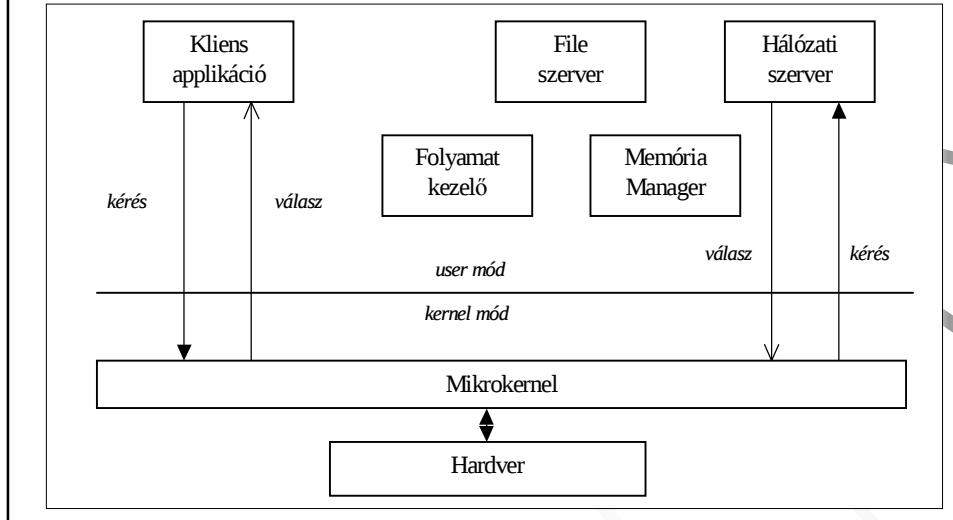
## Réteg szerkezetű operációs rendszer II.

- A modulok rétegszerűen épülnek egymásra (objektum-orientált megközelítés).
- A kommunikáció jól definiált interfészekon keresztül bonyolódik.
- A rétegek csak a felettük illetve alattuk levő réteget érik el közvetlenül.
- Az interfészek használata:
  - *biztonságos:* ellenőrizhető a kérés/hozzáférés jogossága,
  - *megbízható:* nem terjed a hiba,
  - *karbantartható:* fejlesztés, módosítás egyszerű.

hétfő, 2005. de

Operációs rendszerek III.

## Kliens-szerver modellre épülő operációs rendszer I.



## Kliens-szerver modellre épülő operációs rendszer II.

- Önálló, jól definiált funkciókat ellátó komponensek.
- Egymásra épülő szolgáltatások nyújtása, ill. azok használata.
- Mag az ún. *mikrokernel* (mikrokernel-es operációs rendszerek).
- Funkciója:
  - Szolgáltatás kéréseket üzenetek formájában továbbítja az operációs rendszer moduljai számára.
  - Biztosítja a folyamatok számára a hardver elérését. (Hardver elérés csak mikrokernelen keresztül.)

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Kliens-szerver modellre épülő operációs rendszer III.

- A modulok önálló szerver folyamatok.
- Mikrokernel: privilegizált (kernel) mód.
- Folyamatok: felhasználói mód.
- Előny: rugalmas, biztonságos működés.
- Hátrány: nem hatékony hardver kihasználás (környezetváltások!).

Pl.: *Mach* operációs rendszer.  
UNIX daemon folyamatok

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Az NT felépítésének fő jellemzői

- Réteg szerkezet.
- Kliens-szerver működés (alrendszerek, szolgáltatások stb.).
- Szigorú interfész használat.
- Hatékony hardver kihasználás: user módban megvalósítható funkciók kernel módban vannak.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Az NT objektum-orientált szemlélete I.

- *adatrejtés:*  
Az operációs rendszer objektumai csak saját adataikat érhetik el.
- *interfész-használat:*  
Az objektumok formális interfészt használnak egymás elérésére.
- *hierarchikus objektum szerkezet:*  
kernel objektumok, executive objektumok.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

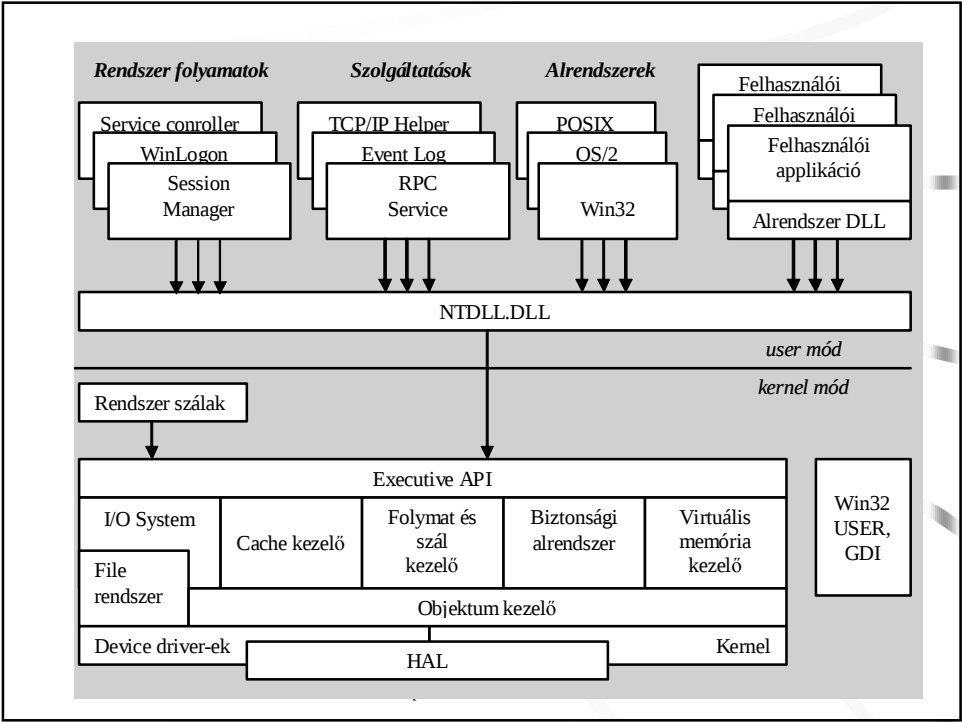
## Az NT objektum-orientált szemlélete II.

- A Windows NT **nem** valósítja meg a következő objektum-orientált tulajdonságokat:
  - *polimorfizmus:*  
Azonos néven különböző objektumok elérése.
  - *öröklődés:*  
Az objektumoknak hierarchikus származási rendszere.
  - *dinamikus adattípus-kötés:*  
Definiálhatóak adattípussal paraméterezett objektumok.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

# A Windows NT felépítés



## HAL (Hardware Abstraction Layer)

- A hardvert teljesen elfedő legalacsonyabb szoftver réteg.
- Az aktuális hardverre támaszkodva egy *virtuális gépet* valósít meg.
- Hardver elérés csak ezen keresztül.
- Processzortól független szolgáltatásokat ad a többi rétegnek.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## HAL II.

- Mindegyik processzorhoz külön HAL-réteg.
- Rendszer a telepítésekor választ HAL-t. (NT CD \i386-os alkönyvtára).
- Azonos architektúrájú processzorok esetén (pl. az x86 típusú processzorokon futó NT-k között) csak a HAL-ban van különbség.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Kernel

- Állandóan memóriában levő kernel (védett) módban futó komponens.
- A kernel a HAL-on kívül az egyetlen hardver-függő rész:
  - HAL a processzor típusától függ,
  - a kernel a processzor architektúrájától.
- A kernel által a nyújtott interface hardver független:
  - **a** a többi komponens hardver független.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## NT hordozhatósága

- Hordozható operációs rendszer megvalósítása:
  - Hardver független interfésszel rendelkező szoftver rétegek:
    - A HAL, ill. Kernel feletti rétegek.
  - Hordozható programnyelv használata:
    - C, C++.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## A Kernel funkciója

- Jól definiált működésű *alapmodulokat* (ún. *primitíveket*) biztosít az NT többi tagja számára.
- A primitívek a következő funkciókat valósítják meg:
  - *thread (szál) ütemezés, környezetváltás,*
  - *kivételkezelés, I/O-kezelés,*
  - *trap-kezelés, multiprocesszor-ütemezés,*
  - *kernel objektumok kezelése.*

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Kernel objektumok

- A kernel objektumok egyszerűek.
- Cél:
  - *gyors kezelés.*
- Megvalósítás:
  - *Nem végez ellenőrzést az egyes objektumok eléréskor, megbízik abban, hogy a rendszer tagjai helyesen használják az objektumokat.*

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Device driver-ek (készülékmeghajtók)

- Az I/O alrendszer és a hardver közötti kapcsolat.
- Hardver elérés: HAL-rétegen keresztül. ("Hordozható" device driver-ek.)
- A device driver-ek négy típusát különböztetjük meg:
  - Hardver driver-ek
  - Fájlrendszer driver-ek
  - Szűrő típusú device driver-ek
  - Hálózat elérését biztosító device driver-ek

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Device driver-ek típusai I.

- Hardver driver-ek (alacsony szintű):
  - hardver egységek elérése, eszközök közvetlen be- és kimeneteit kezelik.
- Fájlrendszer driver-ek: (magas szintű)
  - Fájlrendszerek elérését biztosító kéréseket fogadnak el és I/O kérésekké transzformálják azokat.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Device driver-ek típusai II.

- Szűrő típusú device driver-ek:
  - Többletfunkciókat megvalósítása (réteg szerkezetű device driver struktúra). A magasszintű driver-ek és alacsony szintű driver-ek közé ékelődnek.
- Hálózat elérését biztosító device driver-ek:
  - hálózati kéréseket szolgálják ki, ill. továbbítják azokat.

hétfő, 2005. december 5.

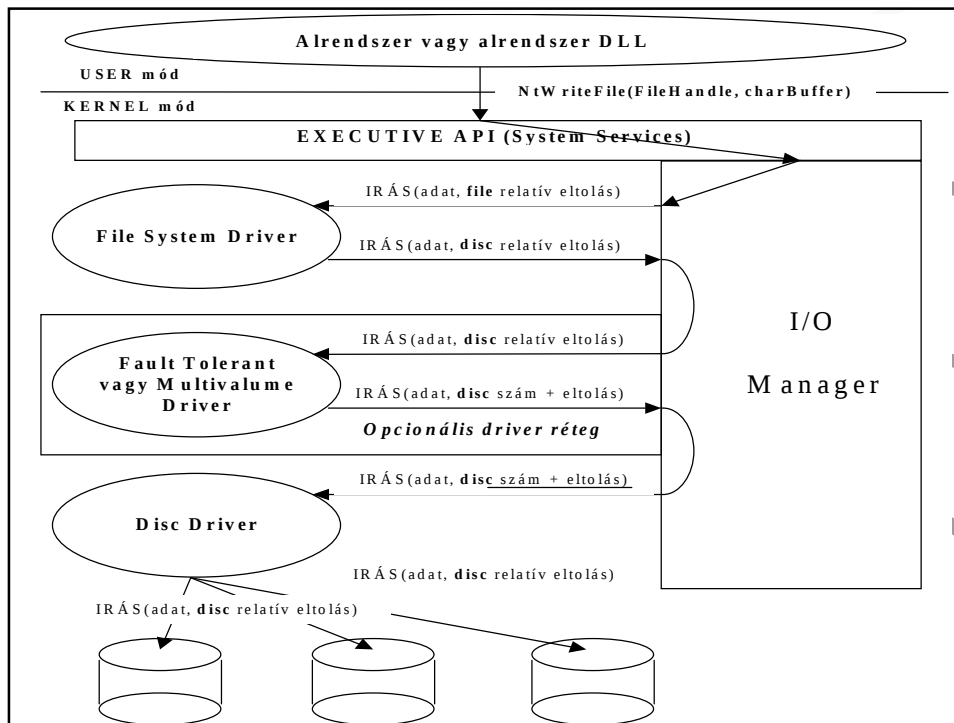
Dr. Benyó Balázs  
Operációs rendszerek III.

## Réteg szerkezetű device driver struktúra

- Különböző funkcionalitású driver-ek rugalmas használata.
- Igényeknek megfelelő rendszer építése.
- Különböző driver szintek:
  - A szintek funkcionalitása, ill. az adott funkcionalitáshoz tartozó interface jól definiált.
- Példa: Lemez írása.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.



## Executive

### Megvalósított függvények:

- elérhető alrendszerek hívások
- nem dokumentált alrendszerek hívások
- dokumentált driver hívások
- nem dokumentált belső (kernel, executive) hívások
- belső hívások

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Executive önálló részei

- *Folyamat és szál kezelő*
  - létrehozás, kezelés, leállítás
- *Virtuális memória kezelő*
  - memória gazdálkodás, cache kezelés támogatása
- *Biztonsági alrendszer (monitor)*
  - lokális gép objektum védelme
- *Cache kezelő*
  - file kezelő I/O gyorsítása
- *I/O rendszer kezelő*
  - eszköz független I/O

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Funkciói I.

- Az NT belső objektumai közötti kommunikáció biztosítása.
  - *LPC (Local Procedure Call, lokális eljárás hívás)*
    - NT IPC (Inter Process Communication) eszköze.
    - Egy user objektum egy másik user objektum függvényét hívhatja meg.
    - *PL.: Így érhetik el pl. az egyes alkalmazások a hozzájuk tartozó alrendszer szolgáltatásait.*

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Funkciói II.

- Az NTDLL.DLL által definiált függvények hívásainak megvalósítása.
- A *run-time library* függvények megvalósítása.
- Rendszer processzek és a szolgáltatások számára támogató funkciókat megvalósító függvények.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Rendszer processzek (rendszer folyamatok)

- Operációs rendszer funkciókat megvalósító önálló folyamatok.
- Felhasználói módban futó folyamatok.
- Rendszer processzek szükséges részei a rendszernek.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Fontosabb rendszer folyamatok I.

- **SMSS (Session Manager):**

A rendszer indításakor létrehozott folyamat, ami az applikációk elindításáért felelős.

Feladatai:

- egyes alrendszerek elindítása, ha futásukra szükség van,
- kapcsolat biztosítása a debugger és az általa futatott applikáció között
- biztosítja a környezeti változók definiálásának és elérésének lehetőségét

hétfő, 2005. december 5.

Operációs rendszerek III.

## Fontosabb rendszer folyamatok II.

- **Logon:**

- A felhasználók ki és beléptetését intéző folyamat.
- Minden ún. SAS (*Secure Attention Sequence*) billentyű kombináció (alapesetben: CTRL-ALT-DEL) aktivizálja.
- Beléptetéskor a user name és password kombinációt az önálló folyamatként futó *Local Security Authentication Server*-hez (LSASS) továbbítva ellenőrzi.
- Ha az azonosítás sikeres, elindítja a *USERINIT.EXE* programot, ami beállítja a felhasználó által definiált környezetet, és elindítja a user által kért shell-t. (alapesetben: *EXPLORER.EXE*)

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Fontosabb rendszer folyamatok III.

- *Service Controller*:
  - A szolgáltatások indításáért és leállításáért felelős folyamat.
  - user interface:
    - szolgáltatás állapota
    - szolgáltatás beállításai
    - szolgáltatás indítása/leállítása

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Szolgáltatások

- Szolgáltató folyamatok:
  - kliens-szerver működés
  - szolgáltató folyamatok pl.:
    - RPC (Remote Procedure Call) szolgáltató
    - NT specifikus eseménynaplózó (event logger) szolgáltató.
    - ...

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Szolgáltatások tulajdonságai

- Felhasználói módban futó folyamatok.
- A szolgáltatásokat megvalósító folyamatok futása nélkül is képes működni az NT.
- A szolgáltatások a Service Manager segítségével elindíthatók és leállíthatók a rendszer működése közben.
- Egyszerű Win32-es alkalmazások, de együttműködnek a *Service Controller (SERVICES.EXE)* folyamattal (regisztrál, indít leállít).
  - Elérés: *Control Panel* → *Services* ikonja

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## A szolgáltatások három elnevezése

- ahogy a szolgáltatás a Control Panel Services alpontján keresztül elérhető,
- ahogy Registry-ben szerepel,
- a szolgáltatást megvalósító futtatható program neve.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## NTDLL.DLL

- Dinamikusan linkelhető könyvtár (*Dinamically Linked Library, -DLL*).
  - Interfész a felhasználói módú folyamatok számára
  - Minden felhasználói objektum az NTDLL.DLL-en keresztül éri el a környezetét (pl. LPC).
- Működés:
  - Hívási paraméterek ellenőrzése.
  - User–kernel módváltás.
  - NT kért függvényének meghívása.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Alrendszerek I.

- Applikációk futtatása: alrendszerek segítségével:
  - Win32 (szükséges),
  - POSIX (opcionális),
  - OS/2 (opcionális).
- Alrendszerek feladata:
  - Alkalmazások futtatásához szükséges szolgáltatások nyújtása.
  - Alkalmazások kontrollálása.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Alrendszerek II.

- Az alkalmazások és az NT interfésze:  
**alrendszer DLL**  
(programozói interface (*API, Application Programming Interface*)).

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Alrendszerek programozói interfésze

- Jól definiált publikus interface; a többi interface (pl. NTDLL.DLL) nem publikus, nem dokumentált.
- Az egyes alrendszerekhez tartozó API-k lényegesen különböznek egymástól. (A legszélesebb a Win32 API.)
- API funkciók: pl. ablakozás, szálak kezelése.
- Az applikáció csak egy alrendszerhez tartozhatnak, nem keveredhetnek egymással egy applikáción belül különböző alrendszerekhez tartozó hívások.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## POSIX alrendszer I.

- Szigorúan a POSIX 1003.1-es szabványban rögzített tulajdonságokat valósítja meg.
- Lehetőség van:
  - fork-olásra,
  - hard link-ek definiálására,
  - interprocess kommunikációra,
  - folyamatok kezelésére, és
  - karakteres I/O kezelésére.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## POSIX alrendszer II.

- **Nincs** lehetőség
  - szálak (thread-ek) létrehozására,
  - ablakkezelésre,
  - RPC-elérésére,
  - socket-ek használatára.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Win32 alrendszer I.

- A Win32 alrendszer:
  - 32-bites alkalmazások,
  - 16-bites és
  - DOS-os alkalmazások futtatása.
- Nélküle nem futhat az NT.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Win32 alrendszer II.

Az alrendszer egy része kernel módban fut. (*WIN32K.SYS*):

- Grafikus képernyő-kezelési funkciók.  
(A *USER32.DLL*, *GDI.DLL*,  
*KERNEL32.DLL*, *ADVAPI.DLL*  
könyvtárakban definiált funkciók.)
- Rendszer gyorsítása miatt kerültek ide,  
módváltás nélkül lehetséges az executive,  
ill. kernel-szolgáltatások (függvények)  
elérése.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Win32 alrendszer III.

Grafikus eszközök és a printer kezelése szabványos felületen történik:

- A *GDI (Graphical Device Interface)*, ill. a hozzá tartozó *GDI32.DLL* (WIN32 API része) teszi lehetővé. Ezek is a WIN32 alrendszer kernel módú részében (*WIN32K.SYS*) vannak megvalósítva

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Win32 API-hívások megvalósítása I.

- Megvalósításuk helye szerint:
  - Az alrendszer DLL-ben:  
Egyszerű funkcionalitású függvények, a végrehajtásukhoz nincs szükség a rendszer más részeinek elérésére.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.

## Win32 API-hívások megvalósítása II.

- Az alrendszerben:  
A hívást az alrendszer DLL továbbítja az *NTDLL.DLL* felé, ami az Executive réteg LPC szolgáltatását igénybe véve, eljut a Win32 alrendszerhez, ami a kérést teljesíti.
- Más (kernel módban futó) rétegben:  
A hívás az Executive réteghez kerül, ami továbbítja a megfelelő kernel komponens felé.

hétfő, 2005. december 5.

Dr. Benyó Balázs  
Operációs rendszerek III.