A Windows operációs rendszer

Mérés laboratórium 4 (VIMIA315)

Mérési segédlet

Készítette: Micskei Zoltán

Utolsó módosítás: 2013. február 06.

Verzió: 1.2.1

Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

1 Bevezető

A mérés során a Windows operációs rendszer felépítésével és beépített eszközeivel fogunk megismerkedni. A mérés a felhasználói felület alapvető ismeretét feltételezi.

A felkészülést a mérésvezető minden alkalommal ellenőrzi. A mérést megelőző otthoni felkészülésként végezzük el az alábbiakat önállóan.

1. Ismételjük át az Operációs Rendszerek tárgy keretében a Windowsról tanultakat!
   a. Segédanyagok: http://mit.bme.hu/~micskeiz/opre
2. Olvassuk el a jelen segédletben szereplő áttekintőt[1]!
3. Válaszoljuk meg az Ellenőrző kérdéseket (a mérési leírás végén található)!
4. **Próbáljuk ki a következő eszközöket** (leírás az 5. fejezetben található)!
   a. *VMware Player*: indítsunk el egy virtuális gépet, és nézzük meg, hogy hogyan lehet képernyőképet készíteni és teljes képernyős módban használni.
   b. *Sysinternals Process Explorer*: nézzük meg a folyamat fa elemeit és egy folyamat tulajdonságait bemutató lapokat.

További információ a segédletben szereplő anyagról:

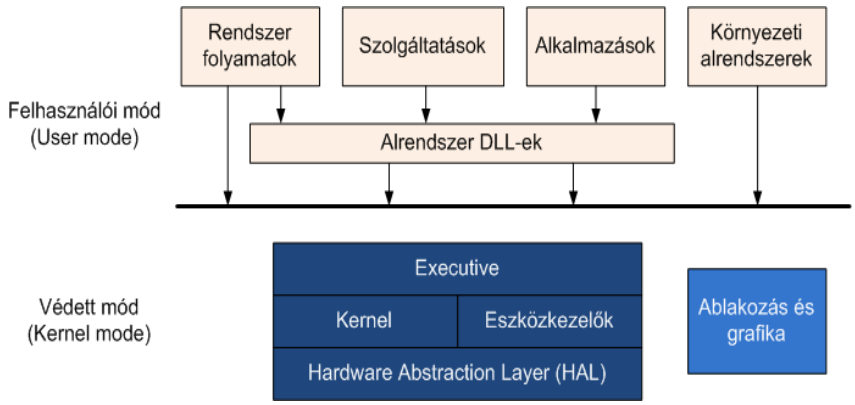
1. Kóczy Annamária, Kondorosi Károly (szerk.): Operációs rendszerek mérnöki megközelítésben, Panem, 2000. http://www.tankonyvtar.hu/hu/tartalom/tkt/operacios-rendszerek/adatok.html
   Operációs rendszerekkel kapcsolatos alapfogalmak (kernel, folyamat, virtuális memória stb.)
2. Gál Tamás, Szabó Levente, Szerényi László: Rendszerfelügyelet rendszergazdáknak, Szak Kiadó, 2007., elérhető online: https://technetklub.hu/Downloads/Browser.aspx?shareid=1&path=PDF
   Rendszermenedzsment eszközök (eseménynapló, szolgáltatások stb.), még Windows Vistához készült a könyv, de az alapok ugyanazok Windows 8 esetén is
3. Novák István *(szerk.)*, Windows 8 fejlesztés lépésről lépésre, JOS Kiadó, 2012, elérhető online: https://devportal.hu/Fajlok/Default.aspx?shareid=1&path=Konyvek
   Windows 8 bevezető, a mérés szempontjából a 2. és 3. fejezet nyújt segítséget

2 A Windows felépítése

Ez a rész áttekinti a Windows operációs rendszerek architektúráját és legfontosabb komponenseit.

2.1 A Windows operációs rendszer főbb komponensei

A következő ábra nagyvonalakban bemutatja a Windows felépítését.



1. ábra: A Windows felépítése

[1] A leírás Windows 8-hoz készült, régebbi Windows verziókban jó pár dolog máshogy volt.

A következő fő komponensekből áll a rendszer.

Védett módban futó komponensek:

- *HAL*: feladata, hogy elfedje a hardver jellegzetességeit, és a felette lévő rétegeknek egy egységes interfészt nyújtson az alacsonyszintű hardverrel kapcsolatos műveletekhez.
- *Kernel*: az operációs rendszer alapfunkcióit nyújtó komponense (pl. ütemezés, megszakításkezelés). Még ebben a részben is lehetnek hardver specifikus kódrészletek, hisz például a környezetváltás megvalósításához ismerni kell, hogy milyen regiszterei vannak a processzornak. Az ntoskrnl.exe fájl tartalmazza. (Szokás a kernel névvel az összes védett módú komponensre is együtt hivatkozni.)
- *Eszközkezelők (device driver)*: kernel módú modulok, melyek az általános kéréseket lefordítják a konkrét eszköznek szóló parancsokra. A Windowsban rétegzett struktúrájú eszközkezelő modell van, az egyes eszközkezelők láncot alkotnak (például az NTFS fájlrendszer és a merevlemez eszközkezelője közé beilleszthető egy modul, ami hibatűrést, különböző RAID struktúrákat, valósíthat meg transzparensen). Az eszközkezelők *sys* kiterjesztésű fájlok.
- *Executive*: az operációs rendszer magasabb szintű funkcióit szolgáltató rétege (memóriakezelés, biztonság stb.). Az adatokat objektumokban tárolja, melyeket *leíró*kkal (handle) lehet csak elérni, jól definiált interfészeken keresztül. Minden objektumhoz biztonsági információ is tartozik, hogy azt ki érheti el és milyen műveleteket hajthat rajta végre, ezeket minden eléréskor ellenőrzi is a rendszer. Bár a kernel funkcióit csak a kernel interfészén keresztül éri el, szintén az ntoskrnl.exe tartalmazza.
- *Ablakozás és grafika*: teljesítmény okokból az ablakozó és grafikus alrendszer átkerült kernel módba. Így kevesebb módváltás kell, hisz a képernyőre való rajzoláskor a végén biztos kernel módban lévő kódnak is kell futnia, mert az éri el a grafikus kártyát csak. A win32k.sys tartalmazza az idetartozó függvényeket.

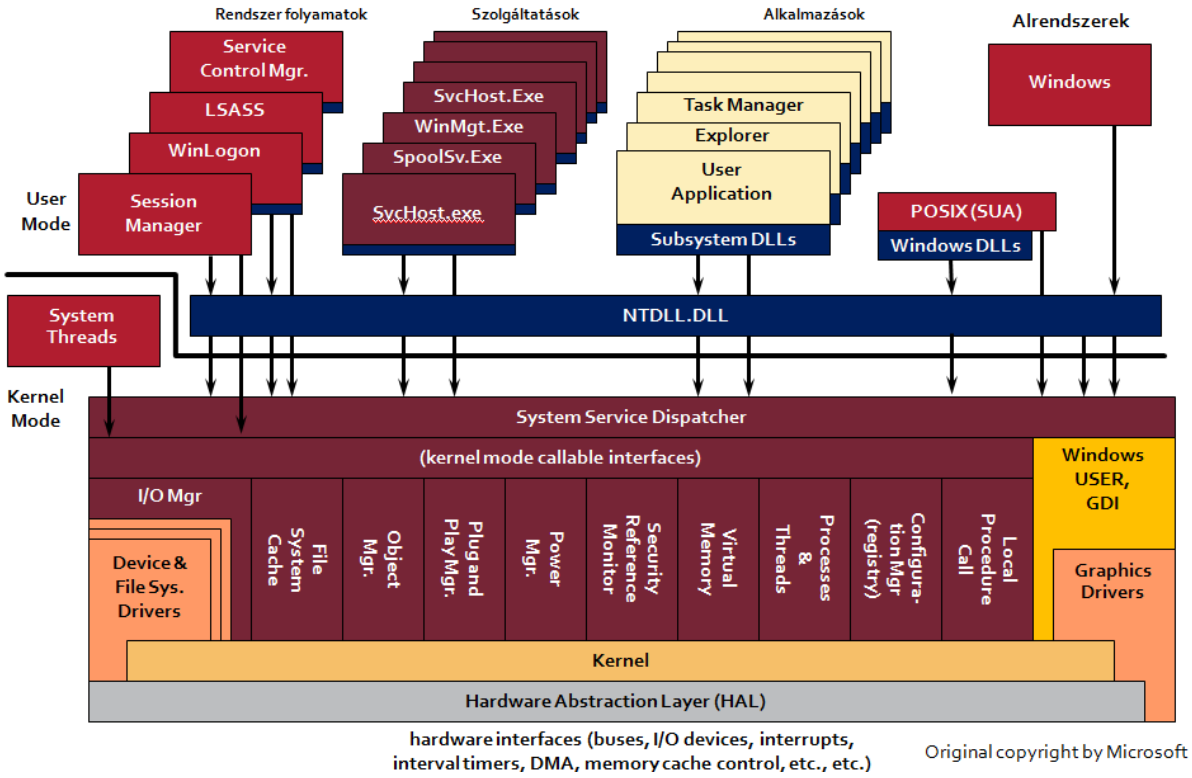
Felhasználói módban futó komponensek:

- *Rendszer folyamatok*: olyan folyamatok, amik felhasználói módban futnak, de a rendszer elindulásához, működéséhez nélkülözhetetlenek.
- *Szolgáltatások (service)*: olyan folyamatok, amik a felhasználói felülettől és belépéstől függetlenül a háttérben futnak. Több mint ötven beépített szolgáltatás van, ezek feladatai a tűzfal kezelésétől a nyomtatási sorig sok mindenre kiterjednek. Saját szolgáltatásokat később más program is telepíthet.
- *Környezeti alrendszerek (environment subsystem)*: A Windows NT megjelenésekor három féle programozói felületet nyújtott: POSIX, Win32 és OS/2 API-n keresztül is elérhették az alkalmazások az operációs rendszer hívásait. Ezt úgy valósították meg, hogy a kernel (pontosabban az Executive) egy közös, belső API-t ajánl ki, ezt használják az alrendszerek. Az alkalmazások pedig nem az Executive-ot hívják közvetlenül, hanem valamelyik alrendszeren keresztül használják annak a funkcióit.

A következő ábra (2. ábra) kicsit részletesebben mutatja be az egyes komponenseket. A következőket érdemes megfigyelni az ábrán:

- Látszanak az Executive egyes fontosabb komponensei (pl., Virtual Memory, I/O Manager).
  - Az ezekben megvalósított funkcióknak csak egy része érhető el felhasználói módból. Ezeknek a megfelelői az NTDLL.DLL fájlban találhatóak. Ez végzi el a hívások fogadását, paraméterek ellenőrzését, majd átváltást kernel módba és a hívások továbbítását a System Service Dispatchernek.
- Fel vannak sorolva az alrendszerek:
  - Windows alrendszer (csrss.exe),
  - Az opcionális POSIX alrendszer (psxss.exe): Ennek az aktuális neve Windows Vista óta Subsystem for Unix Applications (SUA).
  - Egyedül a Windows alrendszer és a Session Manager hívja közvetlenül az NTDLL.DLL-t, az összes többi komponens valamilyen alrendszer DLL-en keresztül éri el az operációs rendszert. Ezek a Windows alrendszer esetén a kernel32.dll, advapi32.dll, user32.dll és a gdi32.dll fájlok.
- Szolgáltatások:

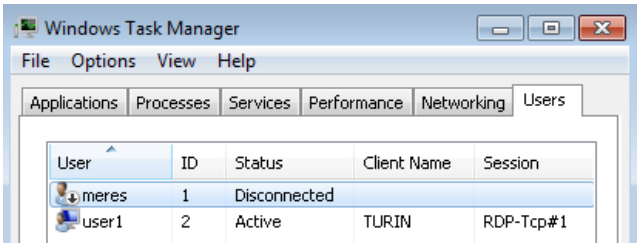
- *SvcHost.exe* (Host Process for Windows Services): általános szolgáltatás futtató folyamat a beépített szolgáltatásoknak. Szerepe az, hogy erőforrásokat takarít meg azzal, hogy nem fut minden szolgáltatás külön folyamatban. Csak beépített szolgáltatásokat futtathat, saját szolgáltatást nem lehet belerakni. Az egyes svchost.exe példányok egy úgynevezett csoportnévvel vannak elkülönítve egymástól (pl. NetworkService, LocalServiceNoNetwork), ami jelzi, hogy milyen felhasználó nevében futnak és milyen egyéb korlátozás vonatkozik rájuk. Vista óta már a Feladatkezelő is meg tudja jeleníteni a Szolgáltatások fülön, hogy melyik svchost.exe-nek mi a csoport neve és mi fut benne.



2. ábra: A Windows felépítése részletesebben

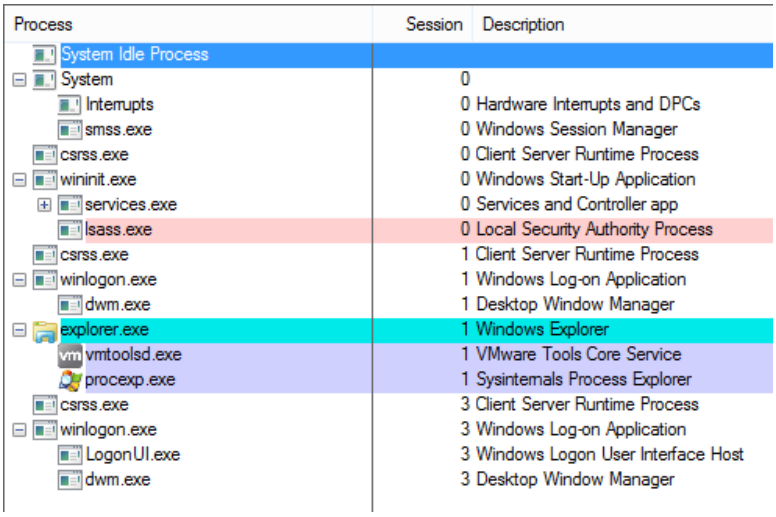
- Rendszer folyamatok:
  - Session Manager (smss.exe): Ez az első felhasználói módú folyamat, ami elindul a rendszerben. Egy gépre több felhasználó is be lehet jelentkezve, az ő folyamataikat úgynevezett *munkamenet*ekben (session) különíti el a rendszer. Az smss.exe ezen munkamenetek elindításáért és kezelésért felelős. A munkamenetek lehetséges fajtái (3. ábra):
    - Session 0: a rendszerfolyamatok és szolgáltatások számára fenntartott munkamenet. Nincsen felhasználói felülete.
    - Console (Session 1): a helyileg bejelentkezett felhasználó munkamenete.
    - Távoli munkamenetek: kliens operációs rendszereken a *Távoli Asztal* (Remote Desktop) segítségével lehet egy távoli munkamenetet nyitni a *Távoli Asztal Kapcsolat* (mstsc.exe) program segítségével. Ilyenkor viszont a konzol munkamenetet kijelentkezteti ideiglenesen. Szervereken alapbeállítás szerint futhat a konzol munkamenet mellett még két távoli, adminisztrációs célokra fenntartott kapcsolat. Ahol telepítve van a Remote Desktop Services komponens, ennél több felhasználói munkamenet is futhat párhuzamosan.
  - *Windows Startup Application (Wininit.exe)*: Az smss.exe indítja el miután létrehozta a 0-s munkamenetet. Feladata a többi rendszerfolyamat elindítása.
  - *Local Security Authority Subsystem (lsass.exe)*: A főbb biztonsági feladatokat ez a folyamat végzi, úgymint felhasználók és jelszavak ellenőrzése, biztonsági naplózás, hozzáférések ellenőrzése.
  - *Service Control Manager (services.exe):* A szolgáltatások kezelésért (elindítás, leállítás, stb.) felelős program. Elindítja az automatikusan induló szolgáltatásokat.
  - *Local Session Manager (lsm.exe):* Feladata a távoli kapcsolatok kezelése.

- *Windows Logon Application (Winlogon.*exe): a felhasználók bejelentkeztetéséért felelős.
- *Windows Logon User Interface Host (LogonUI.exe)*: Ez a folyamat jeleníti meg a bejelentkező képernyőt, és ajánlja fel a különböző bejelentkezési módokat (jelszó, smart card), attól függően, hogy mi van telepítve a rendszerre.
- *Explorer*.exe: A felhasználói grafikus shell, alapértelmezés szerint sikeres belépés után ezt indítja el a winlogon (igazából már nem szoros értelemben vett rendszer folyamat).



3. ábra: A konzol munkamenet jelenleg szétkapcsolt állapotban van, mert a távoli az aktív.

A következő ábrán láthatunk egy képet, hogy hogyan néz ki a folyamat hierarchia (4. ábra). Ahogy a Session oszlop értékeiből is látszik, két munkamenet fut a gépen (a 3-asnál fut a LogonUI.exe, tehát a bejelentkező képernyőt mutatja ott a rendszer).



4. ábra: Rendszerfolyamatok két munkamenet esetén

A folyamatok a rendszer erőforrásait, legyen akár az egy fájl, folyamat vagy megosztott memória, úgynevezett leírókon (handle) keresztül érik el. A leíró megnyitása előtt ellenőrzi a rendszer, hogy van-e az adott felhasználónak joga hozzáférni ahhoz az objektumhoz.

Ezzel nagyjából áttekintettük, hogy mik a fontosabb operációs rendszer komponensek Windows esetén.

2.2 Windows Store alkalmazások

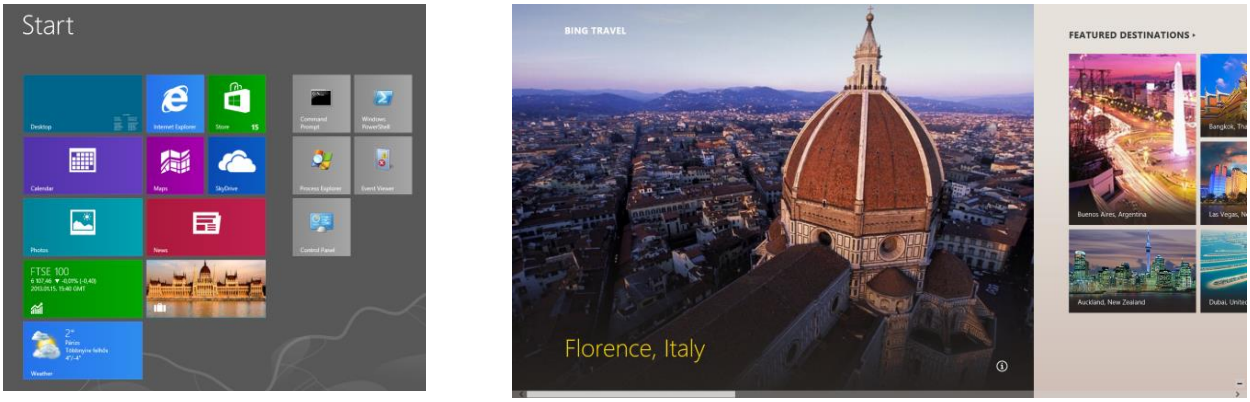
A Windows 8 egyik legjelentősebb változtatása az új típusú, úgynevezett Windows Store alkalmazások bevezetése (korábban Metro stílusú alkalmazások volt a kódnevük). Ezek egy újfajta felhasználói felületet kaptak, az eddigiektől eltérő módon kell telepíteni és az életciklusukat kezelni, továbbá új fejlesztői környezet is tartozik hozzájuk. Az új típusú alkalmazások néhány jellegzetessége:

- új típusú felhasználói felület (pl., teljes képernyős megjelenítés, érintésvezérlésre felkészülés),
- más telepítési modell (pl., az alkalmazás egy csomagban érkezik, amit tipikusan a Windows Store-ból szerezhetünk be, ezt később innen is lehet frissíteni),
- elszigetelt környezet (pl., erősen korlátozva van, hogy mihez férhet hozzá),

- speciális életciklus (az OS felfüggeszthet háttérbe került alkalmazásokat, csak külön hívások segítségével lehet háttérműveleteket végezni).

Az új modell tervezési céljai között szerepelt, hogy könnyebb legyen táblagépeken használni, bánjon gazdaságosabban a rendszer erőforrásaival (pl., kevesebbet fogyasszon), kevesebb lehetőség legyen a meghibásodásra, egyszerűbb legyen kezelni stb.

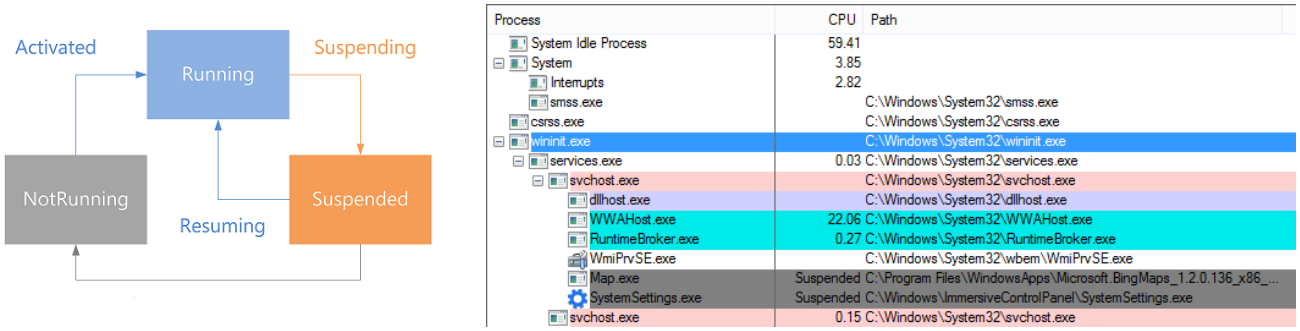
Az új felületen való navigáláshoz kezdetben azt érdemes tudni, hogy a képernyő sarkaiba mozgatva a kurzort érhetjük el a következő funkciókat. Bal alul: kezdőképernyő megjelenítése, bal felül: átváltás másik futó Windows Store alkalmazásra, jobb alul vagy felül: az úgynevezett gombsor (charm bar) megjelenítése. A képernyő szélét a mérésen is használt virtuális gépes környezetben elsőre nehézkes eltalálni, ezért érdemes inkább gyorsbillentyűket használnunk (lásd a függeléket).



5. ábra: Az új kezdőképernyő és egy Windows Store alkalmazás felülete (Travel)

Az alkalmazások életciklusának főbb elveit szemléltetik az alábbi ábrák (6. ábra). Ha egy futó (running) alkalmazásról átváltunk másikra, akkor annak a futását az OS pár másodpercen belül felfüggeszti (suspend). Erről értesíti az alkalmazást, aki kap legfeljebb 5 másodpercet, hogy elmentse az állapotának fontos részét. Felfüggesztett állapotban az alkalmazás nem kap CPU-időt, azonban a memóriatartalma még megmarad. Ha kevés lenne a rendszerszintű szabad memória, akkor az OS akár be is zárhatja a felfüggesztett alkalmazásokat (terminate). Fontos, hogy ilyenkor az alkalmazás már nem kap külön értesítést, tehát ha a felfüggesztés során nem mentette el az állapotát, akkor erre ilyenkor már nincs lehetősége. Ha újra elindítunk egy bezárt alkalmazást, és az korábban elmentette az állapotát, akkor az aktiváláskor azt vissza tudja tölteni.

A jobb oldali ábra egy olyan helyzetet mutat, ahol több Windows Store alkalmazást is elindítottunk. Figyeljük meg, hogy ezeket nem az explorer.exe indítja el, hanem egy svchost.exe-ben futó háttérszolgáltatás. Elindította a felhasználó az új felületre átírt rendszerbeállításokat (SystemSettings.exe), a Térkép/Map alkalmazást (Map.exe, figyeljük meg az elérési útját is!) és a WWAHost.exe-ben futó Utazás/Travel alkalmazást. Ezek közül kettő fel van már függesztve, az Utazásról pedig most váltottunk át, tehát azt nemsokára felfüggeszti a Windows.

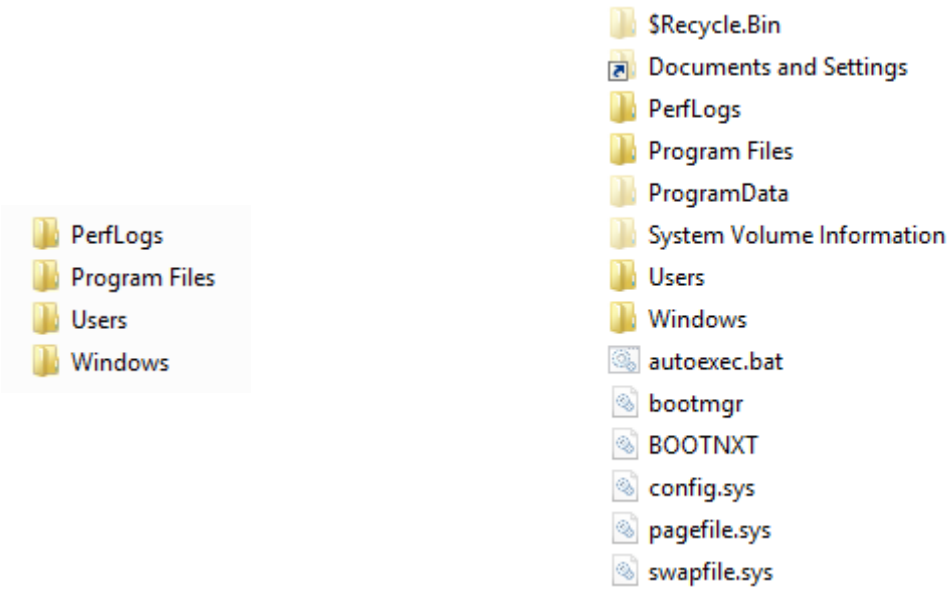


6. ábra: Windows Store alkalmazások életciklusa

A Windows Store alkalmazásoknak azonban ezzel épp csak a felszínét érintettük. Érdeklődőknek a bevezetésben említett könyvet vagy az MSDN dokumentációt tudjuk javasolni.

2.3 Rendszerkönyvtárak

Egy frissen feltelepített rendszeren a következő könyvtárakat találjuk (7. ábra). Ezek szerepe:



7. ábra: Telepítéskor létrejövő könyvtárak (rejtett fájlok megjelenítése nélkül és megjelenítésével)

- *$Recycle.Bin*: Lomtár funkció megvalósítása, benne felhasználókként külön könyvtárban tárolódnak a törölt fájlok és mappák.
- *Boot*: A rendszerindítás beállításait tartalmazza (BCD, Boot Configuration Database). Friss telepítés esetén ez és a Recovery könyvtár egy külön 350 MB-os rendszer partícióra kerülnek.
- *Users* (Felhasználók[2]): A felhasználók profiljait (beállításait és adatait) tárolja itt a rendszer. Régebbi rendszereken *Documents and Settings* volt a neve, kompatibilitás miatt az újabb verziókon is létrejön egy ilyen nevű link[3], ami a Users mappára mutat.
- *Program Files*: A feltelepített programok alapértelmezett helye. Fontos, hogy ide a felhasználóknak csak olvasási joga van, hogy ne tudják felülírni potenciálisan veszélyes változatokkal a programokat. Így ide az alkalmazásoknak nem szabad írnia normális futás során beállításokat vagy adatokat, csak telepítéskor szabad módosítaniuk ennek a könyvtárnak a tartalmát. Ha 64 bites verziót telepítünk, akkor létrejön egy külön `Program Files (x86)` könyvtár is.
- *ProgramData*: az alkalmazások közös, tehát nem felhasználó specifikus adatai kerülnek id. Az itt lévő könyvtárak egy része csak link a `\Users\Public` megfelelő mappájára.
- *System Volume Information*: a *Rendszervédelem* (System Restore) funkció itt helyezi el az elmentett visszaállítási pontokat többek között.
- *Windows*: a rendszerhez tartozó programok, könyvtárak, beállítások, stb. Néhány fontosabb alkönyvtára:
  - *Installer:* Feltelepített alkalmazásokhoz tartozó MSI fájlok a módosításhoz és eltávolításhoz.
  - *SoftwareDistribution:* a letöltött frissítéseket ideiglenesen itt tárolja a Windows.
  - *System32*: a rendszerhez tartozó programok és dll-ek nagy része itt van.
    - *drivers*: a rendszer által ismert eszközkezelők.
      - *etc*: itt található a *hosts* fájl, amiben statikusan lehet IP címekhez neveket rendelni.

[2] Bár magyar verzió grafikus felületén Felhasználókat ír ki a Windows, figyeljük meg, hogy parancssorból nézve a magyar nyelvű verzión is Users a könyvtár neve. Az alap kód nyelv semleges, és utána csak a felhasználói felületen jelennek meg honosított nevek.
[3] Az NTFS is ismeri a *szimbolikus link* fogalmát, ilyet például az *mklink* paranccsal lehet létrehozni.

    - *LogFiles*: néhány rendszer komponens ide rakja a napló fájljait, ha nem az eseménynaplót használja.
  - *Winsxs*[4]: az úgynevezett komponens tár, az egyes komponensek különböző verziói tárodnak itt.

A gyökérkönyvtárban lévő fájlok:

- autoexec.bat és config.sys: A jó öreg DOS-os idők hagyatéka.
- bootmgr: Feladata a BCD beolvasása és a boot menü megjelenítése. A rendszer betöltését már nem ez végzi, az `a %SystemRoot%\System32\Winload.exe` feladata.
- hiberfil.sys: Hibernálás során ebbe a fájlba menti el a memóriatartalmat a rendszer.
- pagefile.sys: A lapozófájl, ide pakolhat ki az operációs rendszer memóriatartalmakat ideiglenesen.

A Users könyvtár felépítését érdemes még megvizsgálni. Itt tárolódnak a felhasználók profiljai, mindegyik egy-egy külön könyvtárban. Telepítéskor létrejövő profilok:

- *Default*: új felhasználó első belépésekor ezt a profilt másolja le neki a rendszer.
- *Public*: Az itt lévő Asztal és Start menü mappában lévő elemek megjelennek minden felhasználónak. A többi alkönyvtár közös fájlok tárolására szolgál.
- *Administrator* (Rendszergazda): A rendszergazda felhasználó profil könyvtára.

Kompatibilitás miatt létrejönnek az All Users (→ \ProgramData) és Default User (→ Default) linkek is.

Egy profilban általában a következő elemek találhatóak:

- *AppData*: az alkalmazások felhasználó specifikus beállításai. Hálózati környezetben lehet úgynevezett *vándorló profilokat* használni (roaming profile), ilyenkor a profil a szerveren tárolódik, belépéskor letöltődik a helyi gépre, kilépéskor pedig a változásokat visszamenti a rendszer. Így a felhasználó bármelyik gépről ugyanazt a környezetet látja. Ennek megfelelően az AppData könyvtárnak több alkönyvtára van.
  - *Local, LocalLow*: Olyan beállítások kerülnek ide, amiket nem kell belerakni a vándorló profilba, tipikus például Temp könyvtár vagy a Temporary Internet Files.
  - *Roaming*: Azok az alkalmazás beállítások, amiket a felhasználó minden gépen el kell, hogy érjen.
- *Desktop*: Az Asztalon lévő fájlok és könyvtárak.
- *Contacts, Documents, Downloads, Favorites, Links, Music, Saved Games, Searches, Videos*
- *ntuser.dat*: a rendszerleíró adatbázisban tárolt felhasználó-specifikus beállításokat tartalmazó fájl.

2.4 Szolgáltatások

A szolgáltatások a háttérben, a bejelentkezett felhasználótól függetlenül futó folyamatok. Kezelésükre a *Vezérlőpult / Rendszer és karbantartás / Felügyeleti eszközök / Szolgáltatások* (Control Panel / System and Security / Administrative Tools / Services) panel való.

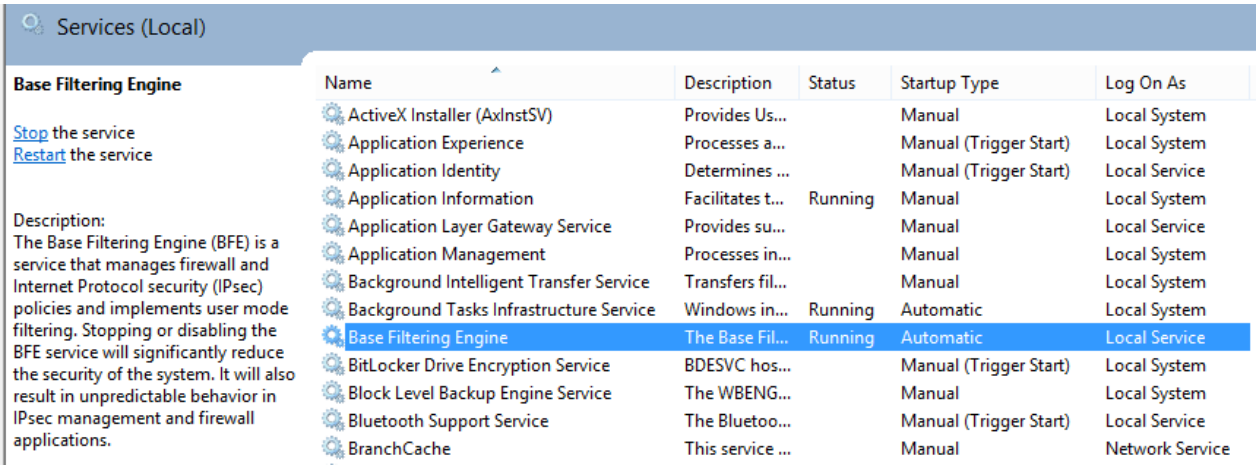
Az áttekintő képernyőn a következő információk látszanak a szolgáltatásokról.

- *Név*: ez a szolgáltatás felületen megjelenő neve.
- *Leírás*: rövid leírás, hogy mire jó a szolgáltatás.
- *Állapot*: elindítva / nincs elindítva.
- *Indítási típus*: Kézi / Automatikus / Letiltva. Lehetséges még késleltetett (delayed) indítás, az ilyen szolgáltatások indításával megvárja a rendszer, hogy az összes automatikus szolgáltatás elinduljon. Így ezek nem lassítják a felhasználók bejelentkezését. Az újabb verziókban bevezették annak a lehetőségét, hogy egy szolgáltatás csak adott típusú eseményekre induljon el (trigger).
- *Bejelentkezés mint*: ki az a felhasználó, akinek a nevében fut a szolgáltatás. Lehetőségek:
  - *Helyi rendszer (Local System):* még a rendszergazdánál is nagyobb joggal rendelkező, interaktív bejelentkezésre nem képes felhasználó. Tulajdonképpen minden helyi jogosultsága megvan (hozzáfér más felhasználó folyamataihoz, biztonsági adatbázis olvasása stb.).

[4] Bővebb információ: Engineering Windows 7, http://blogs.msdn.com/e7/archive/2008/11/19/disk-space.aspx

- *Helyi szolgáltatás (Local Service)*: a Helyi rendszernél gyengébb, de azért még mindig elég sok joggal rendelkező felhasználó.
- *Hálózati szolgáltatás (Network Service)*: A három közül a legkevesebb jogosultsággal rendelkező felhasználó.

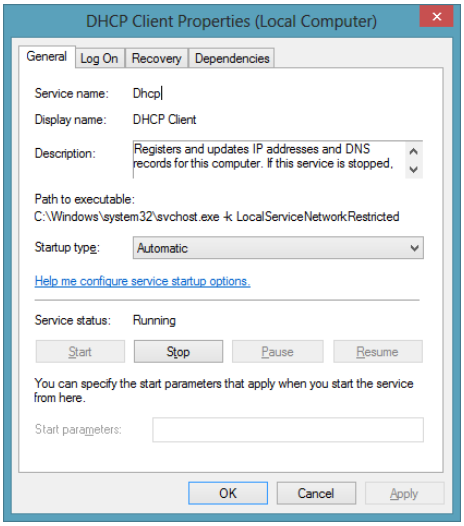
Saját felhasználó: lehetőség van tetszőleges, a gépen létező felhasználó kiválasztására is, csak ilyenkor meg kell adni neki a *Bejelentkezés szolgáltatásként* (Log on as a service) jogot.



8. ábra: Szolgáltatások kezelőpanel

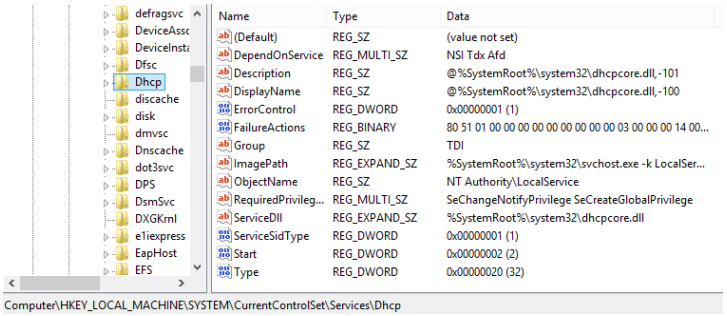
Az egyes szolgáltatásokról sok minden kiderül még a tulajdonságlapjukat megnyitva.

- *Általános* fül:
  - *Szolgáltatásnév*: itt látszik a szolgáltatás belső neve.
  - *Futtatható fájl*: milyen program valósítja meg a szolgáltatást. A képen egy beépített szolgáltatást láthatunk, amit az scvhost.exe futtat, annak is a LocalServiceNetworkRestricted példánya.
- *Bejelentkezés* fül: itt választhatnánk ki, hogy ki futtassa a szolgáltatást.
- *Helyreállítás* fül: mi történjen, ha leáll a szolgáltatás (pl. szolgáltatás újraindítása, egyéb program futtatása).
- *Függőségek* fül: mik azok a szolgáltatások, amiket meg kell várni, hogy elinduljanak, mielőtt a rendszer elindítja ezt a szolgáltatást, és mik azok, amik ettől a szolgáltatástól függnek. Ezeket az értékeket a szolgáltatás telepítésekor kell megadni.



9. ábra: Egy szolgáltatás tulajdonságai

A szolgáltatások a beállításaikat a *rendszerleíró adatbázisban* (registry) tárolják, a `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services` kulcs alatt.

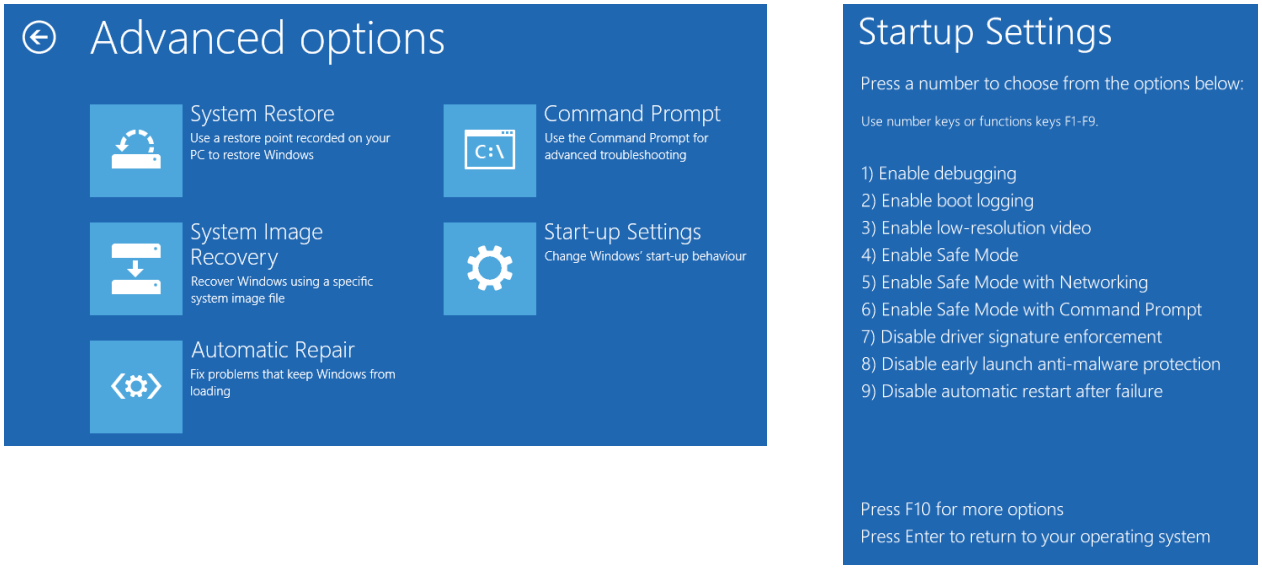


10. ábra: Szolgáltatás adatai a registry-ben

A szolgáltatásokat parancssorból is kezelhetjük, erre szolgál pl. a *Get-Service* PowerShell cmdlet.

2.5 Speciális rendszerindítási módok

Windows 8 esetén áttervezték a korábbi boot folyamatot és a speciális rendszerindítási módok elérhetőségét[5]. Korábban az indulás után kellett az F8 billentyűt megnyomni, azonban az újabb rendszerek túl gyorsan indulnak már el, így ez a megoldás nem lenne megbízható[6]. Ezért Windows 8 esetén a rendszer elindulása után a *Speciális rendszerindítás beállításai (Advanced Startup Options)* résznél tudjuk kiválasztani, hogy újraindítás után jelenítse meg a rendszerindítási menüt (11. ábra). Természetesen ha a rendszer nem tud normál módban elindulni, akkor automatikusan ez a menü jelenik meg.



11. ábra: Speciális rendszerindítási módok

A *csökkentet mód* (Safe Mode) esetén a rendszer csak az alap, beépített meghajtókat tölti be. Így ha normál módban nem indul el a rendszer, például egy hibás meghajtó vagy automatikusan induló szolgáltatás miatt, akkor csökkentett módban eltávolíthatjuk azt.

[5] Bővebben lásd: Building Windows 8 blog, Reengineering the Windows boot experience, URL: http://blogs.msdn.com/b/b8/archive/2011/09/20/reengineering-the-windows-boot-experience.aspx
[6] Bővebben lásd: Building Windows 8 blog, Designing for PCs that boot faster than ever before, URL: http://blogs.msdn.com/b/b8/archive/2012/05/22/designing-for-pcs-that-boot-faster-than-ever-before.aspx

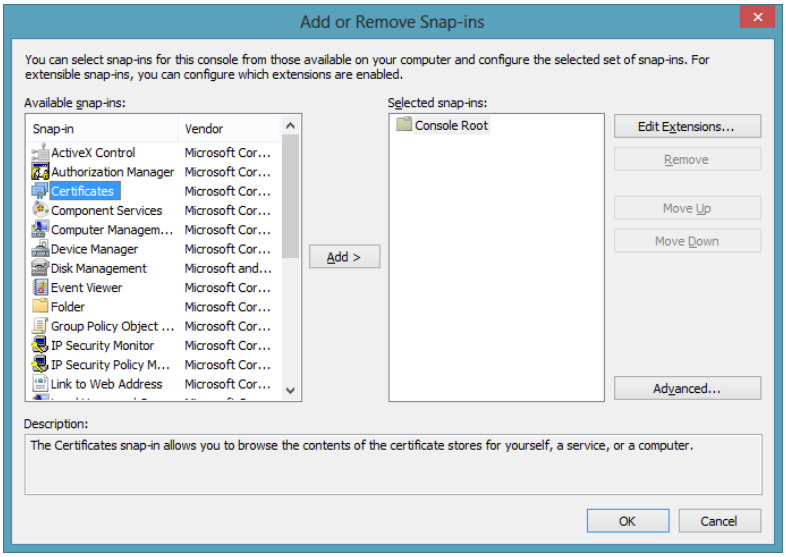
3 Rendszermenedzsment

A következőkben áttekintjük a rendszer működtetéséhez és karbantartásához használható legfontosabb technológiákat és eszközöket. Ezek egy részével már biztos mindenki találkozott, azonban nem árt később kicsit részletesebben is megismerkedni velük.

3.1 Microsoft Management Console

A legtöbb menedzsment eszköznek hasonló felülete van, ugyanis mindegyik általában csak egy-egy modul egy közös konzolhoz, a *Microsoft Management Console-hoz* (MMC). A konzol három fő részből áll, a baloldalon egy fa szerkezetben látjuk a különböző modulokat és azok részeit, a középső részen az adott modul részletei láthatók, a jobb oldalon pedig az a kiválasztott elemekhez tartozó műveletek sorakoznak.

Az MMC elindítása után (mmc.exe) a Ctrl+M segítségével adhatunk hozzá beépülő modulokat (12. ábra).



12. ábra: Beépülő modul hozzáadása MMC-ben

A modul hozzáadása után ki kell választani, hogy a saját vagy egy távoli gépet akarunk menedzselni, a legtöbb feladat távoli gépen is elvégezhető.

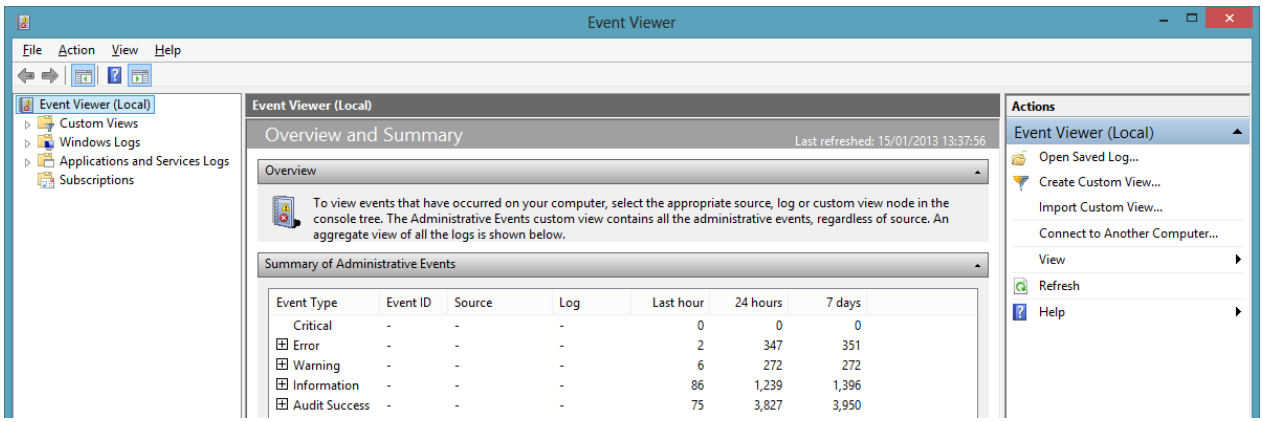
3.2 Eseménynapló

Az *eseménynapló* (Eventlog) a Windows központi naplózó eszköze, elérhető a *Vezérlőpult / Rendszer / Felügyeleti eszközökről* vagy közvetlenül hozzáadható egy MMC konzolhoz. Induláskor egy áttekintő nézet fogad az utóbbi idő legfontosabb eseményeiről, csoportosítva súlyosság szerint (13. ábra).

A fontosabb beépített naplók (Windows-naplók rész):

- *Rendszer*: rendszerüzenetek, pl. elindulás, szolgáltatások problémái.
- *Biztonság*: biztonsági audit napló, ki milyen objektumokhoz fért hozzá, milyen jogokat használt. Csak megfelelő jogosultsággal lehet megnézni a tartalmát.
- *Alkalmazás*: az egyéb beépített és külső alkalmazások írhatják ide az üzeneteiket. A Vista óta az alkalmazások egy jelentős része (pl., mentés, feladatütemező, stb.) saját naplót kapott, és az *Alkalmazás- és szolgáltatásnaplók* rész alatt található.

A Vista óta megváltozott az eseménynapló, új formátumban tárolja az eseményeket (XML alapú, *evtx* kiterjesztés), szűrhető és kereshető lett, a *Nézet* menüben pedig lehet rendezni és csoportosítani az eseményeket. Továbbá lehetőség van, hogy egy gépen gyűjtsük más gépek naplóit is (*Előfizetések* rész).



13. ábra: Az Eseménynapló áttekintő képe

A következő példán keresztül láthatjuk, hogy egy naplóbejegyzés általában milyen mezőket tartalmaz.

Napló neve:	System
Forrás:	Microsoft-Windows-Dhcp-Client
Dátum:	2008.02.03. 9:37:30
Eseményazonosító:	1000
Feladatkategória:	Nincs
Szint:	Hiba
Kulcsszavak:	Klasszikus
Felhasználó:	n.a.
Számítógép:	meres-vm
Leírás:	A számítógép elvesztette a(z) 000C29AD4B05 hálózati című hálózati kártyához rendelt IP-cím (192.168.132.138) címbérletét.

A hibakereséshez az esemény forrását és az esemény azonosítóját nézzük meg, és ez alapján keressünk rá például a http://support.microsoft.com oldalon lévő hivatalos tudásbázis cikkekben, vagy a http://eventid.net oldalon. A másik lehetőség, hogy az esemény tulajdonságainál az Online súgó linkre kattintunk, és ilyenkor a Microsoft webes adatbázisában rákeres erre a bejegyzésre, a gyakori hibákhoz szokott itt is megoldás lenni.

A naplófájlok helyileg a `C:\windows\System32\winevt\Logs` könyvtárban találhatóak. Érdemes megnézni, hogy hányfajta alkalmazás naplófájlja található itt.

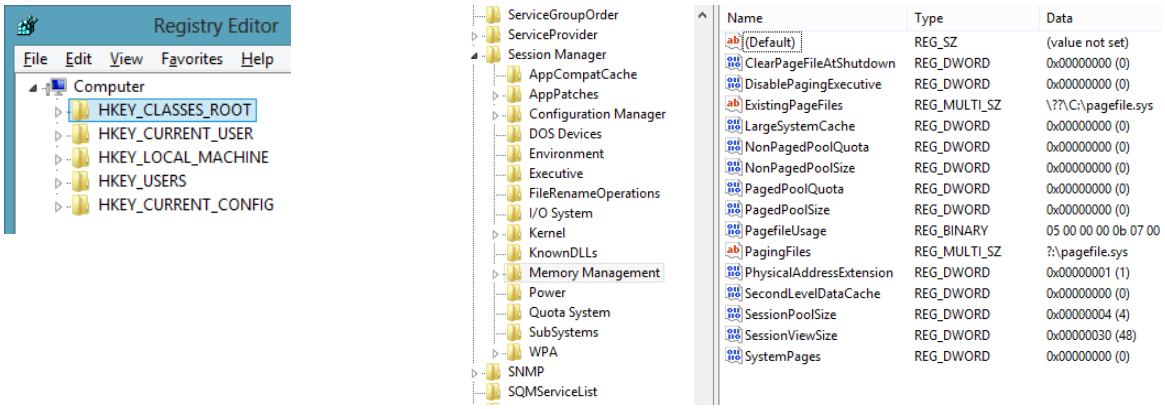
3.3 Rendszerleíró adatbázis

A *rendszerleíró adatbázis* (registry) a Windows központi konfigurációs adatbázisa. Mind a hardvereszközök, az operációs rendszer és az alkalmazások is itt tárolják (tárolhatják) beállításaikat. A rendszerleíró adatbázis fa struktúrájú, a csomópontokat *kulcsoknak* hívják, a kulcsokhoz különböző típusú tulajdonságokat rendelhetünk.

A legfelső szinten a következő kulcsokat találjuk (14. ábra):

- HKEY_CLASSES_ROOT: A különböző fájlkiterjesztésekhez tartozó beállításokat és a rendszeren lévő COM komponensek listáját tárolja. A COM (Component Object Model) technológia lényege, hogy az alkalmazások egyes részeiket komponensként kiajánlhatják, amiket más programok meghívhatnak, felhasználhatnak. Az egyes komponenseket egy CLSID (Class ID) nevű GUID (Globally Unique Identifier) azonosítja.
- HKEY_CURRENT_USER: az aktuális felhasználó beállításainak a rendszerleíró adatbázisban tárolt része. Csak egy link a HKEY_USERS megfelelő bejegyzéseire.
- HKEY_LOCAL_MACHINE (HKLM): a számítógép beállításai, a detektált hardver eszközöktől kezdve a telepített szoftverek listájáig.
- HKEY_USERS: a rendszeren lévő felhasználók beállításai.

- HKEY_CURRENT_CONFIG: az aktuális hardver profil adatai, csak link a `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current` kulcsra.

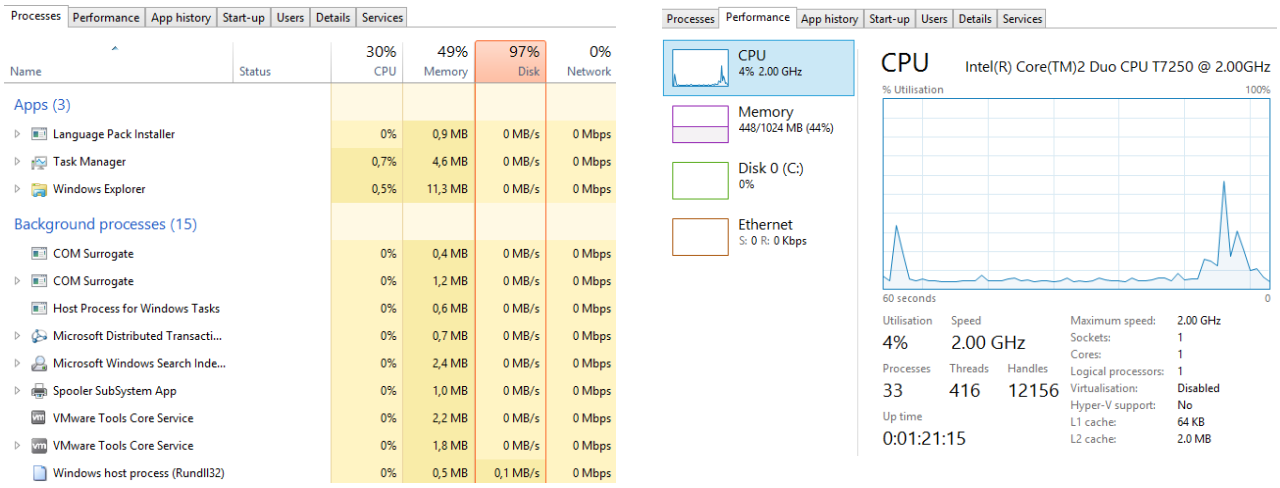


14. ábra: a) a rendszerleíró adatbázis legfelső szintű kulcsai és b) egy kulcshoz tartozó tulajdonságok

A rendszerleíró adatbázis helyileg a `\Windows\System32\config` könyvtárban lévő fájlokban található, szerkesztéséhez a *regedit* program használható.

3.4 Feladatkezelő

A *Feladatkezelő* (Task Manager) az egyik alapvető eszköz a folyamatok kezelésére és a rendszer terheltségének megállapítására. A Ctrl+Alt+Del kombinációra megjelenő menüből hozható elő például.



15. ábra: A Feladatkezelő nézetei

A *Folyamatok* lapot a Windows 8-ban teljesen áttervezték[7] (15. ábra). Kevesebb információt jelenít meg, azonban azt sokkal könnyebben áttekinthető formában (pl. az erőforrás-használat mértékét hő térkép is jelzi). Így egy gyors képet kaphatunk arról, hogy mi terheli jelenleg a rendszerünket.

A korábbi részletes folyamatlistát a *Részletek* (Details) lapon találjuk. Érdemes az oszlopok nevén jobb gombbal kattintva előhozható *Oszlopok kiválasztása* menüben körülnézni, sokkal több mindent tud megjeleníteni, mint ami ki van választva alapértelmezésként (például PID, okozott laphibák száma, teljes elérési út, stb.). A folyamatok jobbgombos menüjében lehet a folyamatokat bezárni, vagy akár megváltoztatni a prioritásukat (a valós idejű prioritás használatával bánjunk óvatosan).

[7] Egy érdekes leírás az új Feladatkezelő tervezéséről: Building Windows 8 Blog, The Windows 8 Task Manager, URL: http://blogs.msdn.com/b/b8/archive/2011/10/13/the-windows-8-task-manager.aspx

A Feladatkezelőben meg lehet azt is nézni, hogy az egyes svchost.exe példányokban mi fut, erre való a *Szolgáltatások* lap, és ott az egyes szolgáltatásoknál a *Részletek megjelenítése* menüpont. Itt a csoport tulajdonságnál megnézhetjük azt, hogy milyen névvel hivatkozik a rendszer az adott svchost példányra.

A *Teljesítmény* fülön egy gyors áttekintést kaphatunk arról, hogy mennyire terhelt a rendszer. Windows 8 esetén az egyes erőforrásokhoz tartozó leghasznosabb információk kaptak itt helyett.

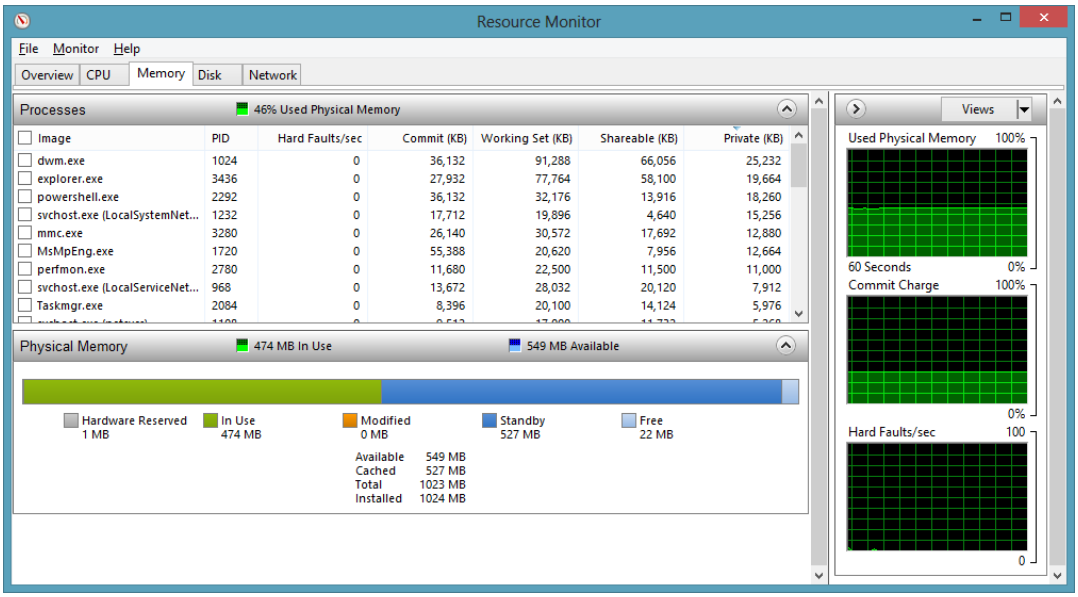
- *CPU*: magonkénti aktuális kihasználtság, összes folyamat és szál száma, CPU hardver adatai…
- *Memória*: aktív (In use) memória mértéke, aktuális és maximális virtuális memória (Commited)…
- *Lemez*: aktuális írási és olvasási sebesség, átlagos válaszidő (ezt is érdemes figyelni, és nem csak az átviteli sebességet)…
- *Hálózat*: küldési és fogadási sebesség, alapvető hálózati adatok…

A Feladatkezelő jó áttekintésre, azonban ha részletesebb képet akarunk kapni a rendszerről, akkor érdemes a következő fejezetben szereplő teljesítményszámlálókhoz fordulni.

3.5 Erőforrás-figyelő és Teljesítményfigyelő

A rendszerrel kapcsolatos részletes teljesítményjellemzőket az *Erőforrás-figyelő*ben (Resource Monitor) és a *Teljesítményfigyelő*ben (Performance Monitor) nézhetjük meg.

Az Erőforrás-figyelő a Feladatkezelőnél egy fokkal részletesebb (16. ábra). Tipikus felhasználása például amikor egy „lassú” rendszernél meg akarjuk nézni, hogy pontosan melyik erőforrástípusból van hiány vagy melyik fájlt írják vagy olvassák éppen.



16. ábra: Az Erőforrás-figyelő Memória lapja

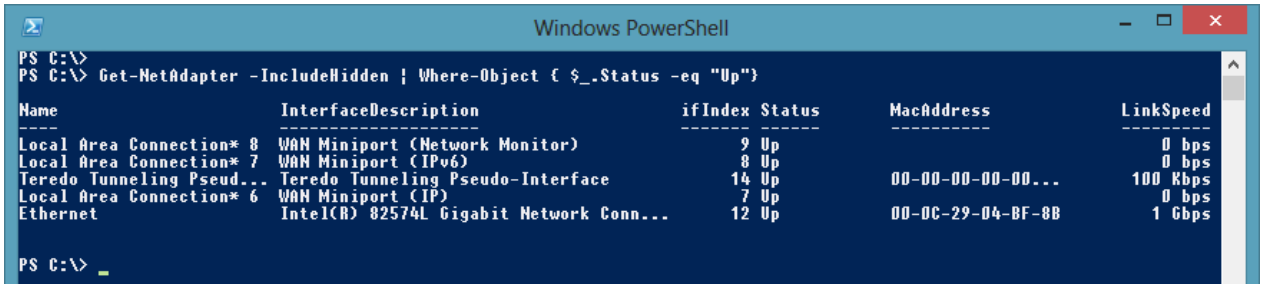
A Teljesítményfigyelőben úgynevezett *teljesítményszámlálók* (performance counter) értékeit jeleníthetjük meg. Az operációs rendszer komponenseinek nagy része biztosít ilyen számlálókat (pl. folyamathoz tartozó virtuális memória aktuális mérete, fizikai lemez írási sebessége, stb.), de egyéb programok is beregisztrálhatnak számlálókat. A teljesítményszámláló hozzáadása résznél érdemes bekapcsolni a *Leírás megjelenítése* opciót. A számlálók hozzáadása után a grafikon tulajdonságainál az *Adat* fül *Méret* beállításánál lehet az adott számlálóhoz tartozó skálázást megadni, ez akkor hasznos, ha nagyon eltérő nagyságrendű számlálókat jelenítünk meg egyszerre.

Az *Adatgyűjtő-csoportosítókban* (Data Collector Sets) a fenti számlálókból állíthatók össze csoportok, amiket utána folyamatosan figyel a rendszer. A *System Diagnostics* csomag például összegyűjti elindításkor a rendszer legfontosabb adatait (milyen hardver van a gépben, mik az automatikusan elinduló programok, stb.), majd figyeli a legfontosabb rendszer teljesítményszámlálókat. Az így összegyűjtött adatokat később a jelentések résznél nézhetjük meg.

A *Megbízhatóságfigyelő*ben (Reliability Monitor) az elmúlt időben bekövetkezett rendszer- és alkalmazáshibákat nézhetjük meg. Az egyes időpontok esetén kiírja az ahhoz kapcsolódó fontosabb változásokat is (frissítés, eszközmeghajtó vagy szoftver telepítése), így hirtelen megjelenő gyakori hibák esetén lehet következtetni, hogy milyen komponens okozhatja a hibát.

3.6 PowerShell

A PowerShell a Windows új parancssori felülete (17. ábra). Előnye a korábbi parancssorhoz (command prompt) képest, hogy ez egy objektum-orientált, a .NET keretrendszerre épülő, flexibilis, kényelmesen használható szkriptelési nyelv és környezet. Windows 8 óta az operációs rendszer legtöbb komponensének van már natív PowerShell felülete is, pl. a hálózati beállításokat is lehet így kezelni a korábbi parancssori programok helyett (ipconfig, netsh stb.). Egyelőre még a régi parancssori programok is működnek, azonban későbbi Windows verziókból ezek kikerülhetnek.



17. ábra: PowerShell utasítás a hálózati adapterek lekérdezésére és szűrésére

4 Biztonság

Egy operációs rendszer biztonsági rendszere elég összetett, itt most csak pár lényeges részre térünk ki.

4.1 A biztonsági rendszer elemei

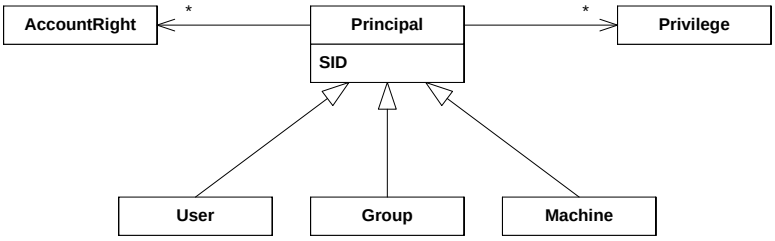
Alapvetően a következő biztonsági feladatokat lehet jól elkülöníteni:

- *Hitelesítés (authentication)*: azonosítani kell, hogy kitől származik a kérés. Például a rendszerbe való bejelentkezéskor felhasználónévvel és jelszóval azonosítjuk magunkat a rendszer felé.
- *Engedélyezés (authorization)*: jogosultság kiosztás, ellenőrizni kell, hogy a kérőnek van-e joga végrehajtani az adott kérést.
- *Auditálás*: a kérések biztonsági naplóba írása.

Ezeket a feladatokat a Windowsban a következő komponensek látják el:

- A felhasználók hitelesítését az LSASS végzi. Képes a helyi felhasználói adatbázis alapján ellenőrizni a bejelentkezést, tartományi (domain) környezetben pedig kapcsolatba lép a központi címtárat tároló szerverrel, és vele végezteti el az ellenőrzést.
- A biztonsági beállításokat a helyi biztonsági házirend tartalmazza, ez is a rendszerleíró adatbázisban tárolódik, a `HKLM\SECURITY\Policy` kulcs alatt.
- A biztonsági rendszer alapja az Executive réteg *Security Reference Monitor* komponense, ez ellenőrzi a futás során, hogy jogosultak-e a védett objektumhoz való hozzáférések. Minden leíró megnyitásakor ellenőrizni kell a hozzáférést.
- Az audit események az Eseménynapló beépített Biztonság naplójába kerülnek bele.

A biztonsággal kapcsolatos legfontosabb alapfogalmakat foglalja össze a következő ábra (18. ábra).



18. ábra: Biztonsági entitások (részlet)

- *Principal*: A biztonsági rendszer által kezelt entitások összefoglaló neve. Csoportokat az egyszerűbb, átláthatóbb adminisztrálás érdekében érdemes létrehozni; a biztonsági beállításokat mindig csoportokra adjuk meg, és felhasználókat pedig csak eltávolítjuk vagy hozzáadjuk a megfelelő csoporthoz. A biztonsági rendszerben a számítógépet az NT AUTHORITY\System felhasználó reprezentálja (a Szolgáltatások résznél Local System néven hivatkoztunk rá).
- *Biztonsági azonosító* (SID, Security Identifier): A biztonsági rendszer nem a felhasználók nevét, hanem egy belső azonosítót, a SID-et használja. Egy SID hasonló alakban jeleníthető meg:

  S-1-5-21-13124455-12541255-61235125-500

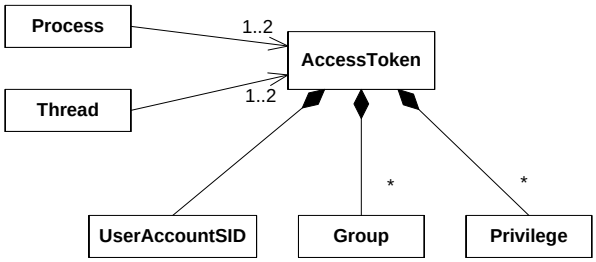
  A számítógép telepítéskor kap egy SID-et, a helyi felhasználók SID-je ebből generálódik, a végére a rendszer egy relatív azonosítót (RID, relative identifier) illeszt. A fenti SID a beépített rendszergazda felhasználó SID-je, az ő RID-je mindig 500. A sima felhasználók RID-jét 1000-től kezdi osztani a Windows. A beépített csoportoknak fix SID-je van, például a Mindenki (Everyone) csoporté S-1-1-0.
- *Jogosultság* (privilege): Az operációs rendszerrel kapcsolatos általános jogok, pl. számítógép leállítása, eszközmeghajtó betöltése vagy folyamat prioritásának megváltoztatása. A jogosultságok belső neve hasonló formájú: SeShutdownPrivilege, SeLoadDriverPrivilege.
- *Fiók jog* (account right): Meghatározzák, hogy egy felhasználó milyen módon léphet be. A belépés formái lehetnek interaktív, hálózaton keresztüli, belépés szolgáltatásként stb. Egy adott formát meg is lehet tiltani, nem csak megengedni.

Egy felhasználóhoz rendelt csoportokat és jogosultságokat például a *whoami* programmal nézhetjük meg (19. ábra).

```
PS C:\> whoami /all

USER INFORMATION
----------------

User Name      SID
============== ============================================
win14-win8\meres S-1-5-21-3647710530-3675882973-2840394154-1001


GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators                 Alias            S-1-5-32-544 Mandatory group, Enabled by default, Enabled group, Group owner
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE               Well-known group S-1-5-4      Mandatory group, Enabled by default, Enabled group
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default, Enabled group
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\NTLM Authentication       Well-known group S-1-5-64-10  Mandatory group, Enabled by default, Enabled group
Mandatory Label\High Mandatory Level   Label            S-1-16-12288 Mandatory group, Enabled by default, Enabled group


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                                        State
============================= ================================================== ========
SeIncreaseQuotaPrivilege      Folyamat memóriakvótájának módosítása              Disabled
SeSecurityPrivilege           A naplózás és a biztonsági napló kezelése          Disabled
SeTakeOwnershipPrivilege      Fájlok és más objektumok saját tulajdonba vétele   Disabled
SeLoadDriverPrivilege         Eszközillesztők betöltése és eltávolítása a memóriából Disabled
SeSystemProfilePrivilege      Rendszerteljesítmény kiértékelése                  Disabled
```

19. ábra: Felhasználó csoportjai és jogai

Az operációs rendszer minden folyamathoz és szálhoz hozzárendel egy *hozzáférési tokent* (access token), ami összefoglalja a legfontosabb biztonsági információkat. Az alábbi ábrán a tokennek csak a legfontosabb részei láthatóak, a hozzáférési tokennek rengeteg egyéb mezője is van (pl. elsődleges csoport, token forrása, stb.).



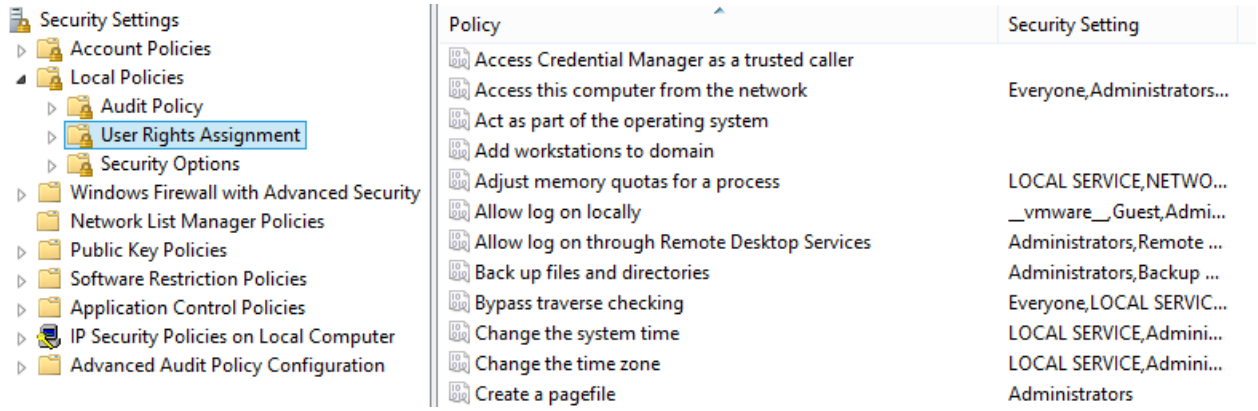
20. ábra: Hozzáférési token (részlet)

A felhasználó bejelentkezésekor a rendszer létrehoz egy tokent, és azt hozzárendeli a felhasználó folyamataihoz. A folyamatok ideiglenesen kaphatnak egy másik tokent, a *megszemélyesítési* (impersonation) tokent, ilyenkor más felhasználó nevében férnek hozzá a rendszerhez. Tipikus példája ennek, amikor egy fájlszerver folyamat ideiglenesen átvált a kérést küldő felhasználóra, és annak a nevében próbálja a kérést végrehajtani, így ellenőrizve azt, hogy a felhasználónak tényleg van joga hozzáférni az adott fájlhoz. A *runas* parancs segítségével lehetőség van arra is, hogy más felhasználó nevében indítsunk el egy programot, ilyenkor az egy teljesen új tokent kap.

A token tartalmazza még a felhasználó összes csoportját, és az összes jogosultságot, amit közvetlenül a felhasználóhoz vagy a csoportjaihoz rendeltünk.

A felhasználókat és csoportokat a *Sajátgépen jobb gomb / Kezelés / Helyi felhasználók és csoportok* résznél tudjuk adminisztrálni.

A biztonsági házirend kezelésére a *Felügyeleti eszközök / Helyi biztonsági házirend* modul szolgál.



21. ábra: Helyi biztonsági házirend

- *Jelszóházirend*: megadható, hogy milyen gyakran kell cserélni a jelszavakat, azok bonyolultak legyenek-e stb.
- *Fiókzárolási házirend*: mennyi sikertelen kísérlet után legyen letiltva az adott felhasználó.
- *Naplórend*: milyen eseményeket naplózzon a rendszer a biztonsági naplóba, pl. felhasználók kezelése vagy jogosultság használata.
- *Felhasználói jogok kiosztása*: jogosultságok és fiók jogok beállítása.
- *Biztonsági beállítások*: egyéb, általános biztonsági opciók, pl. digitálisan alá kell-e írni a hálózati kommunikációt vagy le van-e tiltva a vendég fiók.

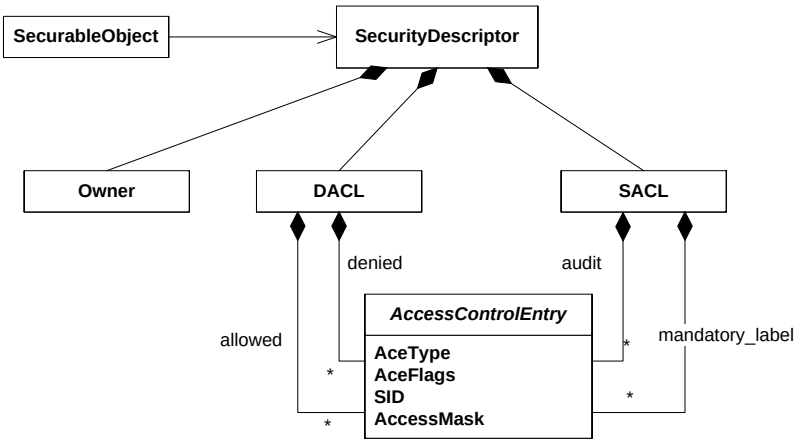
4.2 Hozzáférés-szabályozás

A Windowsban minden objektumhoz lehet hozzáférési védelmet definiálni, legyen az folyamat, fájl, hardver eszköz, megosztott memória vagy egy kulcs a rendszerleíró adatbázisban.

A hozzáférés-szabályozás alapvetően kétféle módszerrel valósítható meg a Windowsban:

- *Integritási szintek*: Vistában bevezetett funkció a *kötelező integritás ellenőrzés* (mandatory integrity control). Minden felhasználó és objektum kap egy integritási szintet (alacsony, közepes, magas vagy rendszer). Az objektumra pedig beállítható, hogy alacsonyabb integritási szintű felhasználók által indított folyamatok ne olvashassák, írhassák vagy hajthassák végre azt.
- *Hozzáférési listák*: A Windows objektumokhoz rendelhető *belátás szerinti hozzáférés-szabályozást* (DAC, discretionary access control) alkalmaz, azaz az objektumokhoz meg van adva, hogy kik férhetnek hozzá, azonban ezt egy megfelelő joggal rendelkező egyén később módosíthatja.

Egy védett objektumhoz való hozzáférési védelmet a biztonsági leíró foglalja össze (22. ábra).



22. ábra: Biztonsági leíró (részlet)

Minden objektumhoz tartozik egy biztonsági leíró, ennek a legfontosabb elemei:

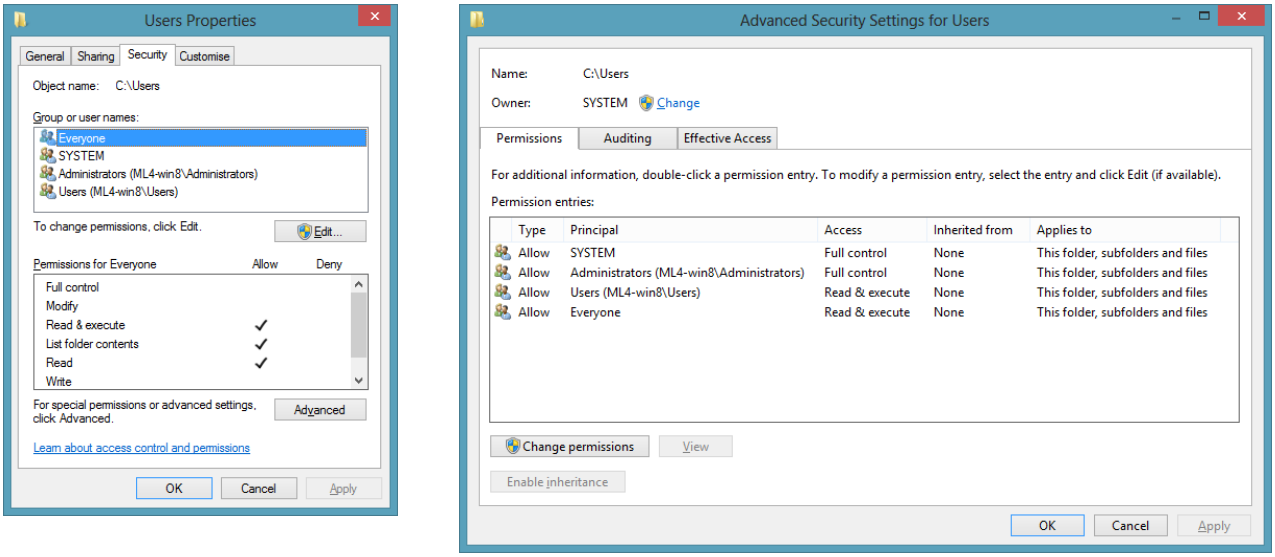
- *Tulajdonos*: Minden objektumnak van egy tulajdonosa, aki hozzáférhet az objektumhoz akkor is, ha a beállított hozzáférési engedélyek ezt nem engednék meg.
- *DACL* (discretionary access control list): A DACL egy lista, ami tartalmazza, hogy ki és milyen módon férhet hozzá az objektumhoz. Kétféle lista elem létezik, a megengedő és a tiltó, a tiltónak mindig magasabb prioritása van. A hozzáférési lista egy elemében (access control entry, ACE) eltárolják az elem típusát, egy flaget, ami az öröklődést befolyásolja, a SID-et, akire vonatkozik az ACE és egy hozzáférési maszkot. DACL esetén ez a maszk általános esetben az írás, olvasás, végrehajtás, tulajdonos írása, DAC írása stb. műveletek kombinációjából áll (ez a lista bővülhet, attól függően, hogy éppen milyen típusú objektumról van szó).
- *SACL* (system access control list): a hozzáférés naplózását és integritás ellenőrzést befolyásolja. Naplózó ACE esetén megadhatjuk, hogy sikeres vagy sikertelen hozzáférést akarunk naplózni. Integritást ellenőrző ACE esetén a SID három speciális SID lehet, amik az alacsony, közepes és magas szintet jelölik, a maszk pedig NO_WRITE_UP, NO_READ_UP és NO_EXECUTE_UP lehet.

Ezeket felhasználva a hozzáférés ellenőrzése a következő lépésekből áll:

1. A kérő megadja a hozzá tartozó hozzáférési tokent és, hogy milyen műveletet szeretne végrehajtani.
2. A rendszer ellenőrzi, hogy a kérő integritási szintje elég magas-e az adott művelet végrehajtásához, ha nem, akkor megtagadja a hozzáférést.
3. A rendszer ellenőrzi a tiltó hozzáféréseket. Ha meg van tiltva a felhasználónak vagy bármelyik olyan csoportnak, aminek tagja az adott művelet, akkor elutasítja a kérést.
4. A rendszer megvizsgálja a megengedő hozzáféréseket, és ha szerepel benne a felhasználó vagy valamelyik csoport, aminek tagja, akkor megengedi a hozzáférést, egyéb esetben elutasítja.
5. Ha az adott felhasználóhoz vagy egy csoporthoz, aminek tagja, volt naplózó SACL elem beállítva, akkor a kérés eredményét a rendszer naplózza.

Az egyszerűbb adminisztráció érdekében a hozzáférési listák *öröklődhetnek* (inheritance). Egy ACE-ra beállítható, hogy az adott objektum gyerekeire (pl. rendszerleíró adatbázis kulcsok esetén a benne tárolt értékek, NTFS mappák esetén a mappa alkönyvtárai és fájljai) is legyen érvényes. A gyerekobjektumokon pedig megadható, hogy használják a szülőkön definiált örökölhető engedélyeket, vagy szakítsa meg a rendszer az öröklési láncot és használjon teljesen új engedélyeket.

Nézzük meg az NTFS hozzáférési listák példáján keresztül, hogy hogyan lehet beállítani a hozzáférést!



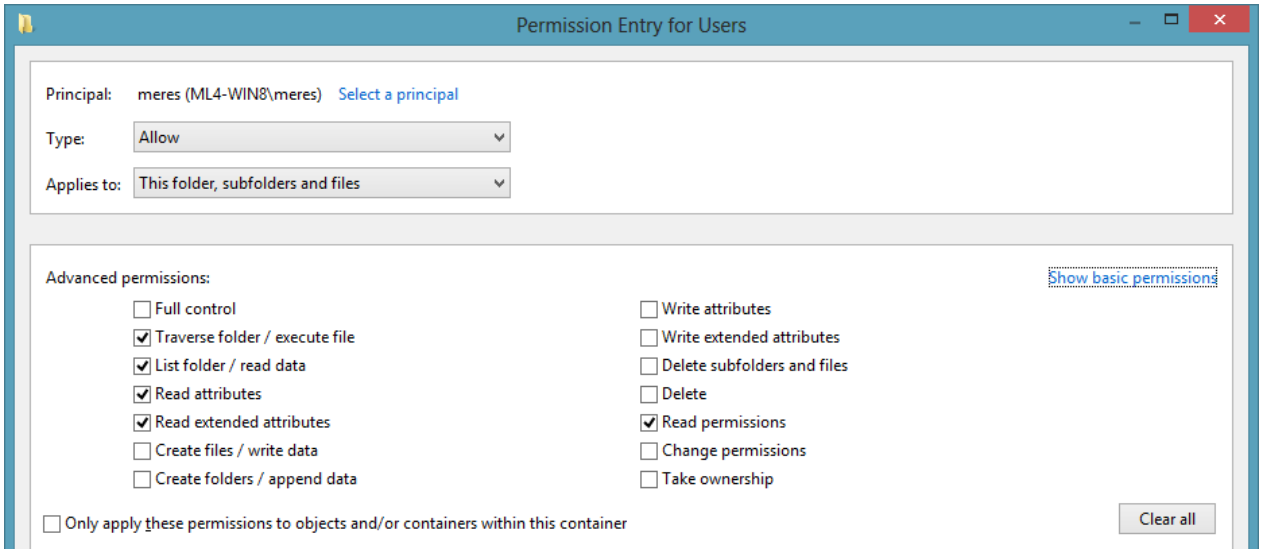
23. ábra: Mappa egyszerű és részletes biztonsági beállításai

A bal oldali ábrán egy mappa biztonsági beállításai láthatóak a mappa tulajdonságainál. Ez a nézet egy egyszerűsített kép, a felületen nem elemi hozzáférési műveleteket láthatóak, hanem azokból képzett gyakran használt csoportok. Például az *Olvasás* magában foglalja az adatok, attribútumok és engedélyek olvasása engedélyeket.

A *Speciális* (Advanced) gombra kattintva jön elő a jobb oldali ábra.

- *Engedélyek (Permissions)*: a fájlokhoz rendelt engedélyek részletes listája.
- *Naplózás (Auditing)*: naplózó ACE-ek hozzáadására itt van lehetőség.
- *Tulajdonos (Owner)*: megtekinthetjük az aktuális tulajdonost, és ha megvan a megfelelő jogosultságunk (SeTakeOwnershipPrivilege), akkor meg is változtathatjuk azt.
- *Hatályos engedélyek (Effective Access)*: leellenőrizhető, hogy egy felhasználónak vagy csoportnak mik lesznek az eredő jogai az összes ACE figyelembe vétele után.

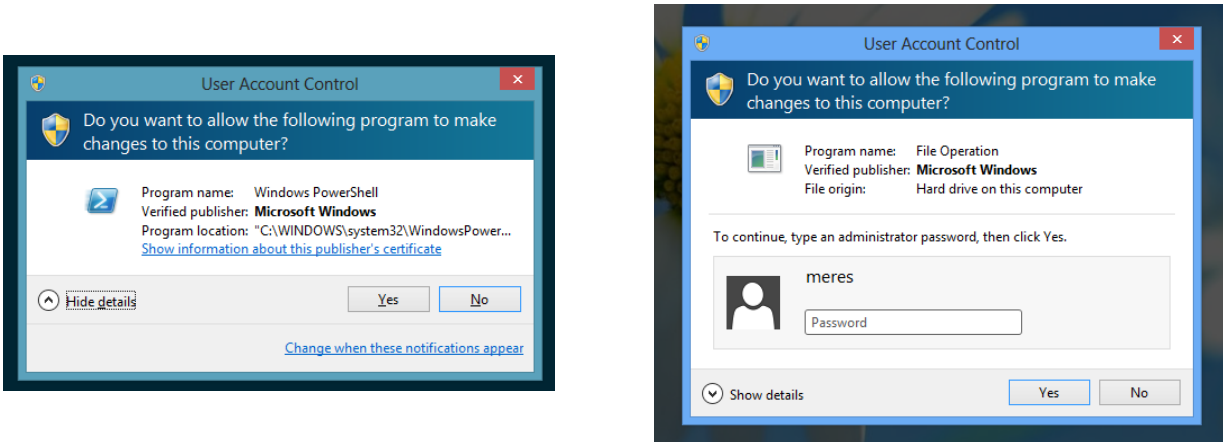
Az engedélyek szerkesztése résznél részletesen beállíthatóak, hogy milyen elemi engedélyek kerüljenek bele ebbe az ACE-ba, és, hogy az hogyan öröklődjön (24. ábra).



24. ábra: ACE hozzáadása

4.3 Felhasználói fiókok felügyelete (UAC)

A Vistában bevezetett egyik komoly biztonsági változás a felhasználói fiókok felügyelete (user account control, UAC).



25. ábra: UAC működés közben, rendszergazda és sima felhasználó alól

A módszer lényege, hogy a rendszergazdai jogokkal rendelkező felhasználó bejelentkezéskor kétféle hozzáférési tokent kap. Az egyik egy sima felhasználói token, amiben nincsenek benne a rendszergazdai jogosultságok, az általa indított programok általában ezt használják. Ha olyan műveletet akar végrehajtani, amihez magasabb jogosultság kell, akkor használja a rendszer a másik tokent, de ehhez a felületen először engedélyezni kell az adott műveletet. A módszer használható akkor is, ha egyáltalán nem rendszergazda felhasználóval vagyunk bejelentkezve, csak ilyenkor még jelszót is kér a rendszer.

Így végre tényleg nem kell rendszergazda felhasználót használnunk a napi tevékenységek során, ami igen komoly biztonsági kockázat.

5 További hasznos eszközök

A mérést VMware Player alatt futó virtuális gépen végezzük. A VMware Player letölthető a http://www.vmware.com/ oldalról.

A virtuális gép a következőket tartalmazza:

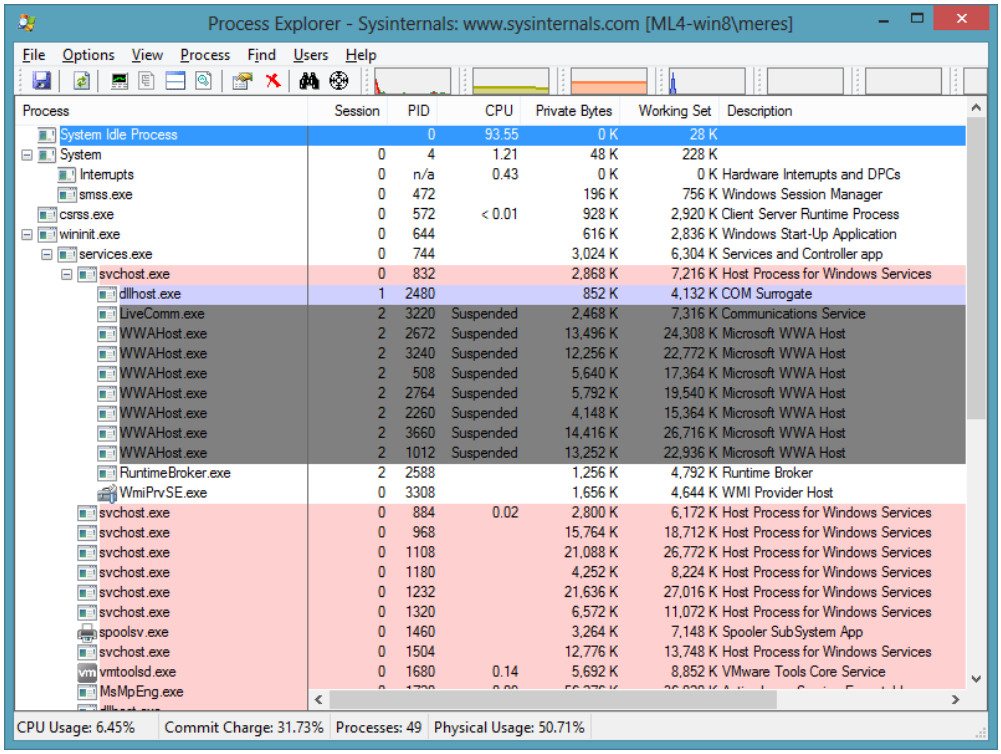
- Windows 8 Enterprise
- Sysinternals eszközkészlet: http://www.sysinternals.com
- Windows Debugging Tools: http://www.microsoft.com/whdc/devtools/debugging/default.mspx

A mérésre való felkészülés része a VMware Player és a Sysinternals Process Explorer kipróbálása, ezek ismerete nélkül a mérés általában nem teljesíthető a megadott idő alatt.

5.1 Sysinternals Process Explorer

A Process Explorer a Feladatkezelő jócskán felokosított változata.

A felső sorban a CPU, memória és I/O kihasználtság grafikonját látjuk. A középső nagy részen a folyamatok listája látható. Az alsó részen a kiválasztott folyamathoz tartozó leírókat vagy dll-eket tudjuk megnézni (a két mód között a Ctrl+H és Ctrl+D billentyűkombinációval válthatunk). Arra figyeljünk, hogy ha a folyamatok nagy részénél nincs kitöltve a leírás és a cég, akkor valószínűleg nem rendszergazdai jogokkal indítottuk el a programot.



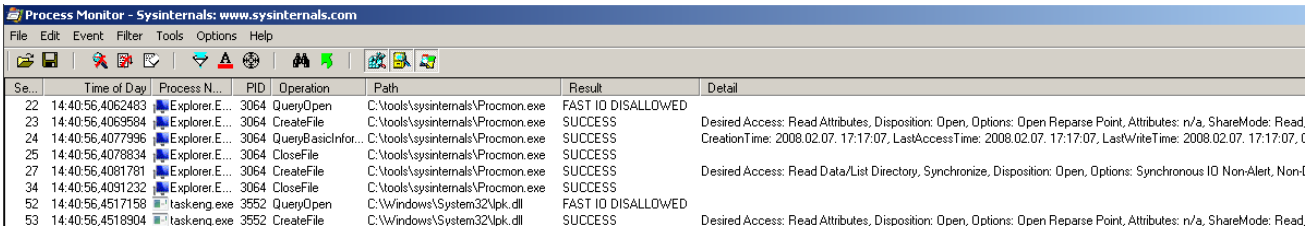
26. ábra: Sysinternals Process Explorer

A program rendkívül sok mindent tud, itt csak pár dolgot emelnénk ki:

- *System Information* (Ctrl+I): a Feladatkezelő teljesítmény lapjához hasonló nézet, csak több információval.
- *Find Window's Process*: egy célkereszttel rábökhetünk egy ablakra, és megmondja, hogy melyik folyamat tartozik hozzá.
- *View / Select columns*: az alapértelmezetten kívül rengeteg mindent meg tud még jeleníteni az egyes folyamatokról.
- *Process / Properties*: részletes információ a folyamatról. Például, hogy milyen szálak tartoznak hozzá, azok épp milyen függvényt hajtanak végre. Milyen TCP/IP kapcsolatokat tart nyitva a folyamat, vagy akár, hogy milyen *sztringek* találhatóak a folyamathoz tartozó binárisban (ez sok mindent elárul az adott folyamat funkciójáról).
- *Find / Find Handle or DLL*: ki tart nyitva egy adott leírót.

5.2 Sysinterals Process Monitor

A Process Monitor segítségével a fájl és rendszerleíró adatbázis hozzáféréseket lehet megnézni valós időben. Ha nem futtatunk semmit a rendszerben, akkor is rengeteg háttérművelet zajlik, így érdemes mindig szűrni a rögzített műveleteket a *Filter* (Ctrl + L) segítségével. A leggyakoribb szűrési feltétel a folyamat neve, de rengeteg más opció beállítható.



27. ábra Process Monitor

A Process Monitort különösen rejtélyes hozzáférési hibák esetén jön jól. Ilyenkor érdemes a Highlight segítségével kiemelni az érdekes sorokat, pl. ahol a Result értéke ACCESS DENIED.

5.3 Debugging Tools for Windows

Általános célú parancssori és grafikus debuggereket tartalmaz. A súgójában nem csak az eszközök konkrét beállításairól olvashatunk, hanem tipikus hibakeresési technikákat is bemutatnak. Képes távoli és helyi kernel debuggolásra is (ezt a Sysinternals csomagban lévő liveKD.exe is tudja).

A használata előtt érdemes az úgynevezett *symbol path*-t beállítani. A debugoláshoz szükség van a megfelelő fájlok debug szimbólumait tartalmazó fájlra is (pdb fájl). Ezt letölthetjük egyben a Debugging Tools weboldaláról, de hogy biztosak legyünk benne, hogy mindig a megfelelő verziót használja, érdemesebb inkább beállítani, hogy futás közben a webről töltse le azt, amire éppen szüksége van. Ehhez például a következőt kell megadni:

```
SRV*c:\symbols*http://msdl.microsoft.com/download/symbols
```

Ilyenkor a c:\symbols könyvtárba menti el a fenti URL-ről letöltött állományokat.

Ha kernel hibakeresést akarunk használni, akkor ahhoz engedélyezni kell először a hibakeresés indítási opciót, például a következő paranccsal:

```
bcdedit –debug on
```

Ezek után már használhatjuk a helyi kernel hibakeresést (File / Kernel debug / Local), csak figyeljünk arra, hogy a WinDbg-ot rendszergazdaként indítsuk el.

6 Ellenőrző kérdések

1. Mik az alrendszerek és mi a szerepük a Windowsban?
2. Hogy néz ki a Windows felépítése (ábra)?
3. Mi a munkamenet (session)?
4. Mik a szerepük a szolgáltatásoknak (service) a Windowsban? Milyen programok kapcsolódnak a kezelésükhöz?
5. Hol tárolja a Windows a felhasználók beállításait?
6. Mi a rendszerleíró adatbázis (registry), hogyan épül fel?
7. Mit csinál az svchost.exe?
8. Milyen eszközökkel érdemes vizsgálni a számítógép teljesítményét Windows alatt?
9. Mit tartalmaz egy ACL (Access Control List)?
10. Mondjon példát teljesítményszámlálókra (legalább 3 darab)!
11. Mondjon példát MMC beépülő modulokra (legalább 3 darab)!
12. Mi az lsass.exe feladata?
13. Hogyan férhet hozzá a rendszergazda egy olyan könyvtárhoz, amire egy felhasználó beállította, hogy másnak ne legyen hozzáférése hozzá?
14. Milyen elemeket tartalmazhat a helyi biztonsági házirend (legalább 3)?

7 Függelék

7.1 Hasznos gyorsbillentyűk

Néhány hasznos gyorsbillentyű, amit érdemes megtanulni a hatékonyabb kezelés érdekében[8].

Windows 8 előtt is használható	
Win (Windows gomb)	Start menü / képernyő megjelenítése
Win + D	Asztal megjelenítése az aktuális ablakok minimalizálásával
Win + E	Windows Explorer megjelenítése
Win + R	Futtatás ablak megjelenítése
Win + balra nyíl	Aktív ablak dokkolása a képernyő bal felére
Win + jobbra nyíl	Aktív ablak dokkolása a képernyő jobb felére
Win + felfelé nyíl	Aktív ablak teljes képernyőssé nagyítása
Windows 8-ban megjelent	
Win + Q	Keresési ablak megnyitása (Search pane)
Win + W	Beállítások között lehet vele keresni (Settings pane)
Win + C	Jobb oldali gombok megjelenítése (Charm bar)

[8] Lista a továbbiakról: Technet, Common Management Tasks and Navigation in Windows Server 2012, URL: http://technet.microsoft.com/en-us/library/hh831491.aspx#BKMK_keys