

Windows Power-User Guide

From zero to hero

Windows as a desktop, with a touch of Linux

Lectures on Modern Scientific Programming
Wigner RCP
23-25 November 2017



- Windows telepítés (Wigner ISO)
- A parancssor meghódítása (PowerShell)
- Csomagkezelés (Chocolatey)
- Kettő az egyben (Linux alrendszer)
- Fejlesztői környezet kialakítása





Windows telepítés

Nyissunk ablakot a világra

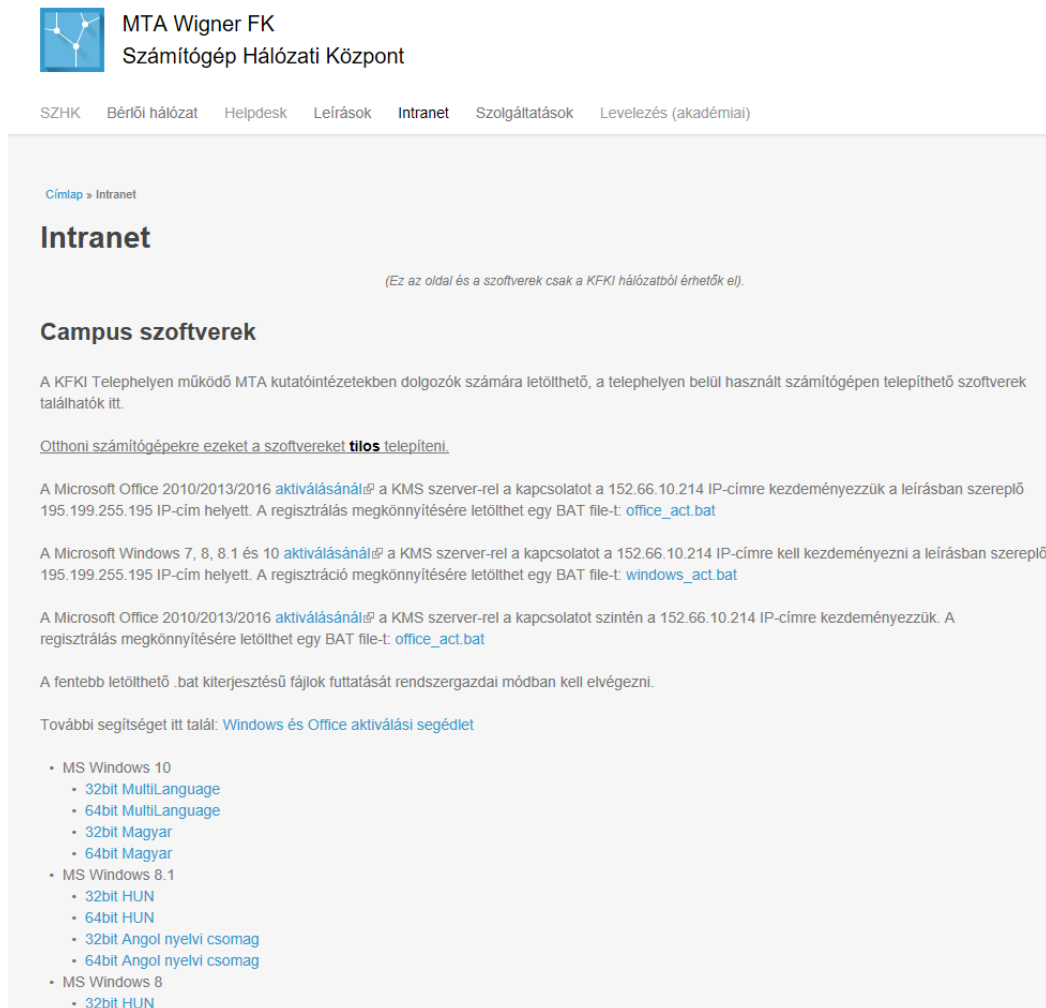
Mit telepítünk?

- Windows 10 Enterprise
- Minden akadémiai intézetben található gépre feltelepíthető
 - Jogi alapja a Microsoft és az MTA közti licenzs szerződés



Honnan?

- kfki.hu/intranet
- MS Windows 10
- 64bit Magyar
 - ~15 éve 64-bites minden gép
 - X86_64, X64, amd64
 - 32-bitet tessék elfelejteni
 - i386, X86 (!)



MTA Wigner FK
Számítógép Hálózati Központ

SZHK Bériői hálózat Helpdesk Leírások Intranet Szolgáltatások Levelezés (akadémiai)

Címlap » Intranet

Intranet

(Ez az oldal és a szoftverek csak a KFKI hálózathoz érhetőek el).

Campus szoftverek

A KFKI Telephelyen működő MTA kutatóintézetekben dolgozók számára letölthető, a telephelyen belül használt számítógépen telepíthető szoftverek találhatóak itt.

Otthoni számítógépekre ezeket a szoftvereket **tilos** telepíteni.

A Microsoft Office 2010/2013/2016 [aktiválásánál](#) a KMS szerver-rel a kapcsolatot a 152.66.10.214 IP-címre kezdeményezzük a leírásban szereplő 195.199.255.195 IP-cím helyett. A regisztrálás megkönnyítésére letölthet egy BAT file-t: [office_act.bat](#)

A Microsoft Windows 7, 8, 8.1 és 10 [aktiválásánál](#) a KMS szerver-rel a kapcsolatot a 152.66.10.214 IP-címre kell kezdeményezni a leírásban szereplő 195.199.255.195 IP-cím helyett. A regisztráció megkönnyítésére letölthet egy BAT file-t: [windows_act.bat](#)

A Microsoft Office 2010/2013/2016 [aktiválásánál](#) a KMS szerver-rel a kapcsolatot szintén a 152.66.10.214 IP-címre kezdeményezzük. A regisztrálás megkönnyítésére letölthet egy BAT file-t: [office_act.bat](#)

A fentebb letölthető .bat kiterjesztésű fájlok futtatását rendszergazdai módban kell elvégezni.

További segítséget itt talál: [Windows és Office aktiválási segédlet](#)

- MS Windows 10
 - [32bit MultiLanguage](#)
 - [64bit MultiLanguage](#)
 - [32bit Magyar](#)
 - [64bit Magyar](#)
- MS Windows 8.1
 - [32bit HUN](#)
 - [64bit HUN](#)
 - [32bit Angol nyelvi csomag](#)
 - [64bit Angol nyelvi csomag](#)
- MS Windows 8
 - [32bit HUN](#)



CB (Current Branch)

- Home és Pro rendszerekhez hasonlóan mindig a legújabb fő verzióra frissül
- Ideális lehet fejlesztői gépekhez, munkaadásokhoz, ahol nem riadnak vissza a GUI változásától

LTSB (Long-Term Servicing Branch)

- Csak a tartomány vezérlő veheti rá, hogy fő verziót váltson
- Ideális lehet munkaadásokhoz, ahol a produktivitásnál fontosabb a stabilitás



Mivel?

- wudt.codeplex.com
- A Codeplex volt a MS saját Githubja
- 2017 December 31-ével leáll
- Letölt, telepít, futtat

CodePlex Project Hosting for Open Source Software

Register | Sign In | Search all projects

Windows USB/DVD Download Tool

HOME | SOURCE CODE | DOWNLOADS | DOCUMENTATION | DISCUSSIONS | ISSUES | PEOPLE | LICENSE

Page Info | Change History (all pages) | Follow (373) | Subscribe

The Windows USB/DVD Download tool allows you to create a copy of your Windows 7/8 ISO file on a USB flash drive or a DVD. To create a bootable DVD or USB flash drive, download the ISO file and then run the Windows 7 USB/DVD Download tool. Once this is done, you can install Windows 7 or Windows 8 directly from the USB flash drive or DVD.

The ISO file contains all the Windows installation files combined into a single uncompressed file. When you download the ISO file, you need to copy it to some medium in order to install Windows. This tool allows you to create a copy of the ISO file to a USB flash drive or a DVD. To install Windows from your USB flash drive or DVD, all you need to do is insert the USB flash drive into your USB port or insert your DVD into your DVD drive and run Setup.exe from the root folder on the drive.

Note: You cannot install Windows from the ISO file until you copy it to a USB flash drive or DVD with the Windows 7 USB/DVD Download tool and install from there.

The copy of the ISO file that the Windows USB/DVD Download tool creates is bootable. Bootable media allows you to install Windows 7 without having to first run an existing operating system on your machine. If you change the boot order of drives in your computer's BIOS, you can run the Windows 7 installation directly from your USB flash drive or DVD when you turn on your computer. Please see the documentation for your computer for information on how to change the BIOS boot order of drives.

For Windows XP Users

The following applications must be installed prior to installing the tool:

- Microsoft .NET Framework v2 must be installed. It can be downloaded at <http://www.microsoft.com/downloads/details.aspx?FamilyID=0856EACB-4362-480D-8EDD-AA815C5E04F5&displaylang=en>.
- Microsoft Image Mastering API v2 must be installed. It can be downloaded at <http://www.microsoft.com/downloads/details.aspx?FamilyID=B5F726F1-4ACE-455D-BAD7-ABC4DD2F147B&displaylang=en>.

Last edited Nov 12, 2014 at 5:46 PM by zacharye, version 4

Search Wiki & Documentation

download

CURRENT 1.0
DATE Wed Dec 9, 2009 at 7:00 AM
STATUS Stable
DOWNLOADS [30 038 377](#)
RATING [★★★★★](#) 44 ratings [Review this release](#)

MOST HELPFUL REVIEWS

★★★★★ Installed Visual Studio 2010 Ultimate with USB, Worked 100 % fine. I would name this tool as Any ISO to USB or DVD Bootable. Though... [\(more\)](#)

★★★★★ great tool!

[View all reviews](#)

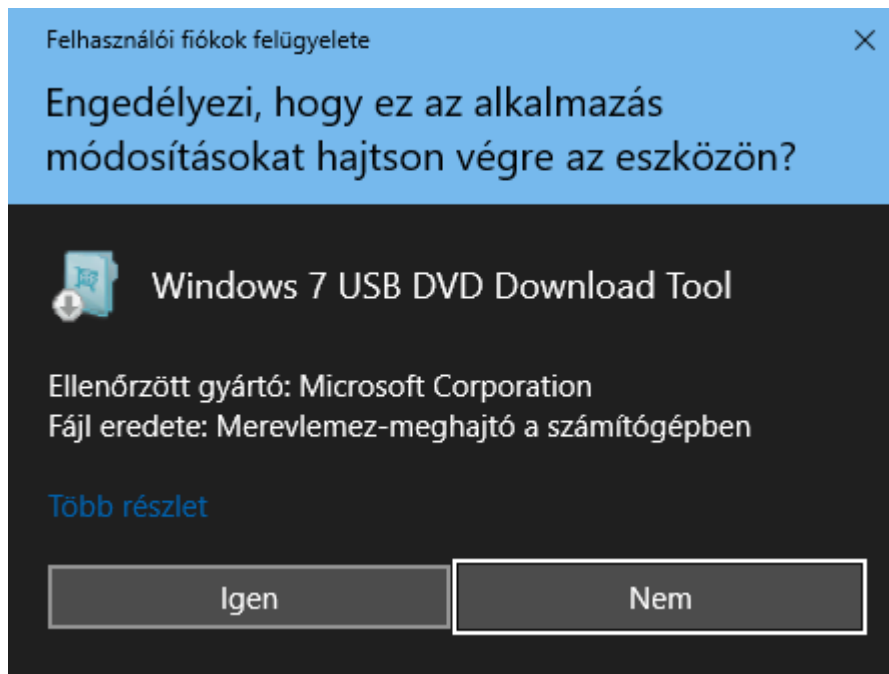
ACTIVITY

DOWNLOADS 138464
Days: 7 30 All [Details](#)

© 2006-2017 Microsoft | Get Help | Privacy Statement | Terms of Use | Code of Conduct | Advertise With Us | Version 6.23.2017.21053

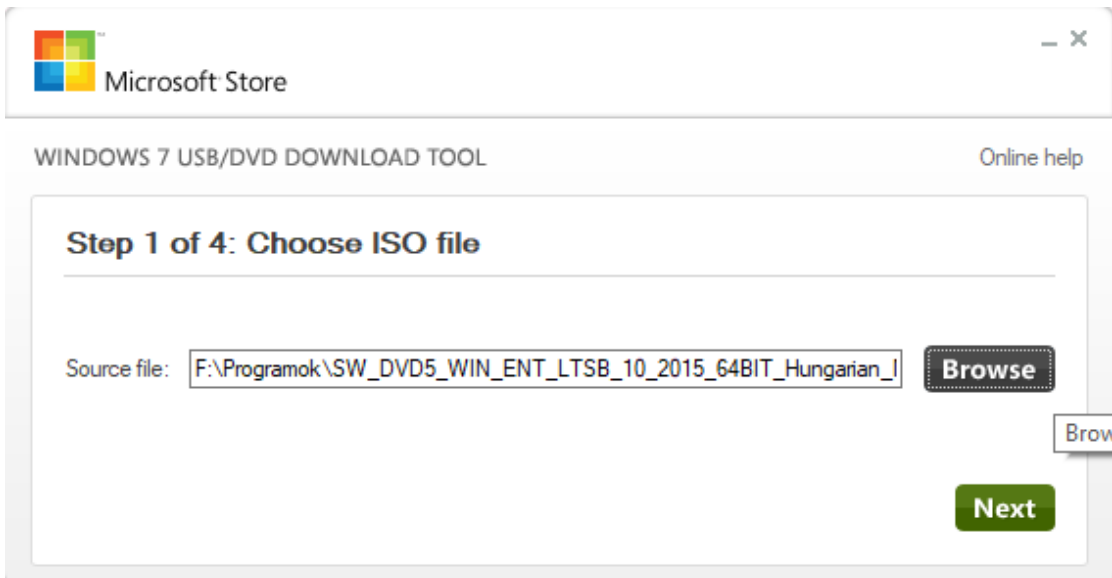


User Access Control



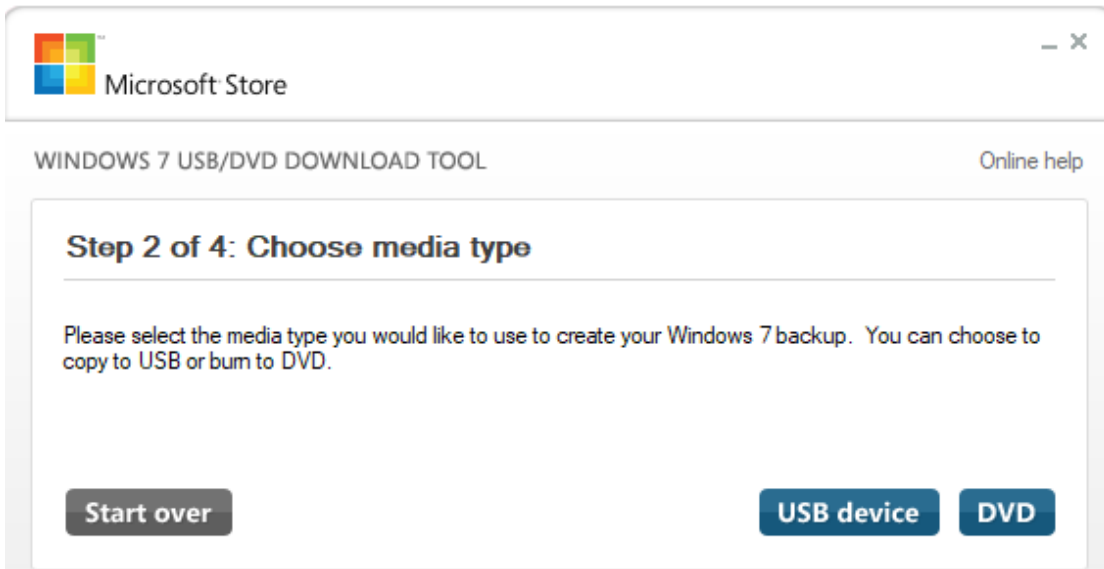
- Felhasználói fiók felügyelete
- Védett grafikus terminál mód
 - A feldobott ablak nem elérhető más programok számára
 - Kártékony programok nem hagyhatják jóvá, csak billentyűzet és egér visszajelzés érvényes
- Miért jött elő?
 - A programnak olyan jogosultságok kellene, amivel a rendszert is módosíthatja
 - Formáznia kell tudni meghajtókat (pendrive)

Telepítő pendrive készítése



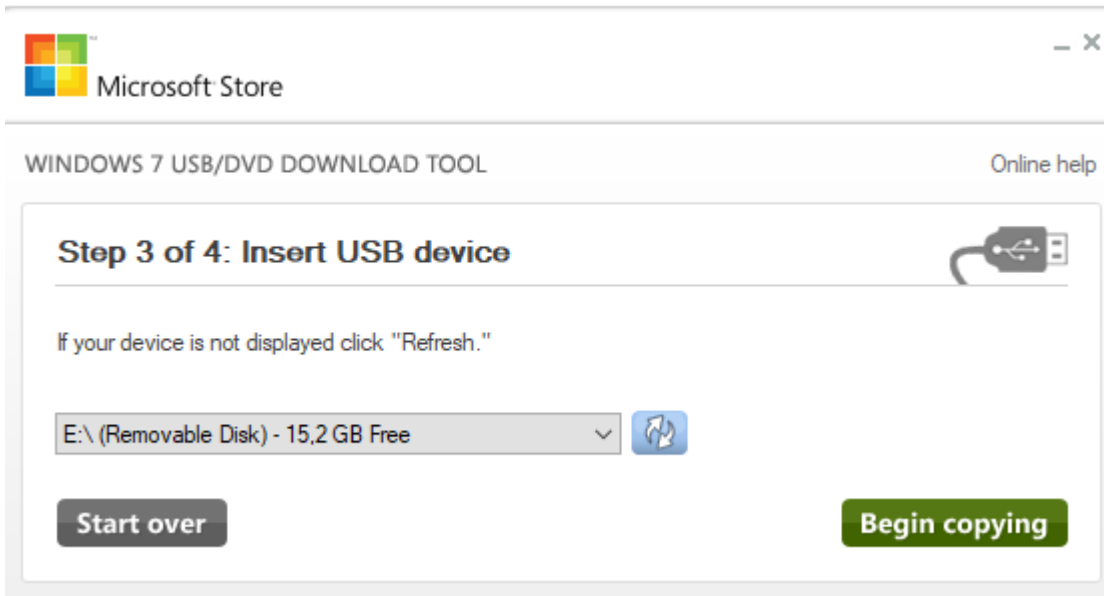
- Első lépésként adjuk meg a letöltött ISO helyét
 - Browse, klikk, klikk
 - Másol+Beilleszt valahonnan
- Next

Telepítő pendrive készítése



- „USB device” gomb, amennyiben telepítő pendrive-ra hajtunk
- Ellenkező esetben „DVD”
- Ha egyik sem, akkor mit keresünk itt??

Telepítő pendrive készítése



- Válasszuk ki a leőzerolni kívánt eszközt
- Szerencsénkre merevlemezeket nem listáz ki a program
 - DE azért legyünk óvatosak!
- Legalább 4GB-os pendrive kell
 - Nem minden 4GB elég nagy ☹
 - 8GB már biztos elég
- „Begin copying”, enjoy!

Telepítő pendrive: kész?

Elégedett felhasználó

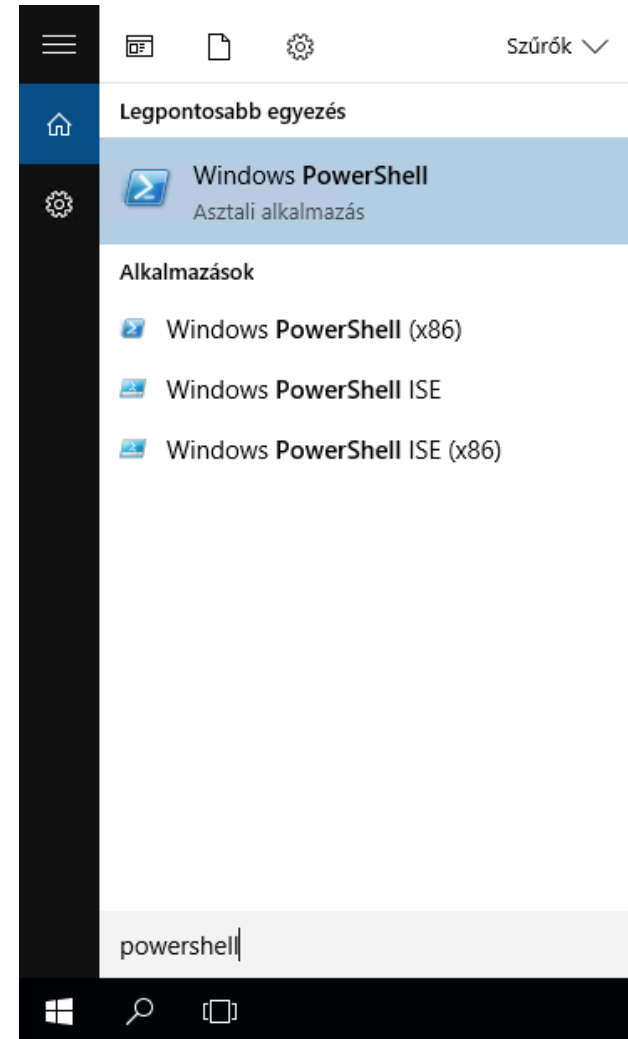


Elégedetlen konzol huszár



Telepítő parancssor módra

- Neten található leírások, többnyire a hagyományos parancssoron (cmd.exe) keresztül a beépített partícionáló (diskpart.exe) segítségével
- Mi azonban a hatalom kék kapuján belépve, PowerShellből fogunk telepítőt készíteni.
- A „Start” gomb után írjuk be: powershell
 - Jobb-klikk, rögzítés a tálcán



Telepítő parancssor módra

- A PowerShell részletesebben később kerül bemutatásra
- Ez csak egy rövid tessék-lássék bemutató lesz, milyen egyszerű megoldani egy ilyen feladatot
- Mivel mi is ugyanazt a partícionáló feladatot szeretnénk ellátni, így nekünk is rendszergazdai jogosultságok kellenek
- *nix környezetekkel ellentétben az emelt jogosultság nem állítható parancs szinten (sudo), hanem a shell létrehozásakor dől el.

```
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All rights reserved.  
  
Loading personal and system profiles took 834ms.  
PS C:\Users\Matty>
```



Telepítő parancssor módra

```
PS C:\WINDOWS\system32> Get-Disk
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
0	Samsung SS...	S1DHNSBF633364L	Healthy	Online	465.76 GB	GPT
2	ST1000LM 0...	9C000000000050506102	Healthy	Online	931.51 GB	GPT
1	SMI USB DISK	AA00000000000489	Healthy	Online	15.23 GB	MBR

- A Get-Disk paranccsal listázhatjuk ki a gépben található lemezeket
- Minket most a 16GB-os pendrive fog érdekelni
- Tegyük vágólapra a sorozatszámot későbbi használatra
 - Bal egér gombot lenyomva kijelöljük (vagy dupla-klikk a szóra)
 - Jobb gombot lenyomva a vágólapra tesszük (újboli jobb gombbal beillesztjük)



Telepítő parancssor módra

```
PS C:\WINDOWS\system32> Get-Disk
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
0	Samsung SS...	S1DHNSBF633364L	Healthy	Online	465.76 GB	GPT
2	ST1000LM 0...	9C000000000050506102	Healthy	Online	931.51 GB	GPT
1	SMI USB DISK	AA000000000000489	Healthy	Online	15.23 GB	MBR

- A Get-Disk paranccsal listázhatjuk ki a gépben található lemezeket
- Minket most a 16GB-os pendrive fog érdekelni
- Tegyük vágólapra a sorozatszámot későbbi használatra
 - Bal egér gombot lenyomva kijelöljük (vagy dupla-klikk a szóra)
 - Jobb gombot lenyomva a vágólapra tesszük (újboli jobb gombbal beillesztjük)



Telepítő parancssor módra

```
PS C:\WINDOWS\system32> Get-Disk -SerialNumber AA00000000000489 |  
>> Clear-Disk -RemoveData -Confirm:$false -PassThru |  
>> New-Partition -UseMaximumSize -IsActive:$true -AssignDriveLetter |  
>> Format-Volume -FileSystem NTFS -NewFileSystemLabel 'Windows 10 Installer'
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
E	Windows 10 Ins...	NTFS	Removable	OK	15.19 GB	15.23 GB

- Mit csinál a fentebbi egy soros parancs?
 - Lekéri az adott sorozatszámú lemezt
 - Ledózerolja kérdés nélkül
 - Létrehoz rajta egyetlen új, bootolható partíciót és mountolja
 - Formázza NTFS-re és elnevezi a kívánt névre



Telepítő parancssor módra

```
PS C:\WINDOWS\system32> Mount-DiskImage -ImagePath F:\win10.ISO -PassThru | Get-Volume
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
G	J_CES_X64...	Unknown	CD-ROM	OK	0 B	3.35 GB

- Mit csinál a fentebbi egy soros parancs?
 - Mountolja az ISO lemez képfájlt egy tetszőleges meghajtó betűre
 - Végül lekérdezi a mountolás eredményét
- Ez a parancs és az előző is visszaadott egy-egy betűjelet, ezeket használva:
 - Átmásoljuk a teljes DVD tartalmát a pendrive-ra (pár perc)
 - Rendszer betöltőt (bootloader) telepítünk a lemezre (Linuxon GRUB)

```
PS C:\WINDOWS\system32> Copy-Item -Recurse G:\* E:\
PS C:\WINDOWS\system32> Set-Location G:\boot\
PS G:\boot> .\bootsect.exe /nt60 E:
```



GUI vs. négy-soros szkript

- Átlag felhasználók számára ez persze túl sok voodoo
 - Nem kell mindenkinek konzol huszárnak lenni
- Power-userek számára viszont egy érdekes ujjgyakorlat

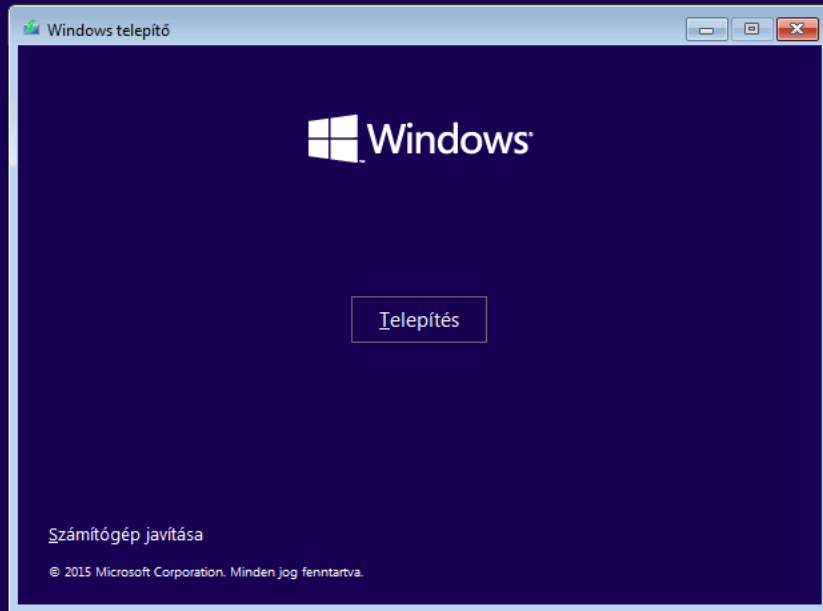


Windows telepítőn átszaladás



- Miután BIOS-ban beállítottuk, hogy USB-ről bootoljunk elsősorban, elindul a telepítő
 - Gombnyomásra indul csak
- Mindent hagyhatunk magyaron

Windows telepítőn átszaladás



• Vajon...



Windows telepítőn átszaladás

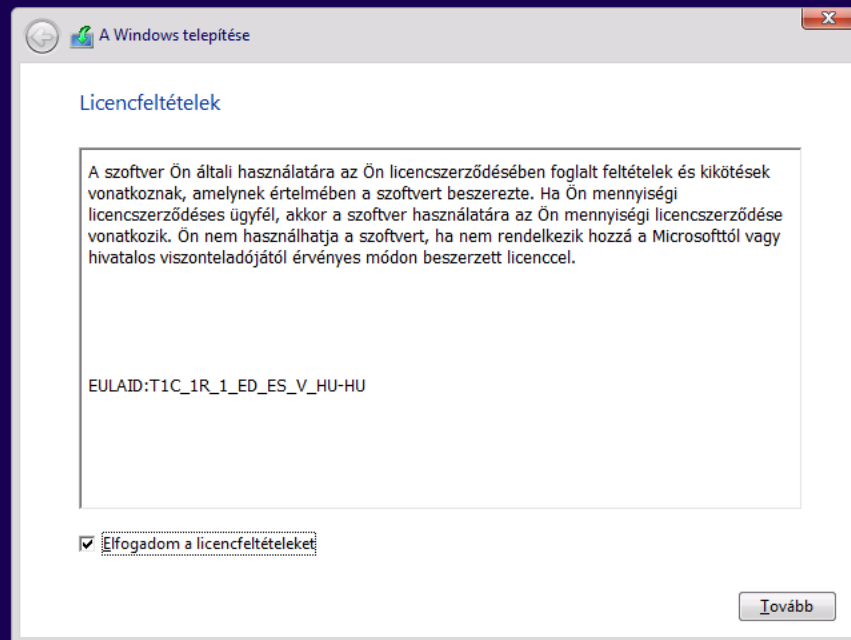


- Fehér ember várni homokóra



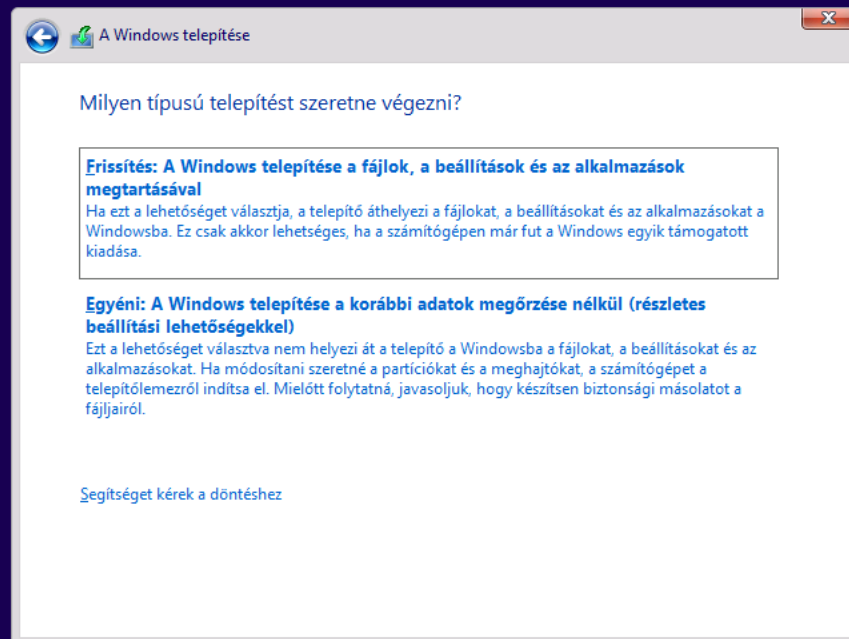
A telepítő elindul

Windows telepítőn átszaladás



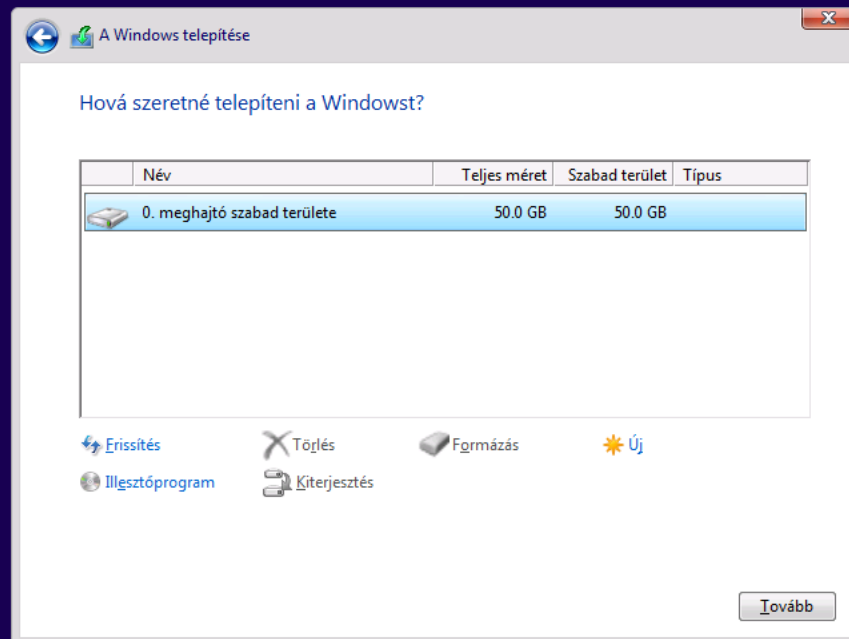
- Önre az a szerződés vonatkozik, amit itt nem áll módunkban megjeleníteni

Windows telepítőn átszaladás



- Egyéni
- Meglévő rendszer frissítése sosem végződik jól
 - Fő verziók közti váltás sosem röccenés mentes

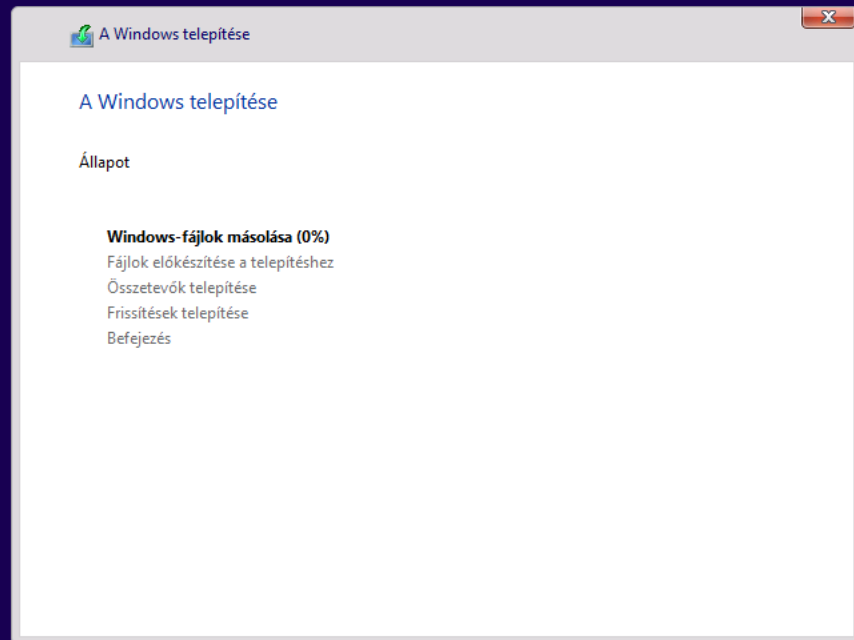
Windows telepítőn átszaladás



- Minden partícióra
 - Klikk
 - Töröl
- Utolsó partícionálatlan területet kiválaszt
 - Tovább



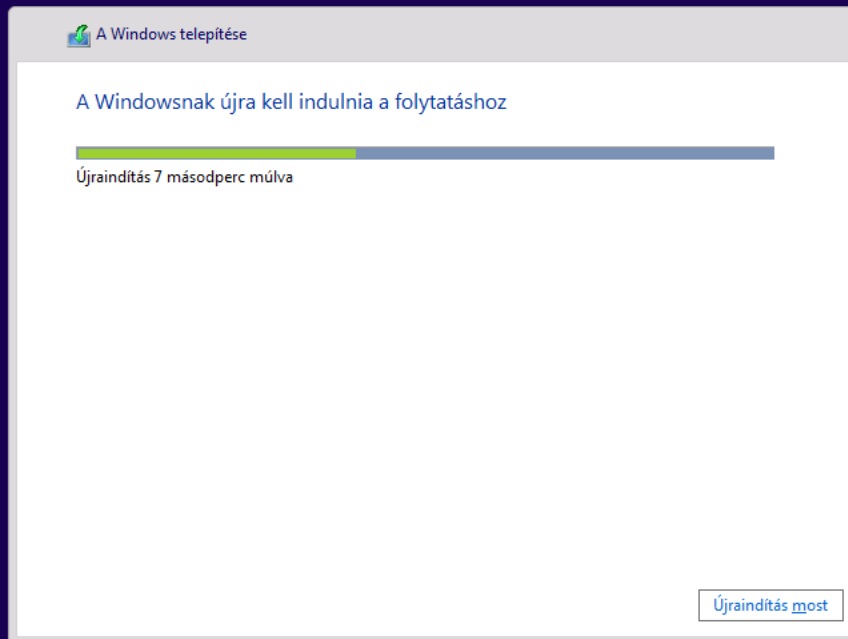
Windows telepítőn átszaladás



- Fehér ember várni homokóra



Windows telepítőn átszaladás



- Újraindítás
 - Mivel a telepítő csak gombnyomásra indul, ezért az se baj, ha benne marad a pendrive és arról bootolna
 - Tovább megy HDD bootra
 - Ha úgy volt beállítva

Windows telepítőn átszaladás

- Általános beállítások, a sokat vitatott adatgyűjtési lehetőségekkel
 - Türelmetlenek számára
 - „Gyorsbeállítások”
- Óvatos/paranoid felhasználóknak
- „Beállítások testreszabása”

Azonnal nekikezdhetsz

Ezeket a beállításokat bármikor módosíthatja. Ha a **Gyorsbeállítások** lehetőséget választja:

A beszédfelismerés, a gépelés és a szabadkézi műveletek személyre szabása a névjegyek, naptárak és más kapcsolódó bemeneti adatok elküldésével a Microsoftnak. A Microsoft ezeket az adatokat felhasználhatja a javaslattevő és felismerési platformjának fejlesztéséhez.

A Windows és az alkalmazások elkérhetik a helyadatait, beleértve a korábbi tartózkodási helyeket, illetve felhasználhatják a hirdetési azonosítóját a személyre szabott élmény biztosítása érdekében. A Microsoft és megbízható partnerei megkapnak bizonyos helyadatokat a helymeghatározási szolgáltatások fejlesztése érdekében.

Védelem a kártékony webes tartalom ellen, valamint a weblapok előzetes betöltése a Windows böngészőben a gyorsabb böngészés, a kényelmesebb olvasás és a jobb általános böngészési élmény érdekében. A Microsoftnak megkapja a böngészési előzményeket.

Automatikus kapcsolódás a javasolt nyitott hozzáférési pontokhoz és megosztott hálózatokhoz. Nem minden hálózat biztonságos.

További információ

Beállítások testreszabása

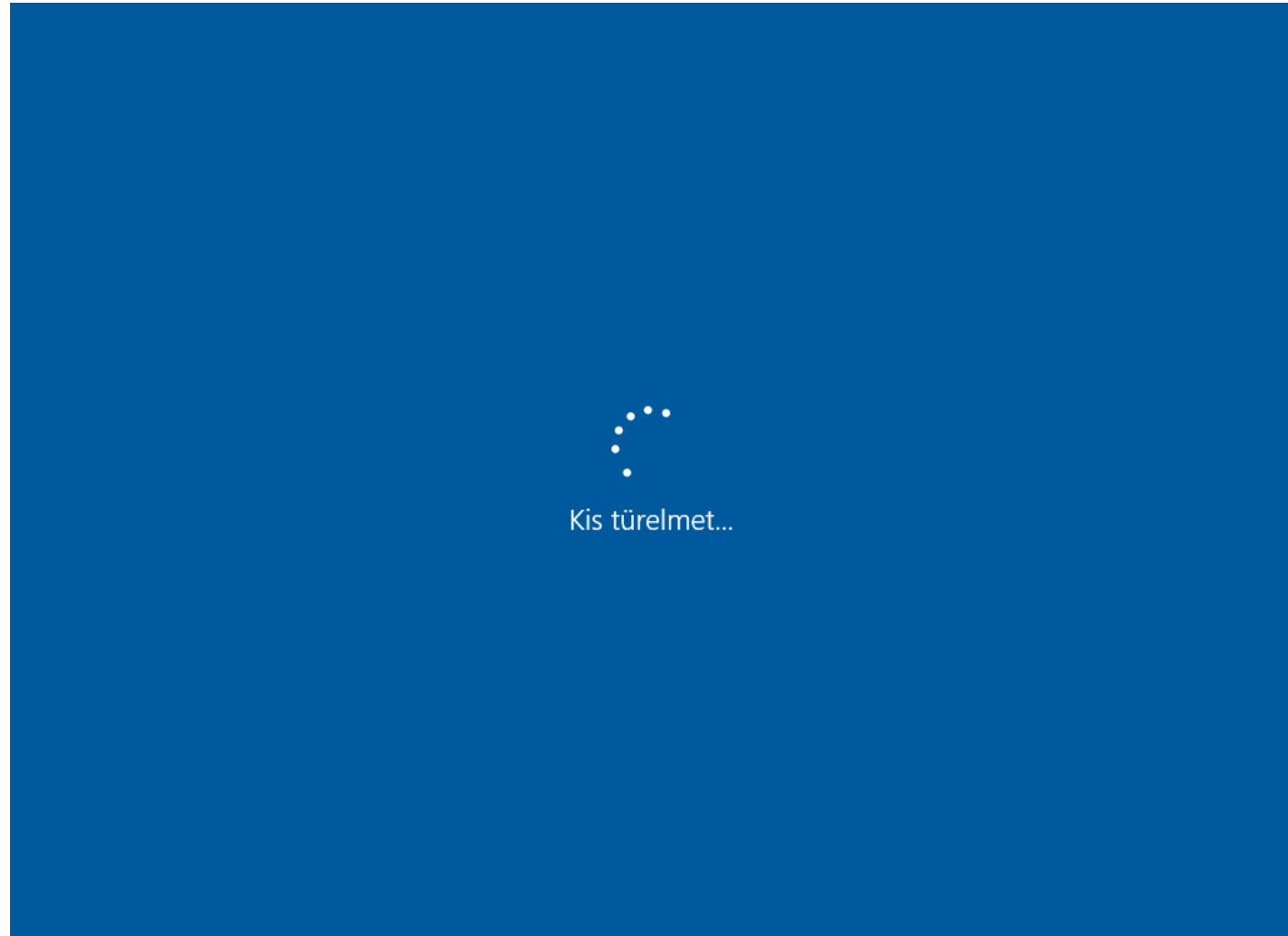


Gyorsbeállítások használata



Windows telepítőn átszaladás

- Fehér ember várni homokóra



Windows telepítőn átszaladás

- Ha van internet a telepítés során, akkor pár frissítést már itt leszed

Fontos frissítések letöltése

Ne kapcsolja ki a gépet



Windows telepítőn átszaladás

- Kicsi magyarázat
 - Kivonat a „Windows guruhoz” szánt kérdésekből



GYIK (Windows 10 Enterprise): Wigner ISO

- Mi ez a kérdés egyáltalán?
 - Mivel ez egy enterprise telepítő, így feltételezi, hogy a felhasználó tagja egy felhasználói adatbázisnak, és a gépe nem az éterben létezik, hanem egy vállalat/intézet részét képezi
 - A kérdés arra irányul, milyen felhasználói adatbázisból vegye a rendszer a valid felhasználókat? Az Azure felhő szolgáltatóból, vagy egy helyi adatbázisból?
- Mi a piztráng az az Active Directory?
 - Az Active Directory a Microsoft implementációja a Kerberos hitelesítési protokollnak, pár kiterjesztéssel
 - A Wigner kiszolgálóinak egy tekintélyes része is Kerberos hitelesítést használ
 - Többek között a GPU-Labor valamennyi gépe a központi Wigner Kerberos kiszolgálót használja hitelesítéshez



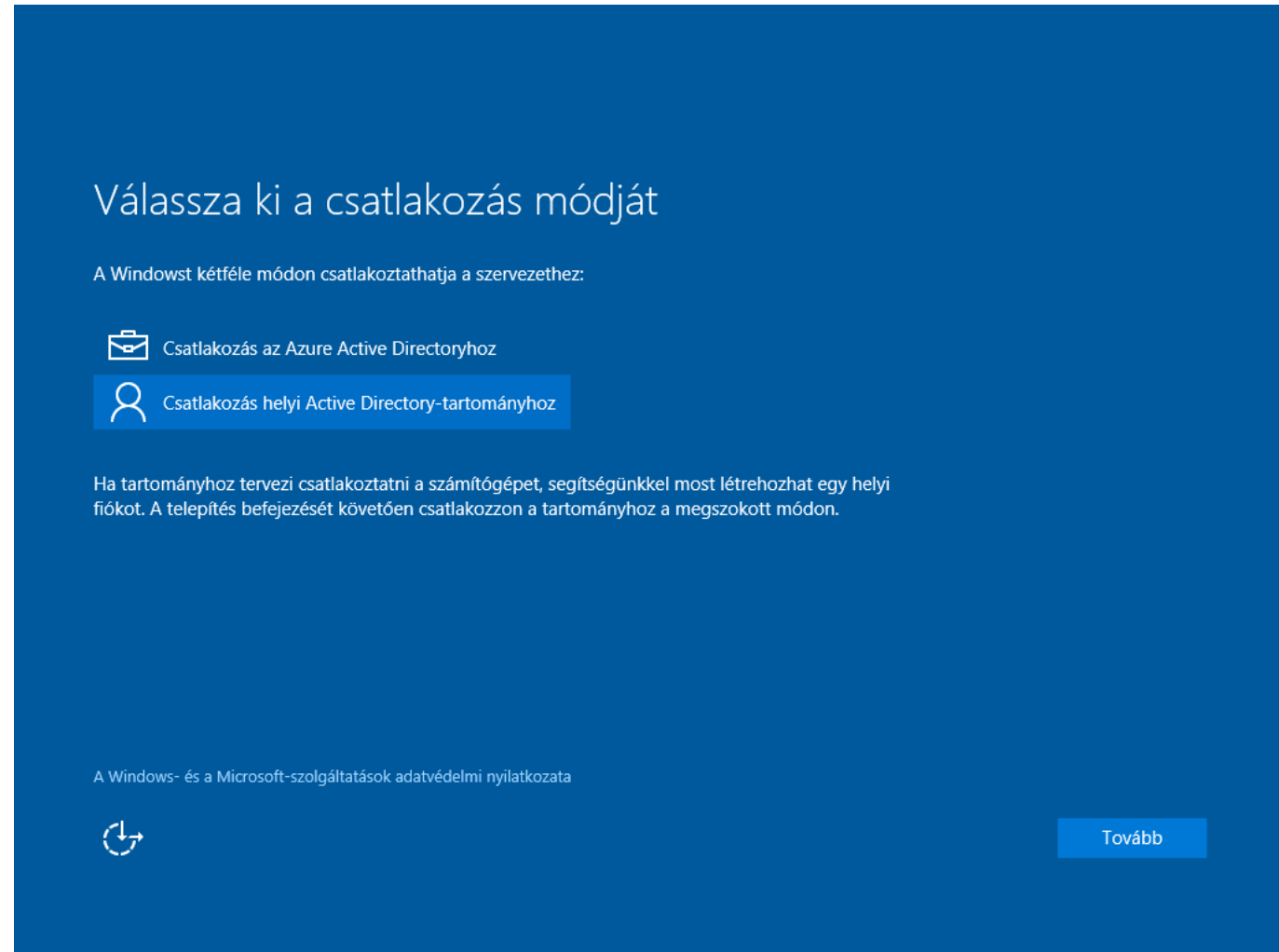
GYIK (Windows 10 Pro): valahonnan

- Mi az a Microsoft account?
 - A MS fiók más szolgáltatókhoz hasonlóan egyetlen központi fiókhoz rendeli valamennyi szolgáltatását.
 - Windows Store (alkalmazás bolt), OneDrive, Office 365, MSDN, Azure, Skype, etc.
 - Ha Android telefon lapul a zsebedben, akkor Google fiókod is bizonyára van, mert a Google Play áruházban megrendelések is ahhoz vannak kötve.
- E nélkül már nem is telepíthetitek Windowst?
 - De, lehet helyi fiókkal is telepíteni, ha semmilyen felhős szolgáltatást nem akar használni az ember
 - Így is lehet, csak lépten-nyomon hitelesítenie kell magát az embernek (jelszavak), ahelyett hogy a munkafolyamat része lenne a hitelesítő adat
 - A zárolási képernyő feloldása hitelesíti a felhasználót



Windows telepítőn átszaladás

- Válasszuk ki a helyi AD tartományt
 - Evvel későbbi intézkedésig a számítógép maga lesz egy egy-személyes AD tartomány



Windows telepítőn átszaladás

- Nem túl bonyolult

Fiók létrehozása ehhez a géphez

Ha jelszót szeretne használni, az legyen könnyen megjegyezhető, de mások által nehezen kitalálható.

Ki fogja használni ezt a gépet?

Felhasználónév

Figyeljen a biztonságra.

Adja meg a jelszót

Írja be újra a jelszót

Jelszó-emlékeztető

 [Vissza](#) [Tovább](#)



Windows telepítőn átszaladás

- Fehér ember várni homokóra



Folyamatban van a beállítás

Windows telepítőn átszaladás

- Fehér ember várni homokóra



Kis türelmet

Alkalmazások telepítése



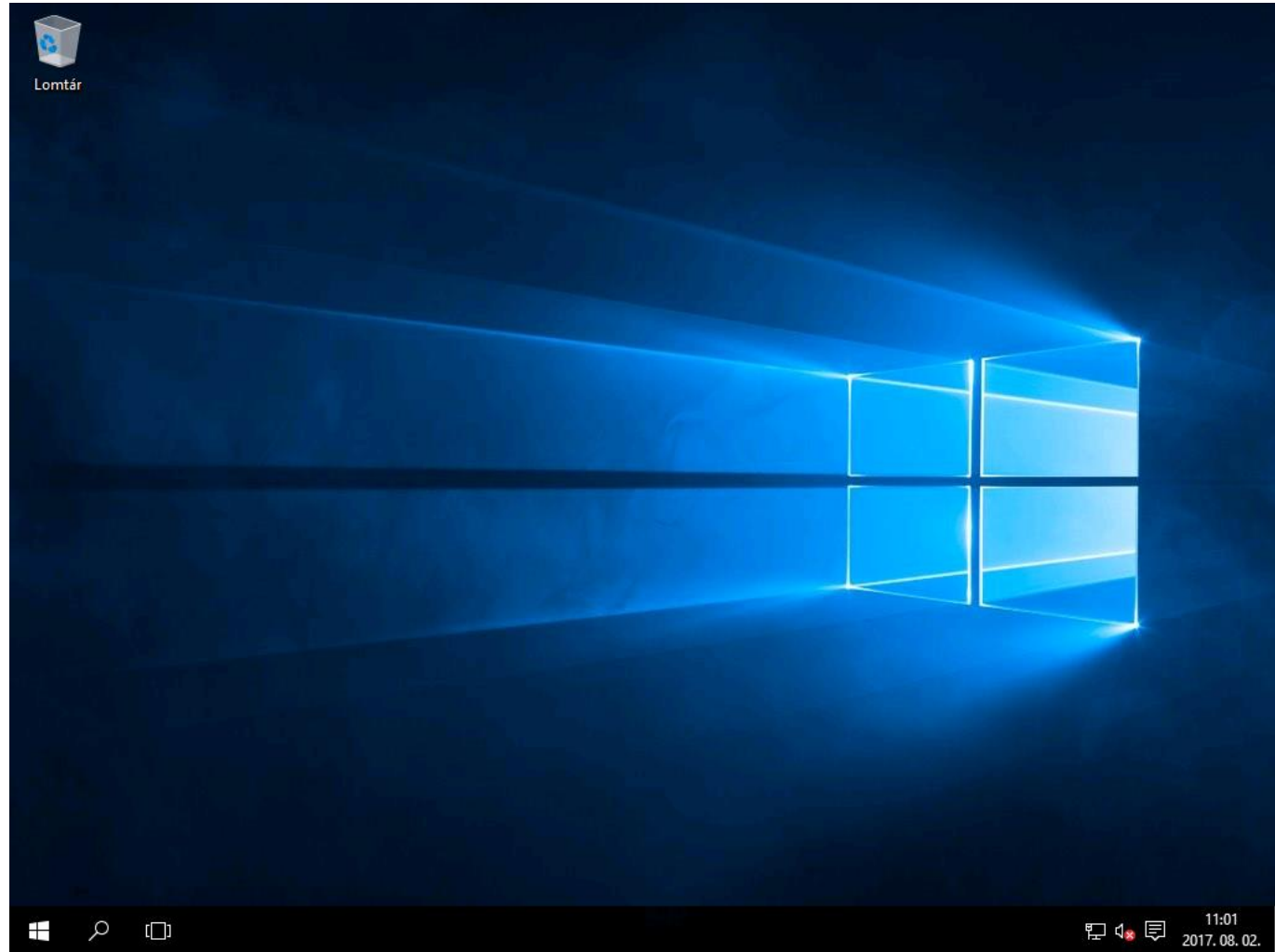
Windows telepítőn átszaladás



Kezdődhet valami új

És lőn világosság!

- A telepítőt követően a letisztult Windows logó fogad bennünket egy üres asztallal
 - A kép egyébként valódi fénykép és így készült





Lakjuk be a rendszert

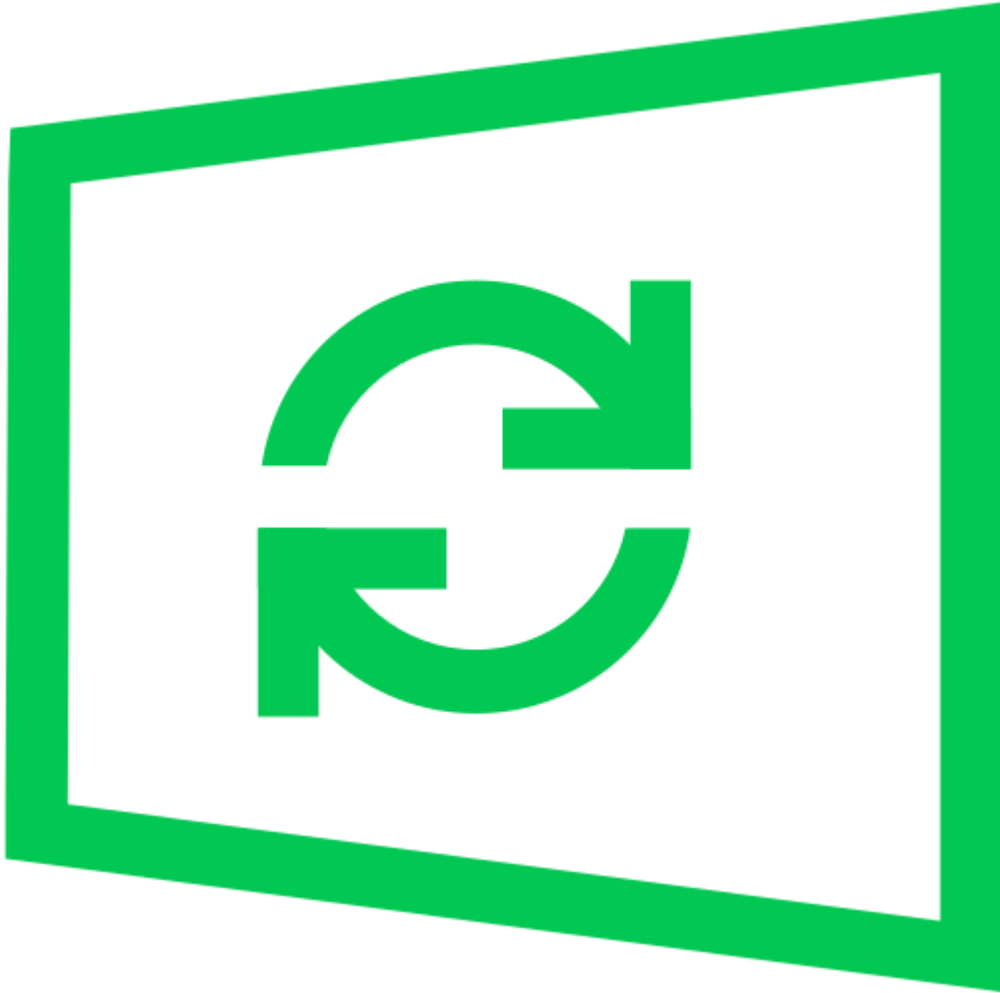
Tegyük egy kicsit otthonossá

Telepítsünk meghajtókat

- Kaptam egy halom meghajtó (driver) telepítő lemezt az alaplapomhoz, videokártyámhoz, stb.
- Biztos jó lesz nekem, ha feltelepítem ezeket.



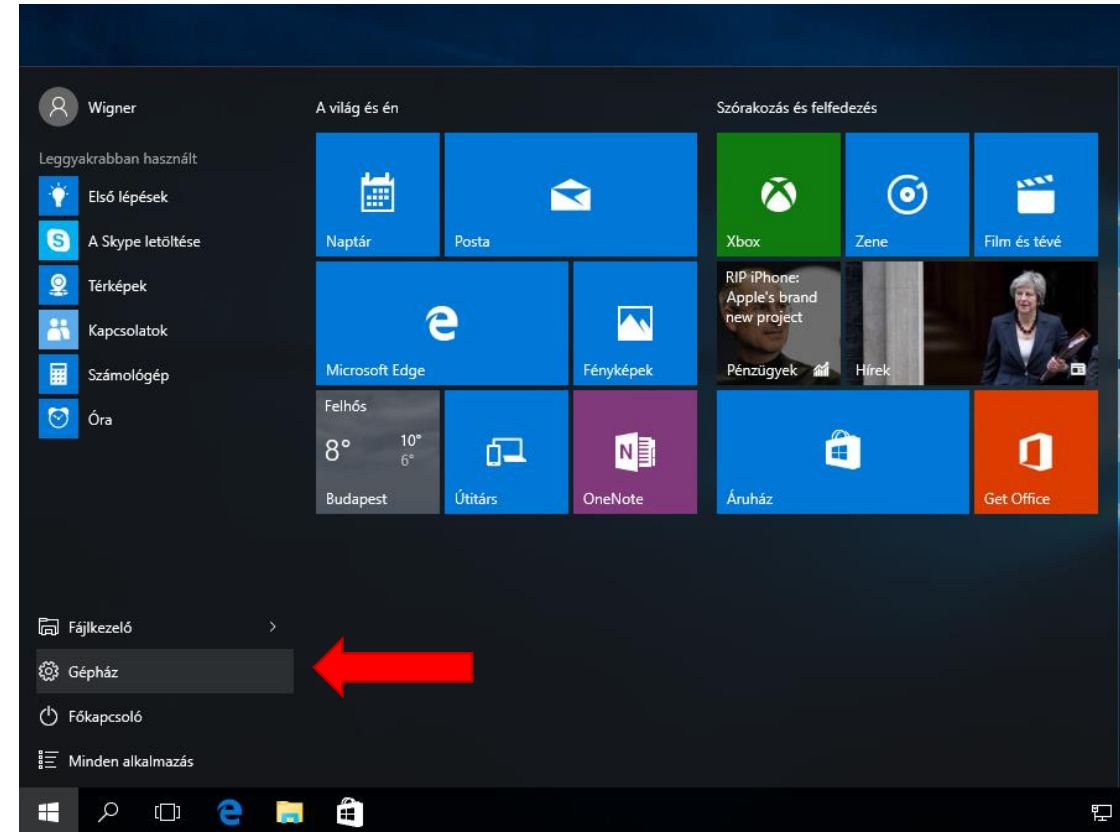
Windows Update

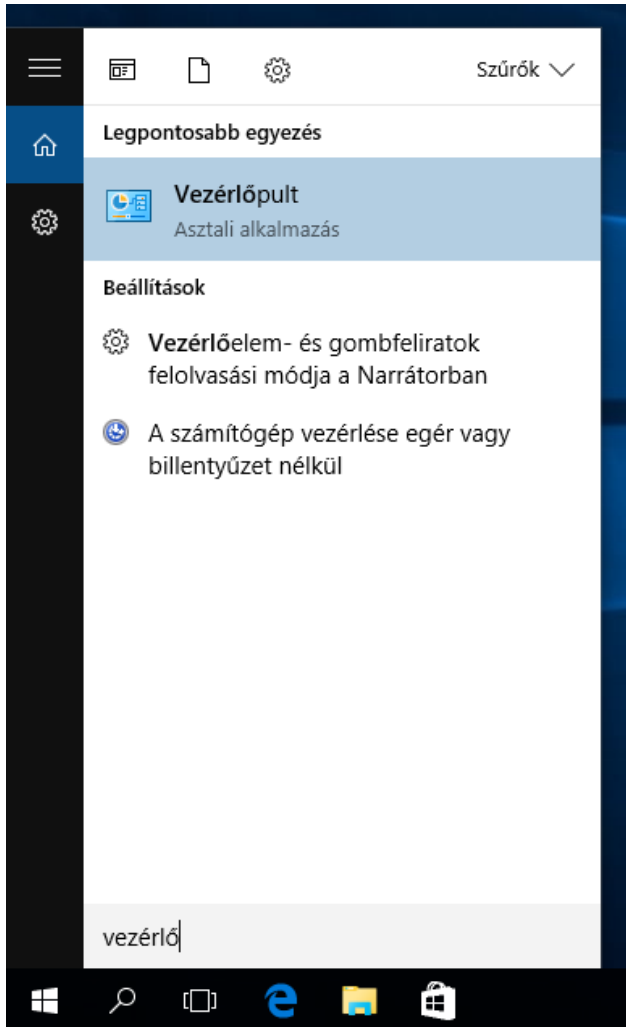


- A Windows Update amit valamiért egyesek csípőből kikapcsolnak minden^[1] eszökhöz feltelepíti a legújabb^[2] eszökhözilllesztöt
 - [1]: ha nem sokkal újabb a hardver, mint a telepítő
 - [2]: amit a gyártó elérhetővé tett a Microsoft számára
- Biztonsági és nagyobb rendszer frissítések is innen jönnek
 - Ezek késleltethetők (maximum 1 évig?), de tovább nem, mert biztonsági kockázattal jár)



- Ellenőrizzük, hogy minden frissítést sikerült-e telepítés során felrakni!
- Nyissuk meg a Gépházat!
 - „Start”-gomb, Fogaskerék
 - „Start”-gomb + i
- De mi a fene az a Gépház?

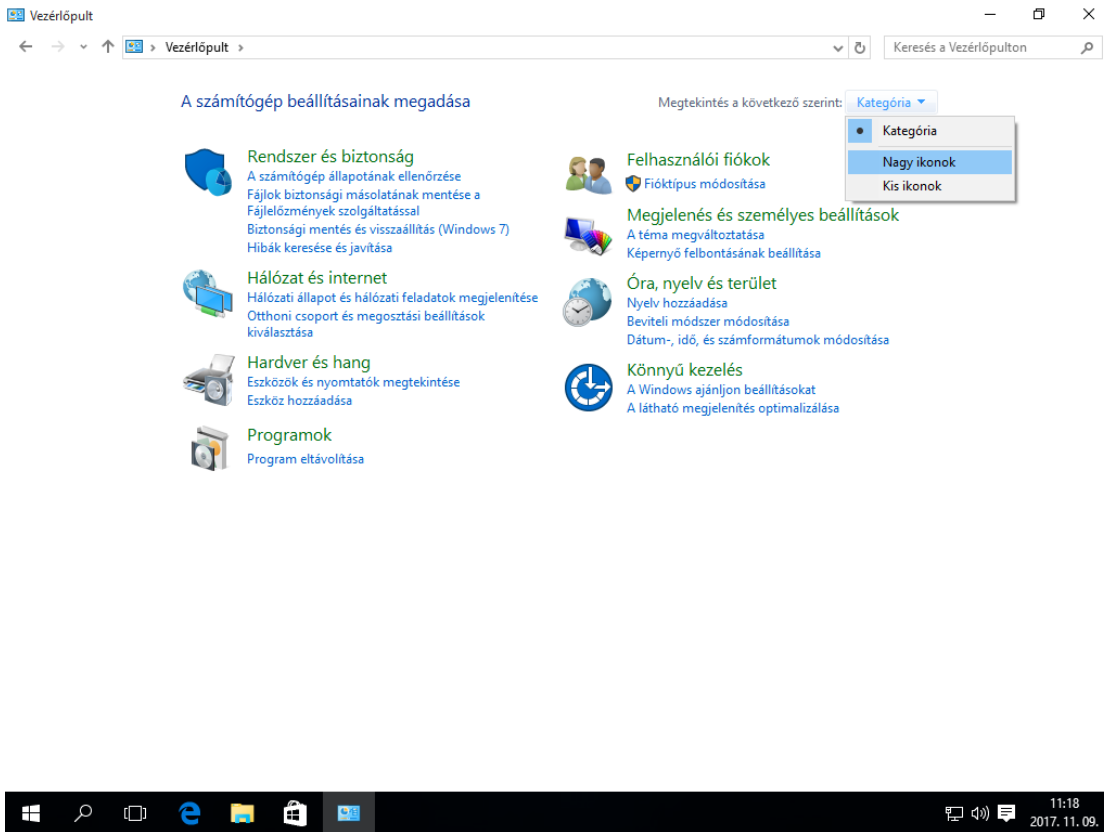




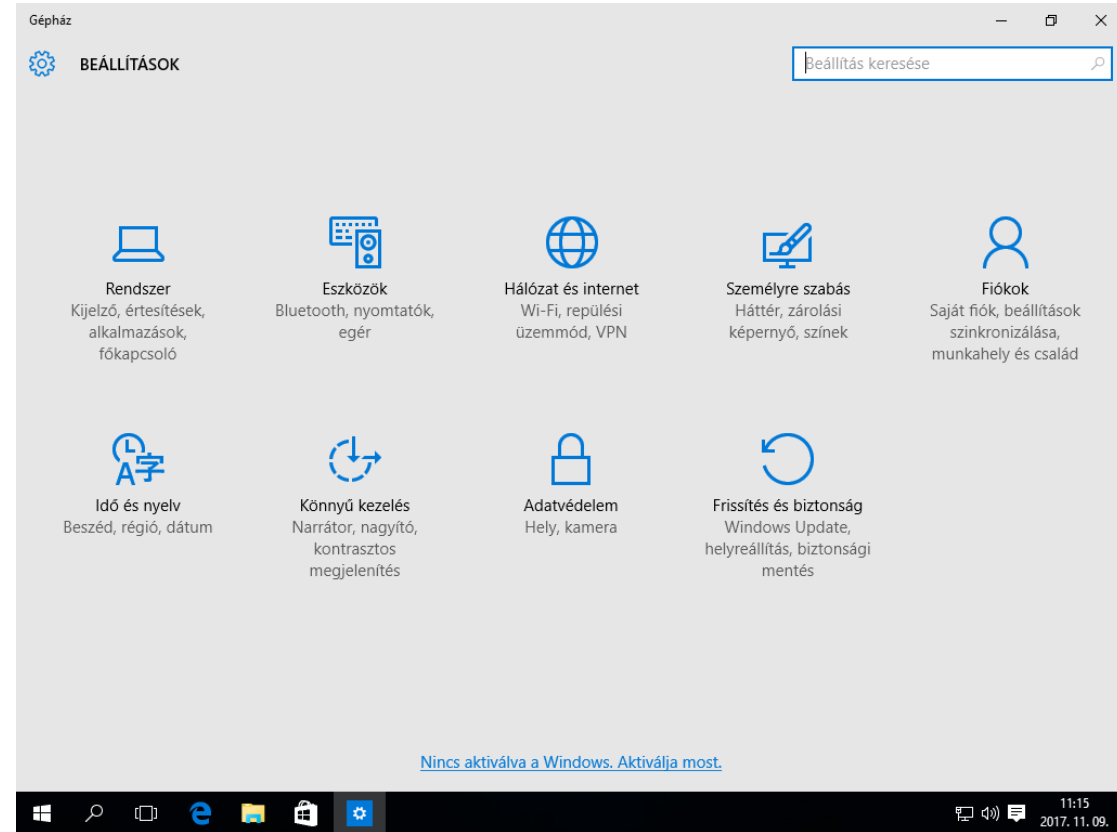
- De hová tűnt a régi jó Vezérlőpult?
 - Ott van az még mindig, csak kicsit jobban el van dugva
- „Start”-gomb és kezdjük el gépelni: vezérlő...

Something old, something new

Vezérlőpult



Gépház



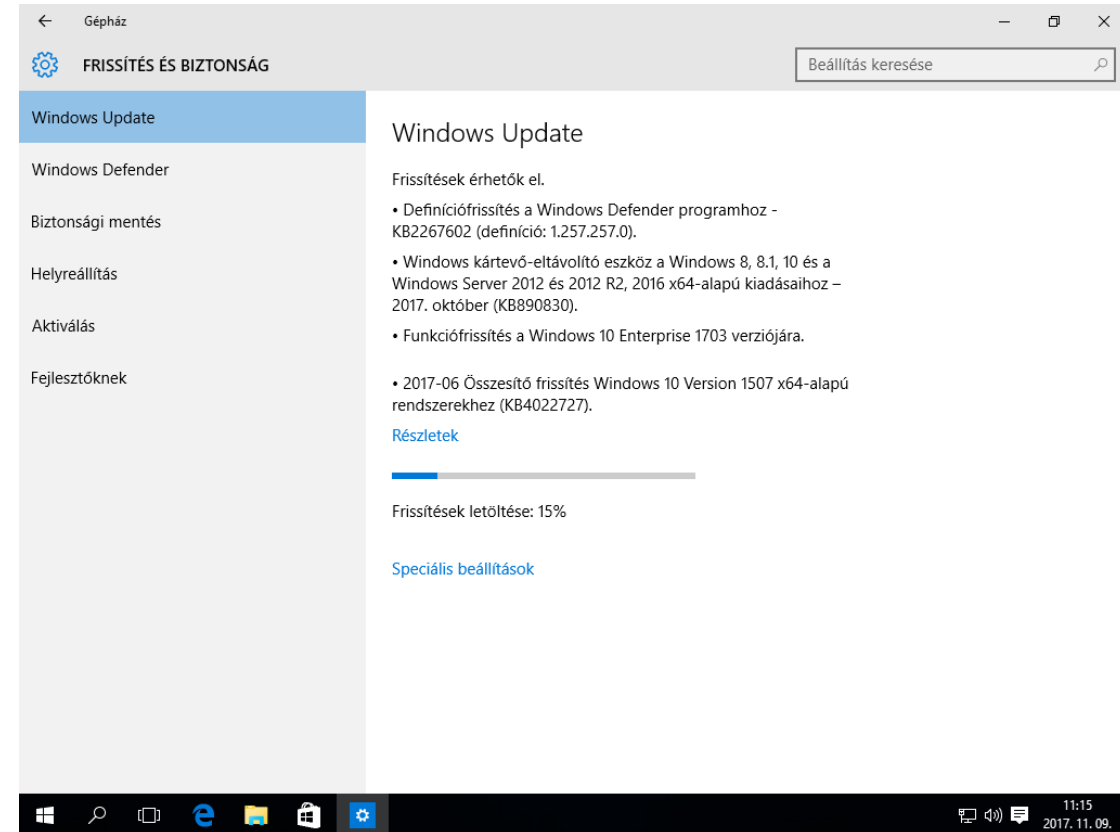
A két arcú Windows

- A Windows egy pokolian lassú átalakuláson megy keresztül
- A Windows 95 óta használt WinAPI-ra épülő WinForms alapú GUI technológiát apránként váltja a WinRT-re (Windows Runtime) épülő WPF (Windows Presentation Foundation) alapú GUI
- Miért van erre szükség? („Miért idegesítenek ezzel?”)
 - „A változatosság gyönyörködtet!”
 - Ami 1995-ben működött, az 2017-ben már nem
 - A mobil platformok előtérbe kerülésével egy adaptív kezelőfelület kell
- Miért nincs már kész? („Vagy ez, vagy az!”)
 - Mert minden változásra így is fellázadnak az emberek (csempék)
 - Mert maguk sem tudják, mit akarnak pontosan (lásd előző pont)



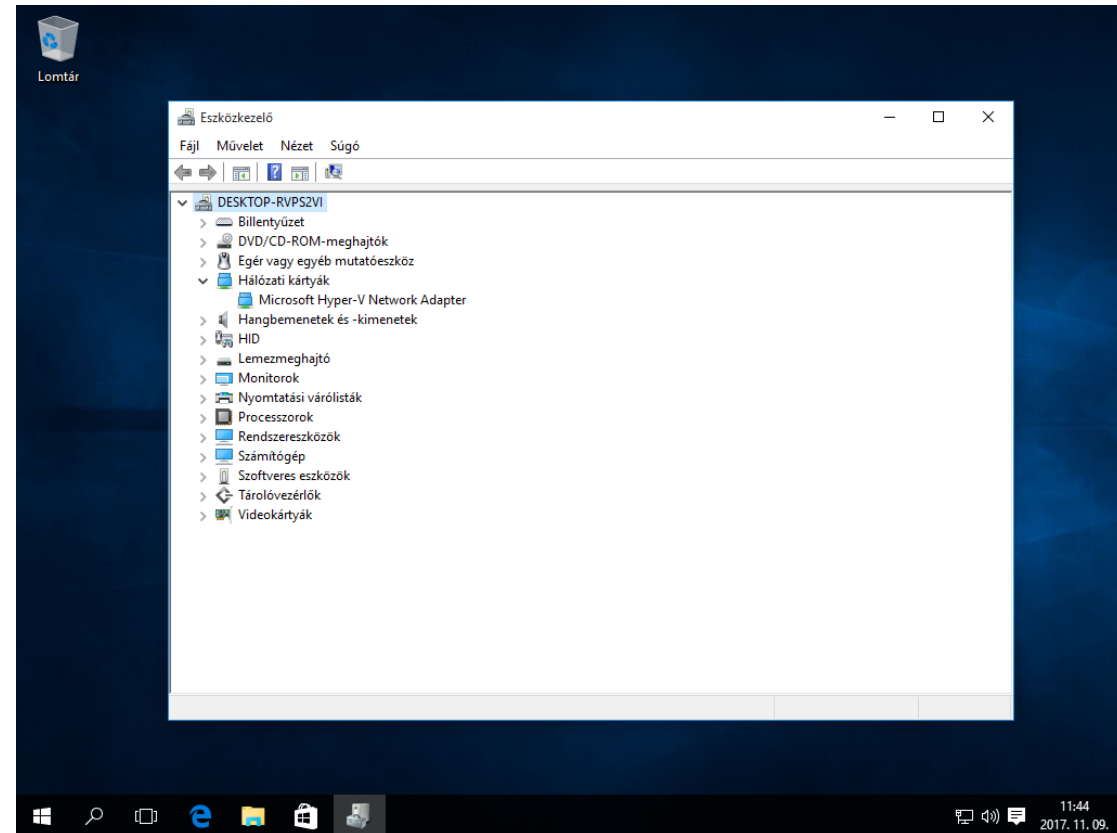
Windows Update

- Nyissuk meg a „Frissítés és biztonság” menü pontot
- Attól függően, hogy:
 - mikori a telepítőnk
 - telepítéskor felismert-e már a hálózati kártyát
 - hajlandóak voltunk-e felereszteni az internetre a telepítőt
- mást tapasztalhatunk
- Legjobb esetben minden kész



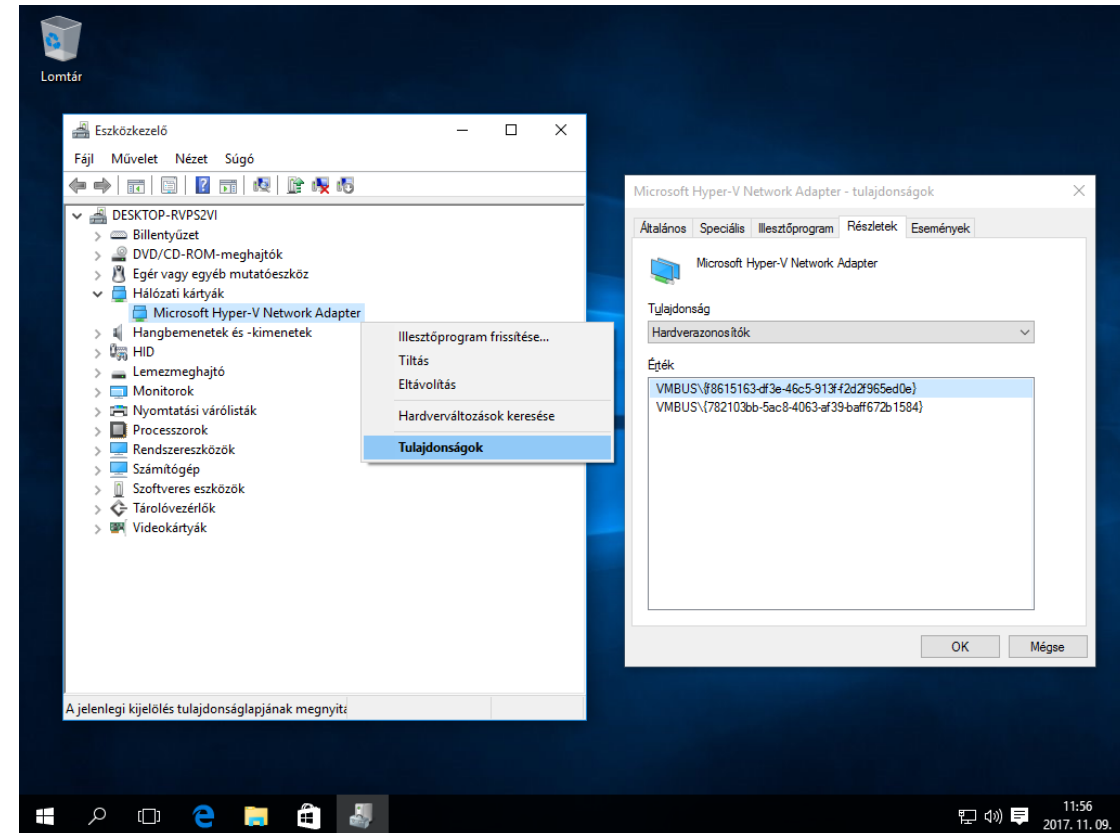
Telepítsünk meghajtókat

- Ellenőrizzük, hogy minden rendben települt-e!
 - „Start”-gombra jobb-klikk
 - Eszközkezelő
 - (Vezérlőpultról is nyitható)
- Keressünk sárga felkiáltó jeleket az eszközök ikonjai mellett
- Ha van bárhol, akkor...



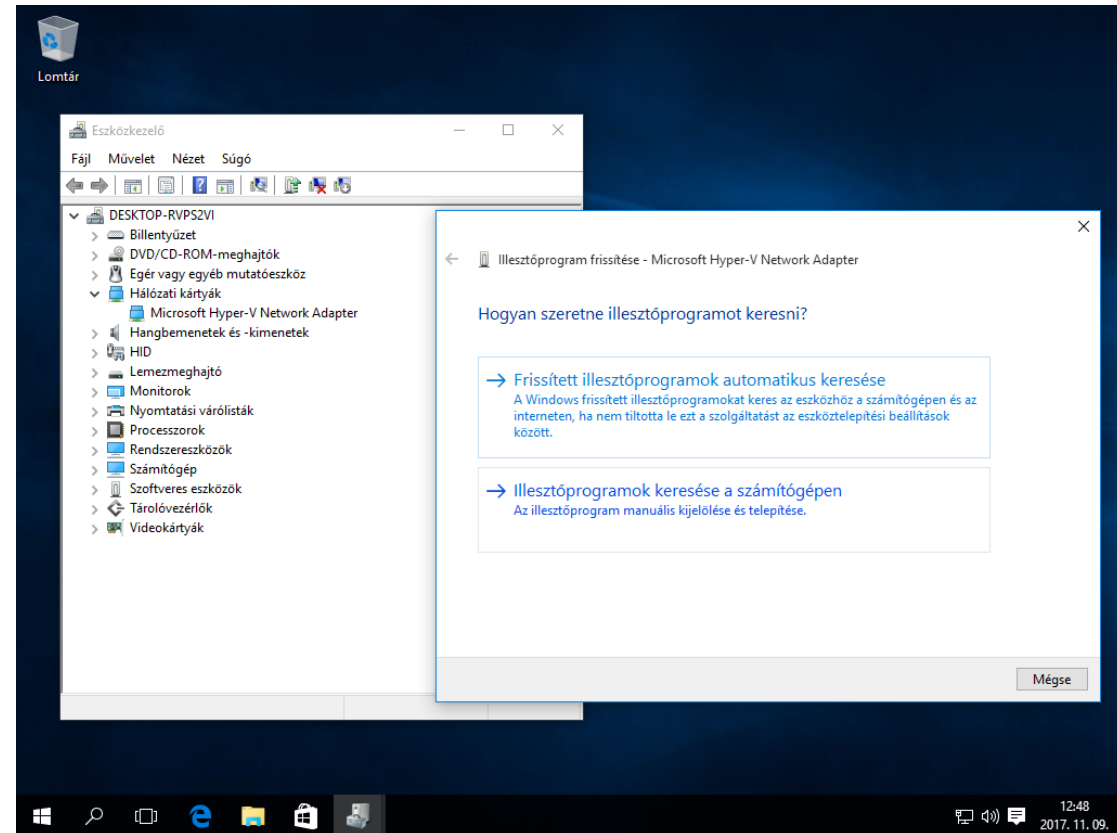
Telepítsünk meghajtókat

- Jobb-klikkeljünk a felkiáltó jellel ellátott eszközre
 - Tuljdsónságok
 - Részletek
 - Legördülő menüből „Hardverazonosítók”
- Az itt látható karakter sorozatokat egy webes keresőbe beütve megtalálhatjuk mi is az adott eszköz



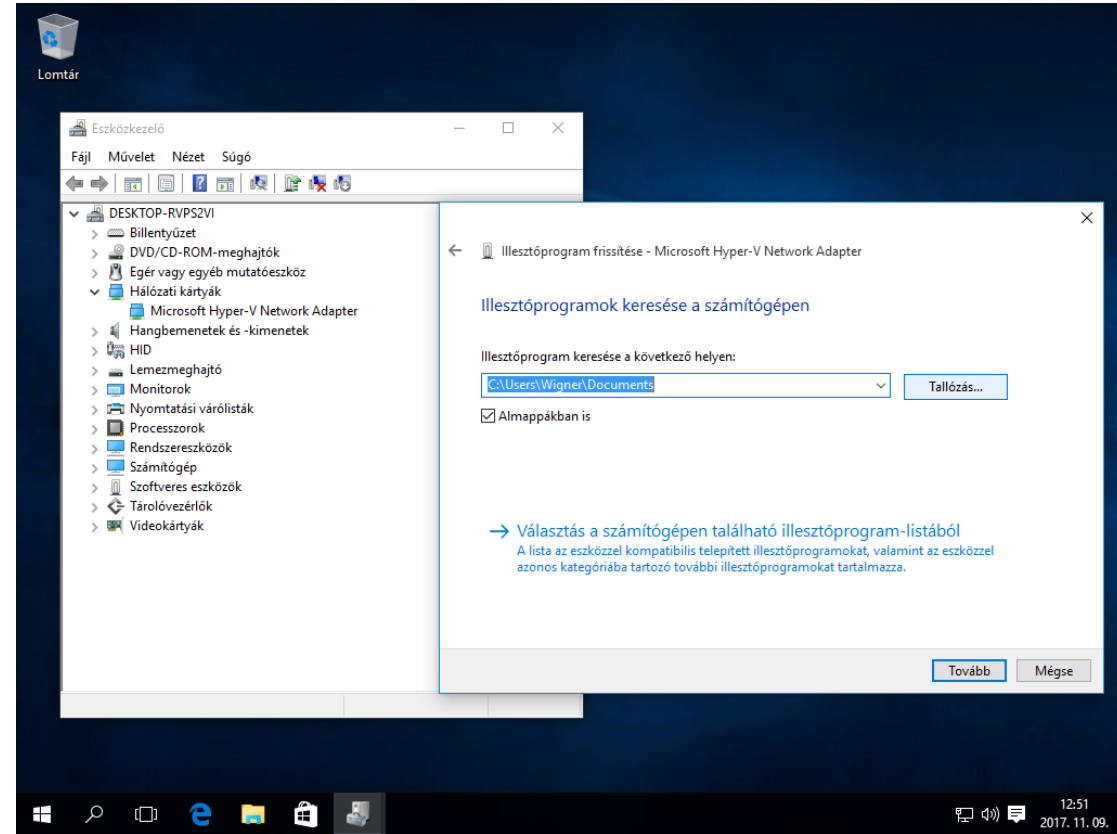
Telepítsünk meghajtókat

- Ha kaptunk hozzá lemezt, tegyük be
- Ha nem, a gyártó honlapjáról szerezzük be a meghajtót
- **Tartózkodjunk** a
 - meghajtó kezelő programoktól!
 - harmadik féltől származó meghajtótól!
- Jobb-klikk
 - Illesztőprogram frissítése...



Telepítsünk meghajtókat

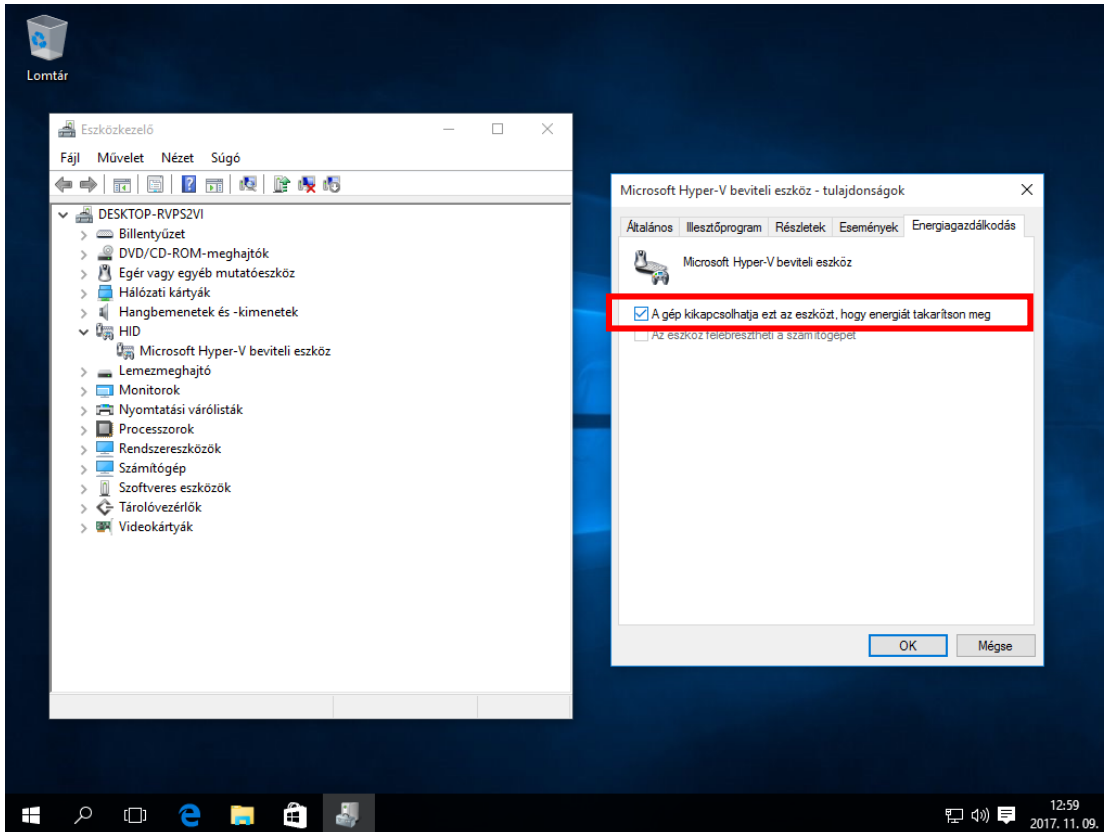
- A „Tallózás...” kattintva irányítsuk valami ésszerűen közeli helyre, aminek almappájában ott lesz a meghajtó
 - .inf kiterjesztésű fájlokat fog keresni (metaadatok), a meghajtó pedig .cab vagy .bin fájlban lesz benne



Trükk - energiagazdálkodás



- Egyes meghajtók tévesen (nem) jelzik az operációs rendszer felé, mikor aktívak és mikor nem
- Ennek következtében működés közben is kikapcsolhatja az eszközt a rendszer
 - Eldobja a hálózatot, bluetooth eszközt, stb.
- Jobb-klikk,
 - Tulajdonságok
 - Energiagazdálkodás



Telepítsünk vírusírtót

- Windows gépre természetesen kell valami tisztességes vírusírtó
- Szerencsére
 - előtelepítve kaptam (bloatware)
 - van már egy kedvencem
- De tényleg kell?



Windows Defender



- Beépített vírus és kártevő kergető program
- „Normális” használat mellett tökéletesen megfelel
 - Nem konferencia előadói gép
 - Nem látogat az ember kétes megbízhatóságú weboldalakat
 - Nem tör/crackel programokat
- Ingyenes, észrevétlenül frissül
 - Windows Update!



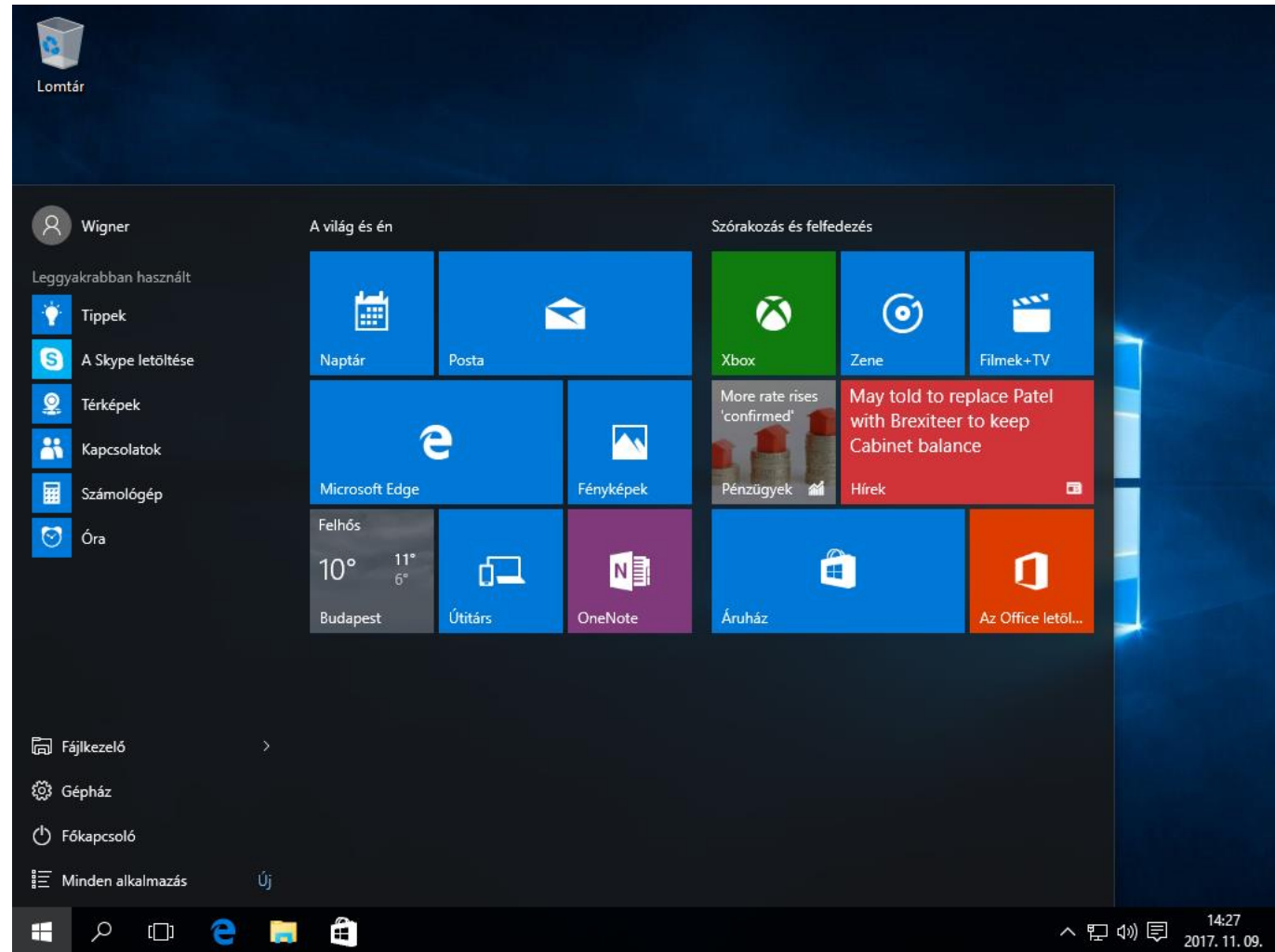
Eduroam és Wigner SSID

- Az Eduroam és Wigner SSID-jű WiFi hálózatok beállítása jobbra változatlan
 - A nagyon részletes WiFi beállítások maradtak a Vezérlőpulton
 - eduroam.kfki.hu



App a lelke mindennek

- Csempék, appok egy asztali gépen?
 - Mi?
 - Hogy?
 - Miért?
- Aki hiányolja a régi „Minden program” nézetet, az most a „Minden alkalmazás” alatt találja meg



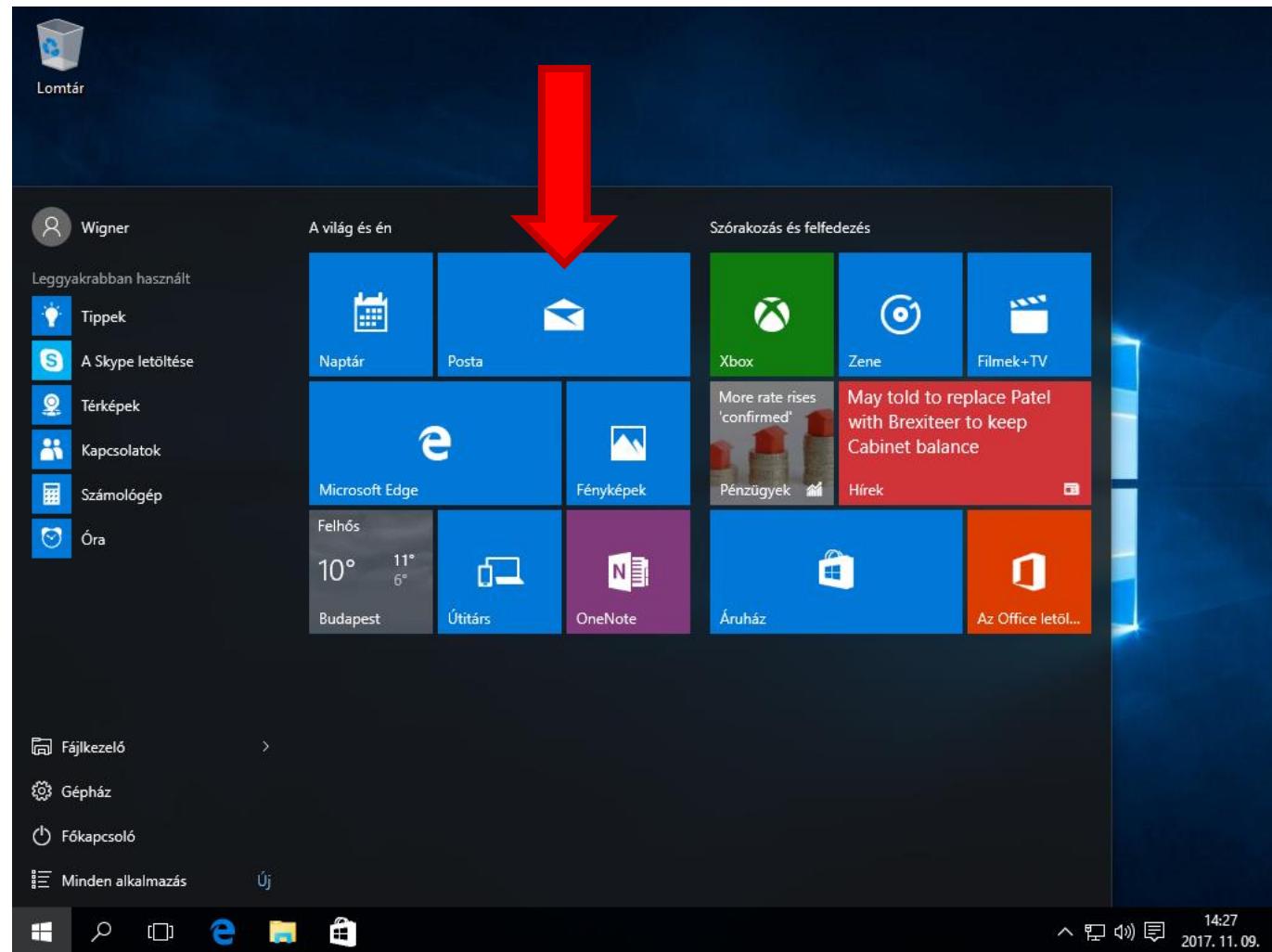
App a lelke mindennek

- A Microsoft mobil törekvései több szempontból tekinthetők úttörőnek
 - Miért merülnek gyorsan a mobil eszközök akkumulátorai?
 - Race to sleep, nem szabad folyamatosan futni
 - Új application model nagyon szigorú feltételeket ró az appokra, mikor jutnak erőforráshoz
 - Hogyan lehet vállalati szintű biztonságot garantálni a végfelhasználók részére?
 - Az új application model számol az untrusted application létevel (erre épül a Microsoft alkalmazásboltja), ami minden appot egy sandboxban futtat
 - 5 éves bevezetése óta feltöretlen
 - Miért lassul le a Windows idővel?
 - A jelenség „Windows rot” névre hallgat és a [Windows Registry](#) hízásának tudható be
 - Kizárólag appok használata mellett a Windows nem lassulna

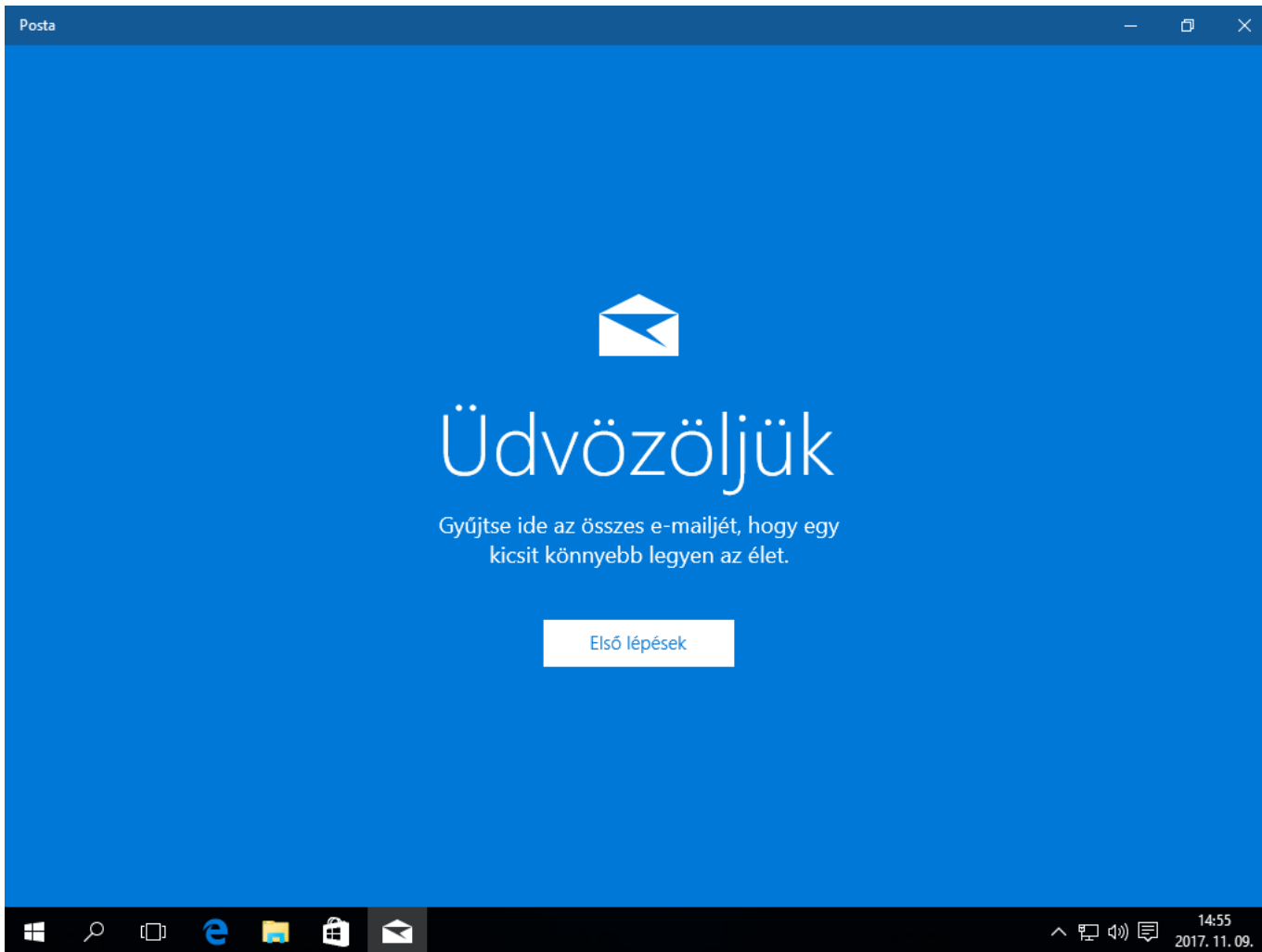


App a lelke mindennek

- Akkor miért jó?
 - Biztonságos
 - Gyorsabb marad tőle a gép
 - Kényelmes telepíteni
 - Nem zaklat frissítésekkel
 - Csak csinálja
- Akkor miért ne használjam?

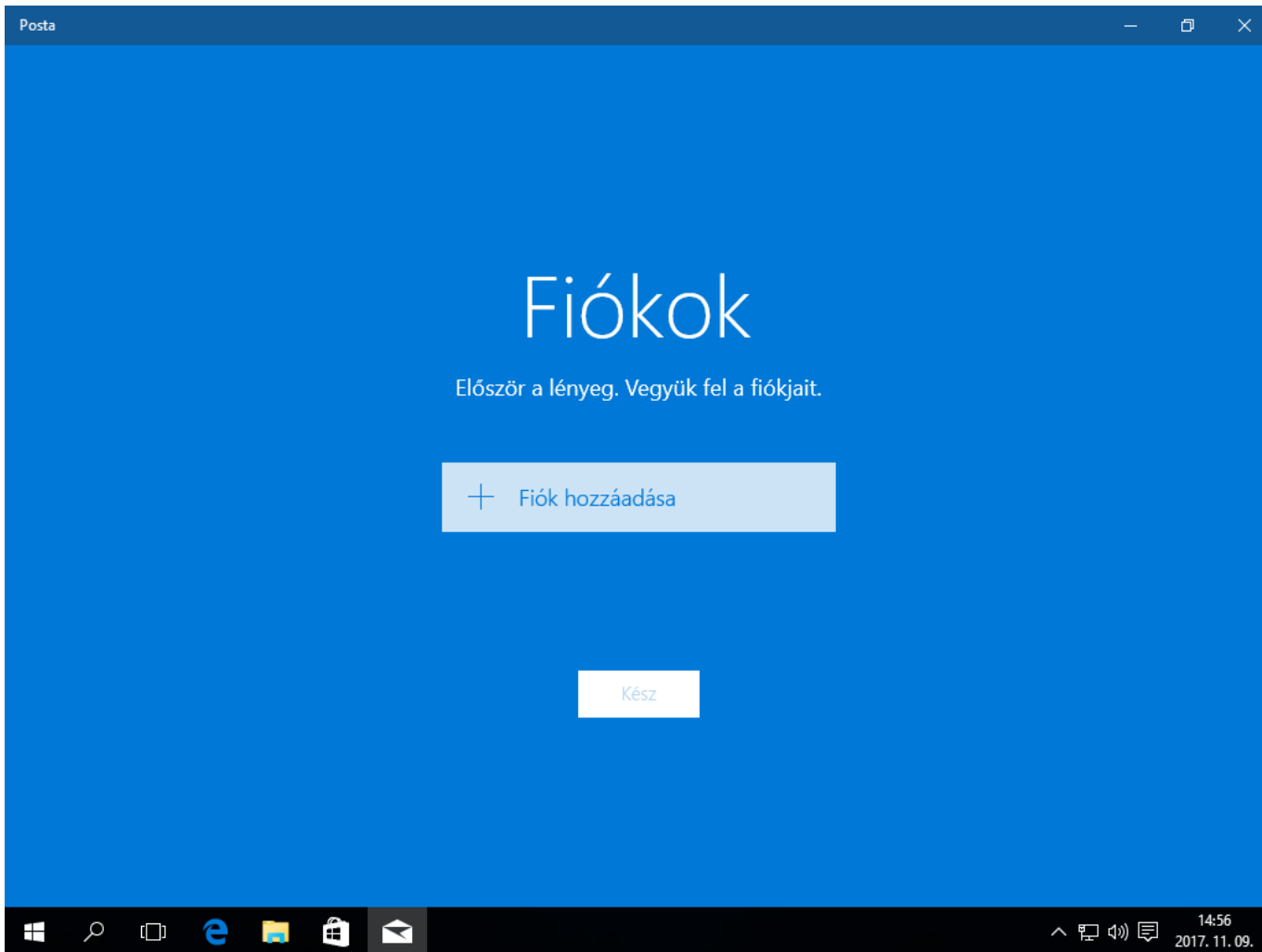


Posta app



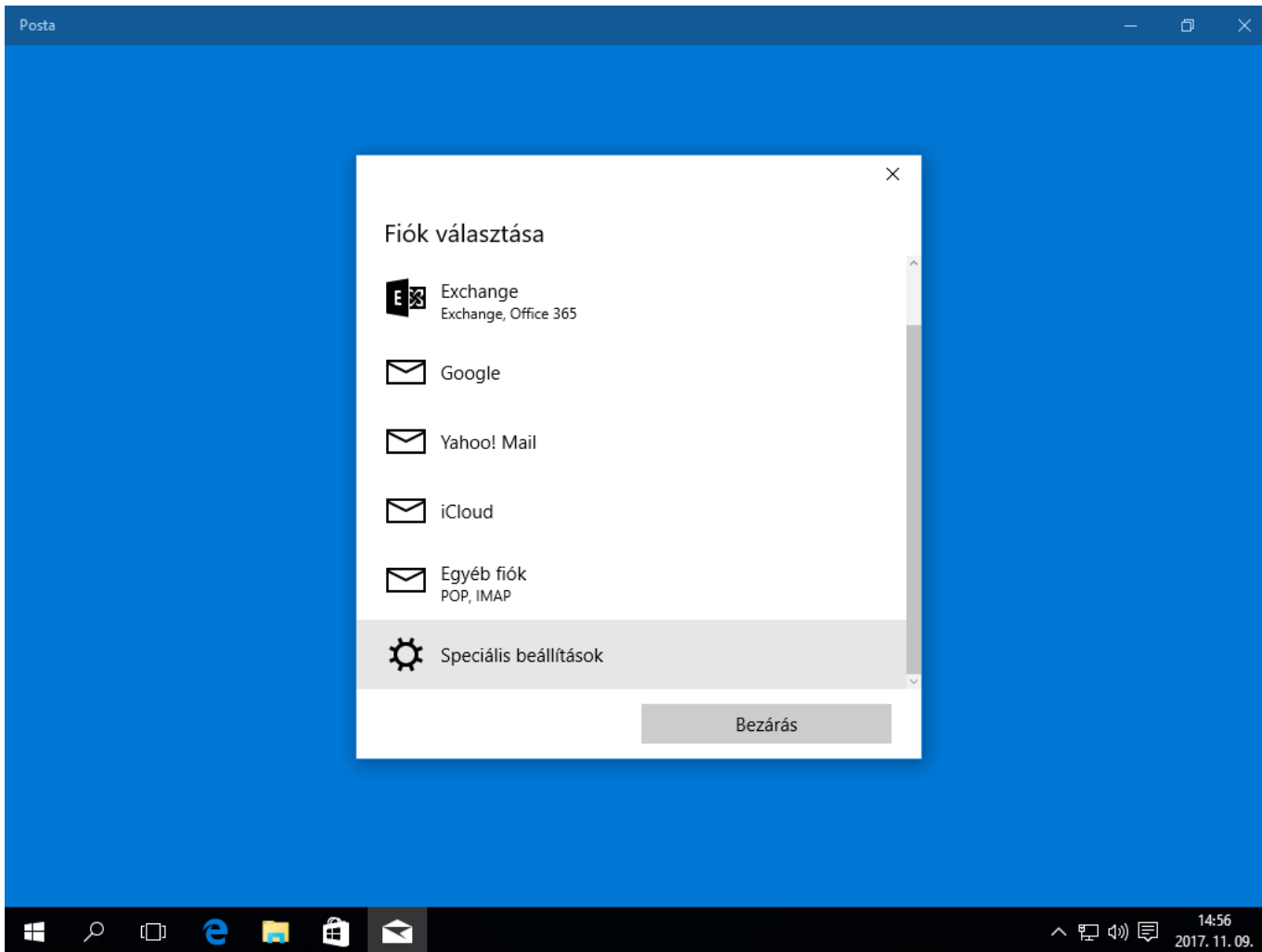
- Szép letisztult
- Beállításhoz
 - Első lépések

Posta app



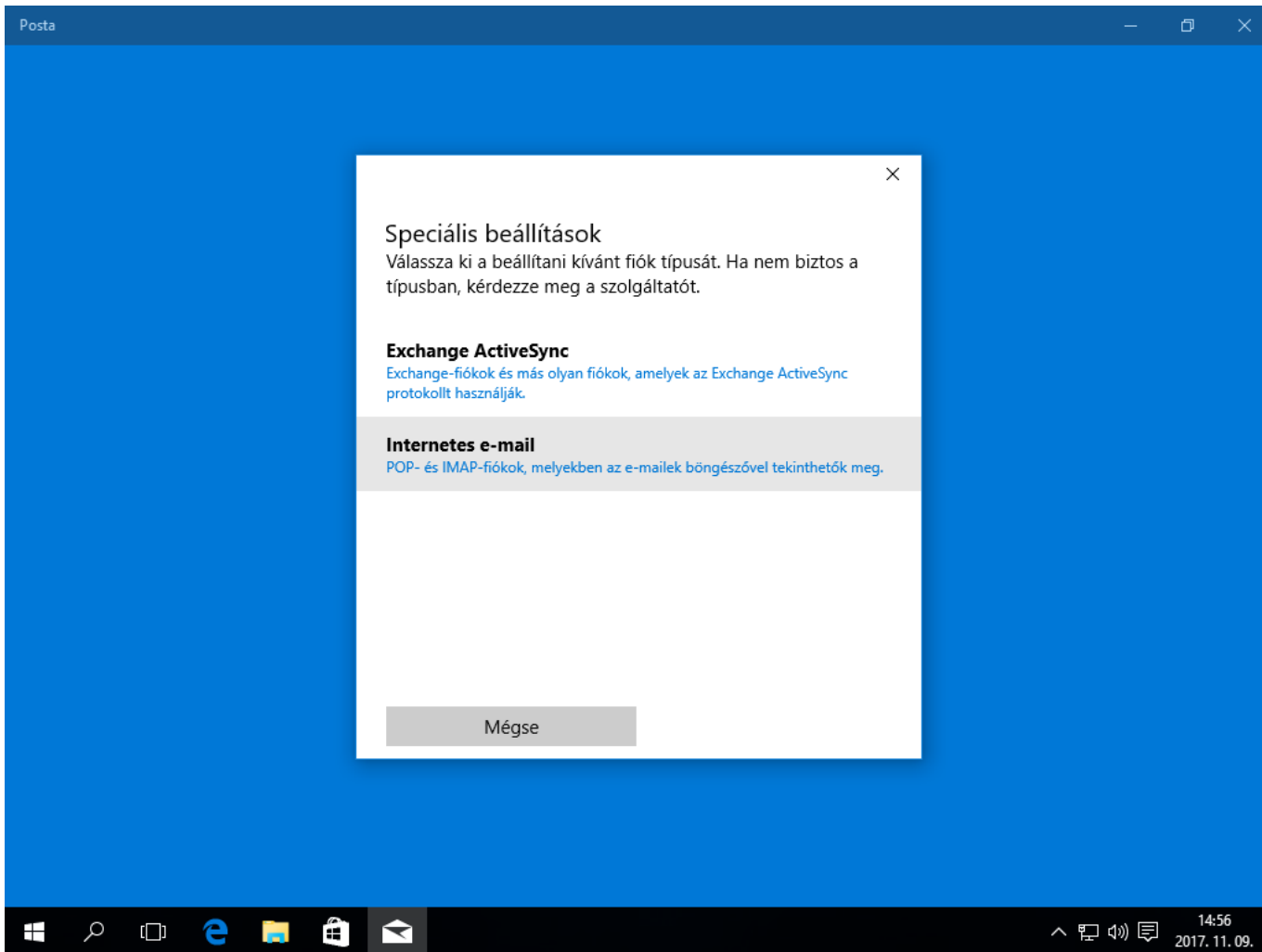
- Ha az ember Microsoft Accounttal hozta létre a felhasználóját, akkor az ahhoz tartozó fiók alapból itt van
- Mi most hozzáadunk egy wigner-es e-mail címet

Posta app



- A levelező kliens automatikusan tud konfigurálni egy pár divatos e-mail szolgáltatót
 - outlook.com
 - Google
 - Yahoo!
 - iCloud
 - Exchange fiókok
- Mi töviről hegyire kézzel állítunk mindent

Posta app



- A Wigner FK-nak nincs Exchange szolgáltatása
 - olyan mint a hagyományos IMAP, pár extrával
- Mi mezei IMAP (Internet Mail Access Protocol) fiókot állítunk be

Posta app



Posta

Internetes e-mail fiók

Fióknév

Név

Ez a név lesz üzeneteiben a feladó.

Bejövő e-mailek kiszolgálója

Fiók típusa

E-mail cím

Felhasználónév

Példák: valaki, valaki@contoso.com, tartománynév\valaki

Jelszó

Mégse Bejelentkezés

15:01
2017. 11. 09.

- Beállítunk
 - Valami beszédes fiók nevet
 - Feladó neve
 - Bejövő levelek kiszolgálója
 - Protokoll
 - Mail cím
 - Felhasználónév
 - Ez a UNIX felhasználónevünk (lxserv például)

Posta app



Posta

Internetes e-mail fiók

Jelszó

.....

Kimenő e-mailek kiszolgálója (SMTP)

smtp.wigner.mta.hu

☒ A kimenő e-mailek kiszolgálója hitelesítést igényel

☐ Ugyanazon felhasználónév és jelszó használata e-mailek küldéséhez

Bejelentkezési adatok a kimenő e-mailek kiszolgálójához

Felhasználónév

gipsz.jakab@wigner.mta.hu

Példák: valaki, valaki@contoso.com, tartománynév\valaki

Jelszó

.....

☒ SSL-kapcsolat megkövetelése a bejövő e-mailekhez

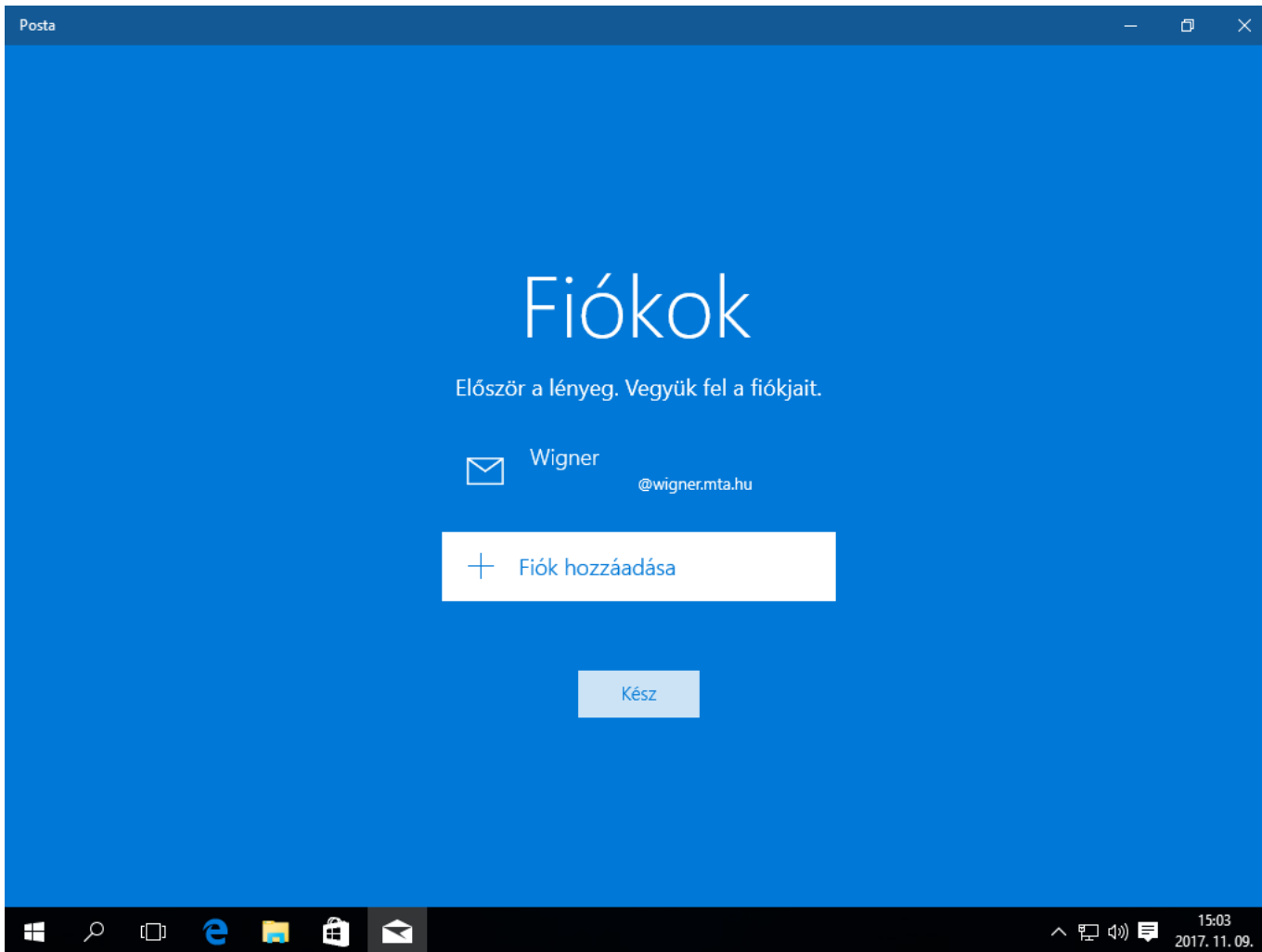
☒ SSL-kapcsolat megkövetelése a kimenő e-mailekhez

Mégse Bejelentkezés

15:02
2017. 11. 09.

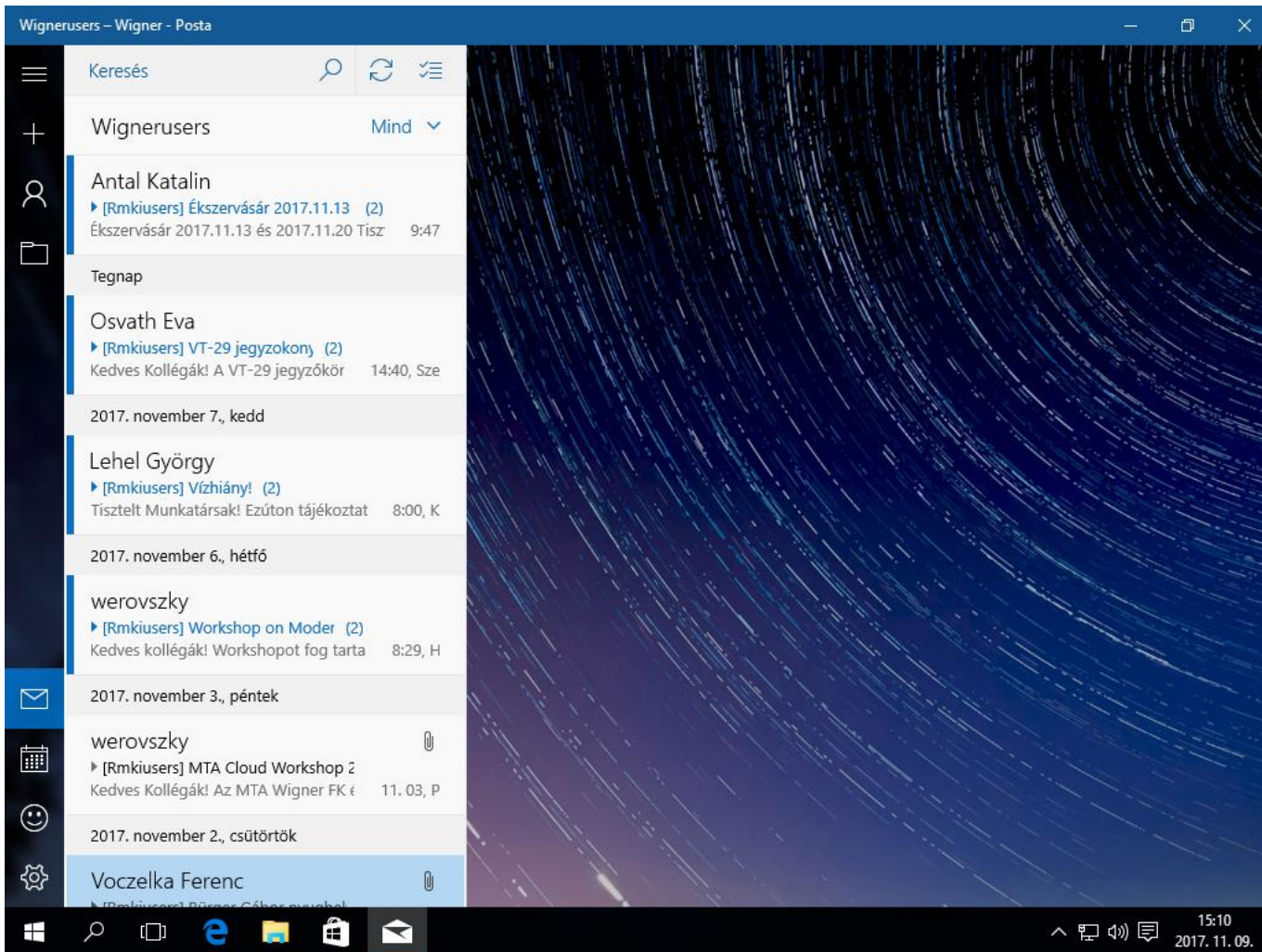
- Beállítunk
 - Hozzá tartozó jelszó
 - Kimenő levelek kiszolgálója
 - Teljes e-mailcím a felhasználónév
 - Spam szűrő jelszó
 - SSL-titkosítás ki/be

Posta app



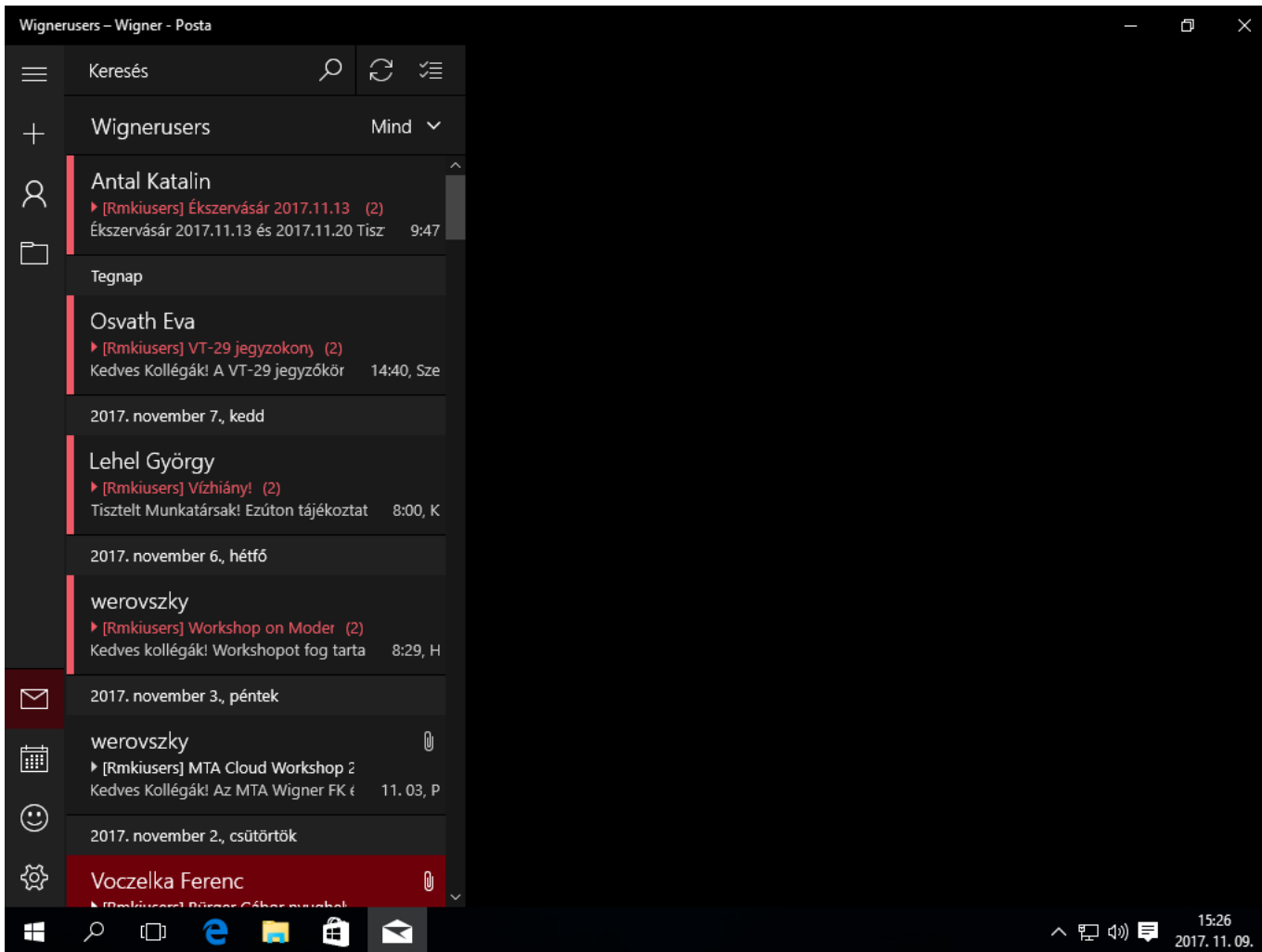
- Ha nem akarunk mást hozzáadni (például személyes Gmail fiókot, hogy ne kelljen azt is külön nézegetni)...
- „Kész”

Posta app



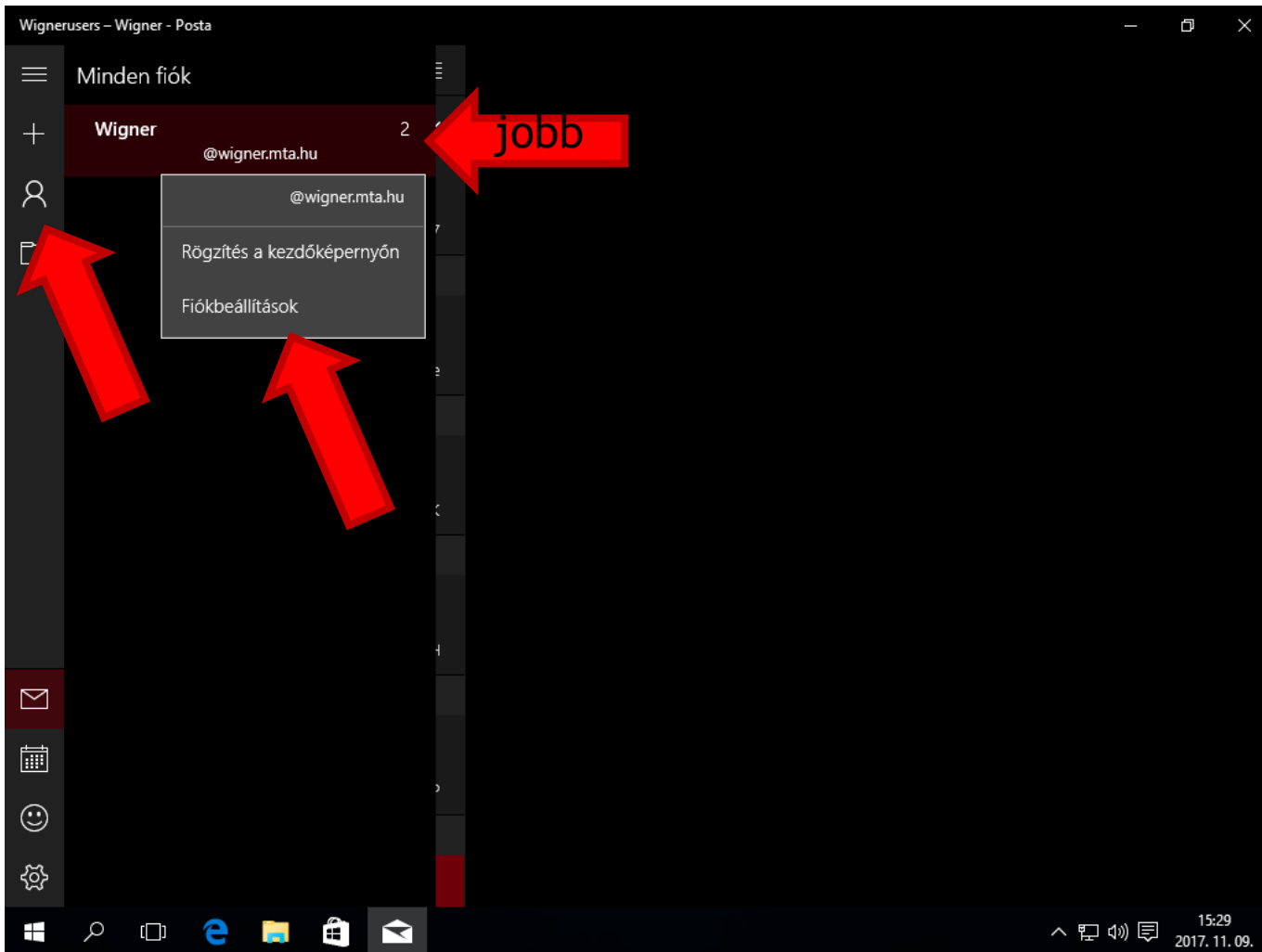
- Letisztult ablaktáblás nézet
- Testreszabható
 - Háttérkép
 - Kiemelőszínek
 - Sötét/világos téma
 - Levelek szinkronizálási beállítások
 - Milyen régi leveleket tartson meg offline?

Posta app



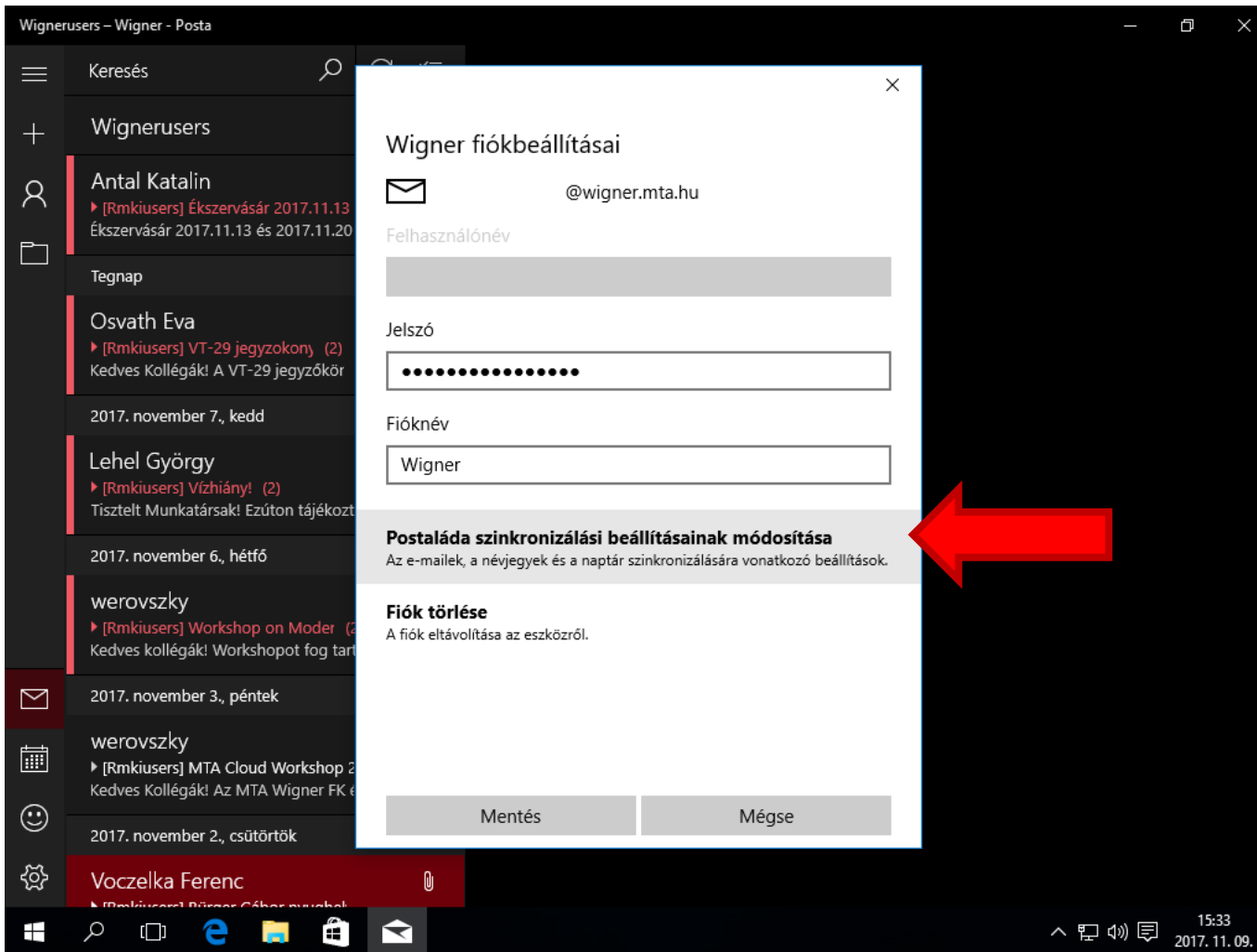
- Például
 - Sötét téma
 - Wigneres piros/vörös kiemelőszínnel
 - Háttér nélkül

Posta app



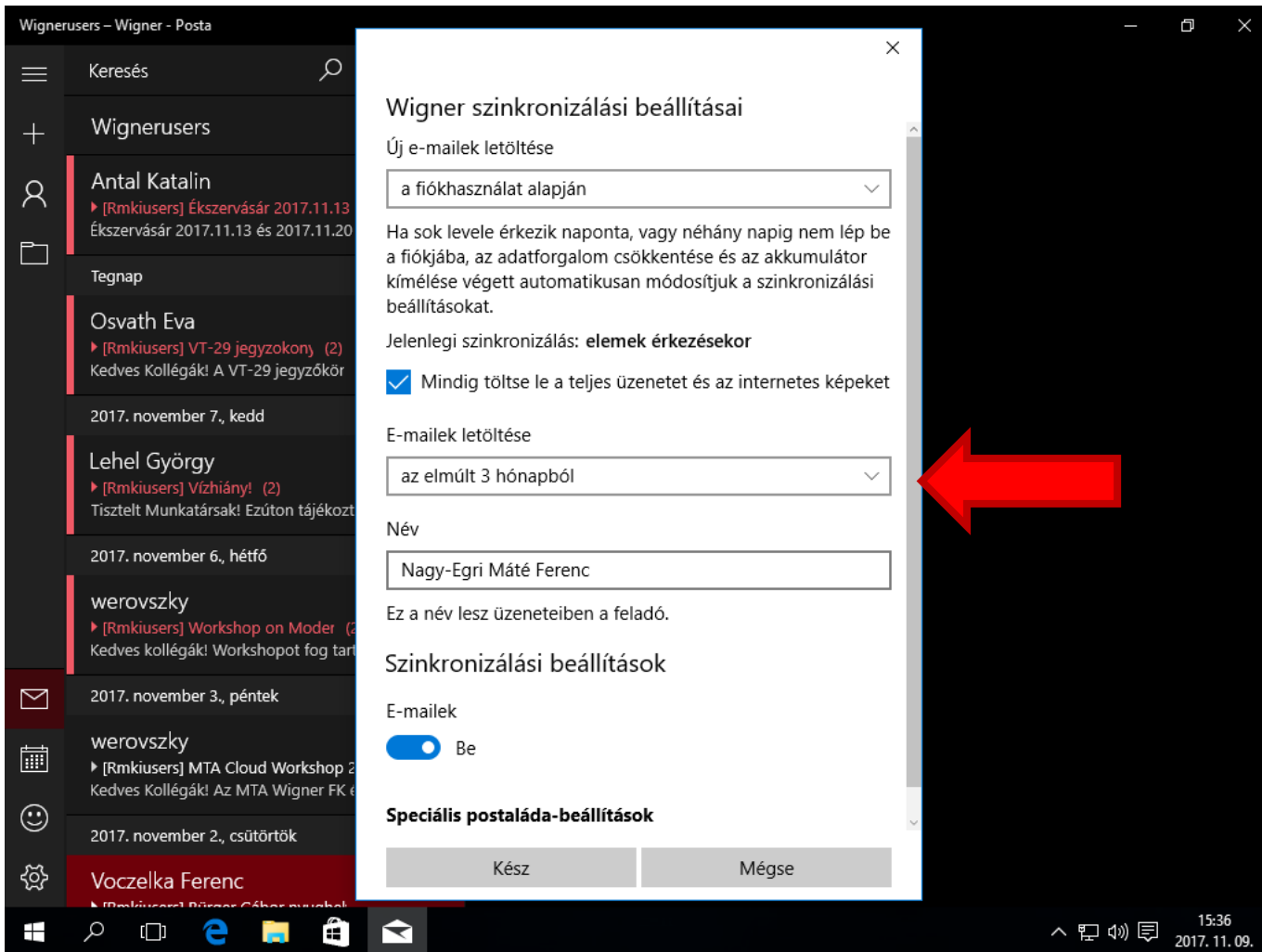
- Állítsuk be, hogy offline, csak az elmúlt 3 hónap leveleit lássuk
 - Emberke gomb
 - Jobb-klikk a fiókra
 - Fiókbeállítások

Posta app



- Postaláda szinkronizálási beállításainak módosítása

Posta app



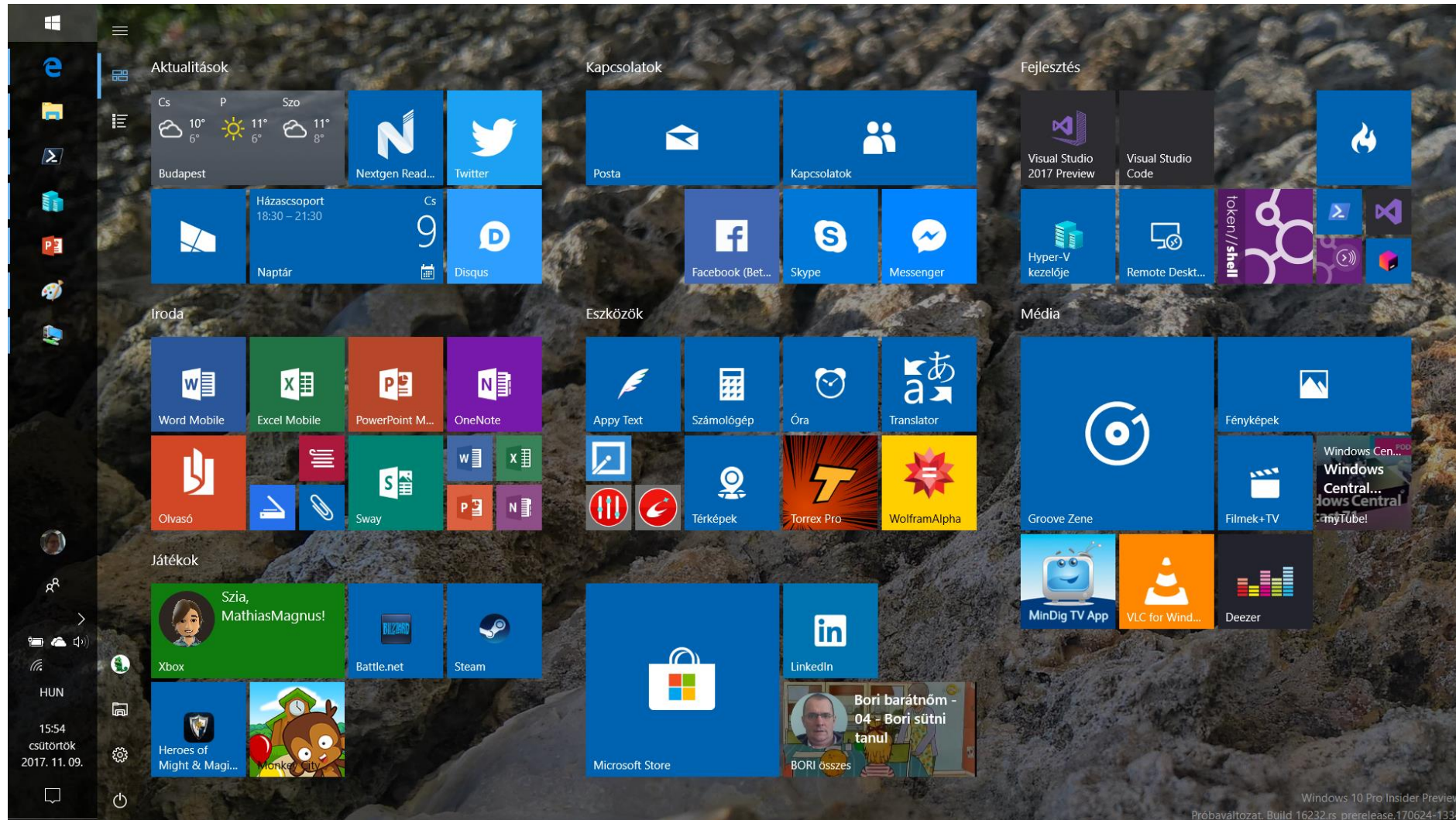
- Új e-mailek letöltése
 - „Elemek érkezésekor a leggyrosabb”
- E-mailek letöltése
 - Az elmúlt 3 hónap az alapértelmezett
 - Eztán csak a korlátlan van

Start menü

- Tételezzük fel, hogy megbékéltünk azzal a borzasztó különbséggel, hogy csempének hívják az ikonjainkat
- Mit lehet kihozni a Start menüből?



Start menü



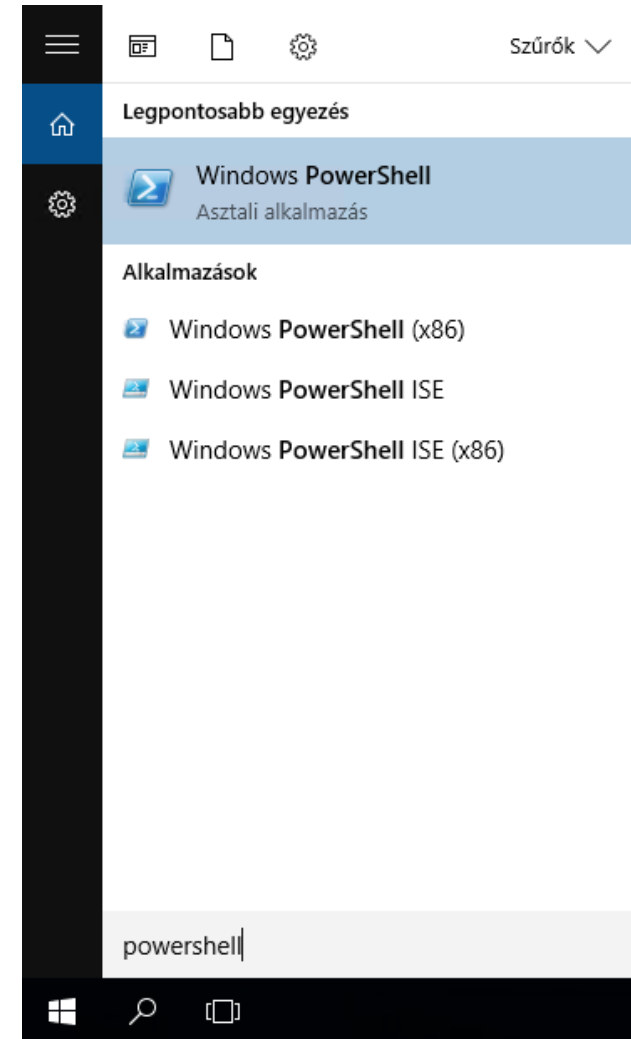
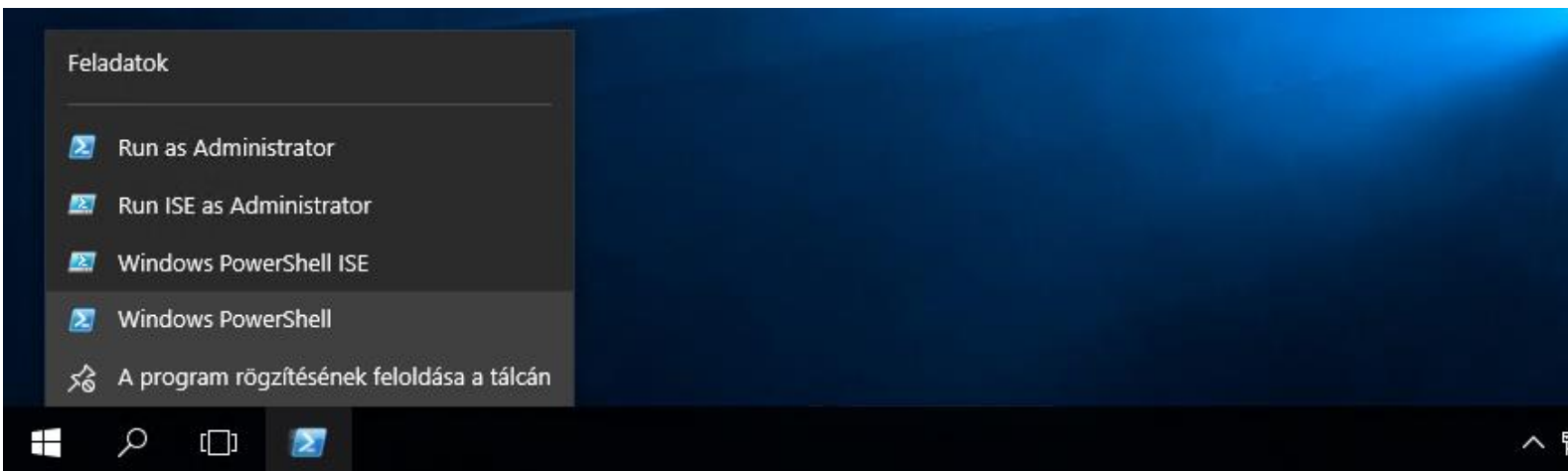


A parancssor meghódítása

Nyissunk egy kék ablakot a világra

A parancssor meghódítása

- Korábban már említettük, hogy a PowerShell hogyan indítható és hogyan készíthető benne telepítő pendrive
- De mi is az a PowerShell pontosan?



De mi is az a .NET pontosan?

- Egy szoftver futtatási keretrendszer (Common Language Runtime, aka. CLR) és egy hozzákapcsolt szabványos keretrendszer könyvtár, amelyek közösen nyelvi átjárhatóságot garantálnak
- Miért .NET?
 - A nyelvi átjárás észrevétlen
 - Könyvtárakat fejleszthetünk egy nyelvben és egy másiktól használhatjuk



De mi is az a .NET pontosan?

- Mely nyelvek tartoznak a .NET családba?
 - C#, a leggyakoribb .NET nyelv. MS alternatíva a Java-ra.
 - Java ~ C#, Java VM ~ .NET Runtime
 - F#, multi-paradigm (tisztán funkcionális, imperatív) nyelv
 - L#, egy Lisp implementáció .NET-re építve
 - C++/CLI, a C++ nyelvi kiterjesztés, amely hidat képez a natív és a kezelt (managed) .NET kód között
 - IronPython, egy Python implementáció .NET-re építve
 - PowerShell, egy interaktív shell és szkript nyelv
 - etc...



Windows PowerShell

- A .NET Full Framework-re épül
- A Windows szerves része
- Első verziója 2006
- A bash+Ruby/Perl windowsos megfelelője
- Elsődleges célja az automatizálás



Microsoft PowerShell



- A nyílt forrású .NET Core-ra épül
- Elérhető:
 - Windows, Linux, OSX
- Használható benne a teljes .NET Standard 2.0 könyvtár
- Szexi fekete ikonja van 😊

Something old, something new

```
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All rights reserved.
```

```
PS C:\Users\Matty> $PSVersionTable.PSVersion
```

Major	Minor	Build	Revision
5	1	16232	1000

```
Powershell v6.0.0-beta.9  
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
Loading personal and system profiles took 834ms.  
PS C:\Users\Matty> $PSVersionTable.PSVersion
```

Major	Minor	Patch	Label
6	0	0	beta.9



Mi a különbség?

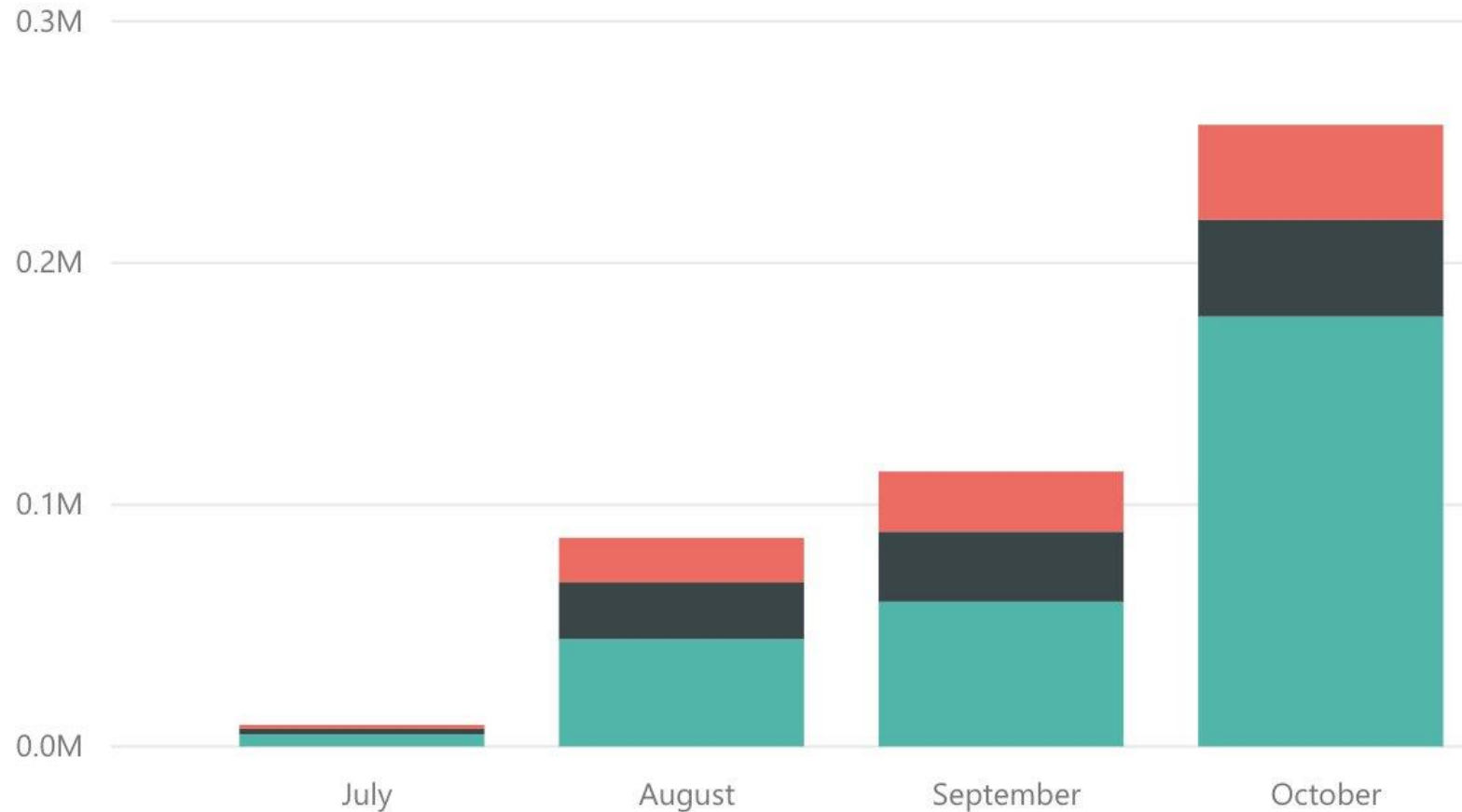
- A PowerShell 5 és leszármazottjai kompatibilitási okoknál fogva még maradnak egy ideig
- A legtöbb fejlesztést egyértelműen a PowerShell 6 kapja, ez lesz a PS jövője
- A PowerShell 6 mivel kereszt-platformos, ezért (sajnos) egy halom Windows-specifikus dolog hiányzik belőle
 - COM interop, hálózat kezelés, szolgáltatások kezelése, tűzfal kezelés...
- A PowerShell 6, mivel Linuxra és OSX-re is született, minimálisan másképp működik, hogy jobban kézre álljon a bash-en felnőtt embereknek is
 - & parancs vs. parancs &



PowerShell Core indítások száma

Monthly Usage by OS (90 Day)

userAgent ● Linux ● Mac ● Windows



Könyvtár struktúra

```
PS C:\Users\Matty> gci C:\
```



Könyvtár struktúra

```
PS C:\Users\Matty> gci C:\
```

Directory: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	2016. 05. 02. 11:43		Chocolatey
d-----	2017. 10. 27. 14:38		Jatekok
da----	2017. 11. 05. 17:36		Kellekek
d-----	2016. 05. 22. 23:54		Logs
d-----	2016. 07. 15. 15:13		Microsoft
d-----	2017. 06. 25. 9:08		PerfLogs
d-r---	2017. 11. 04. 13:04		Program Files
d-r---	2017. 11. 04. 13:06		Program Files (x86)
d-r---	2017. 10. 10. 18:48		Users
d-----	2017. 05. 06. 22:05		Virtual Machines
d-----	2017. 11. 10. 11:52		windows



Könyvtár struktúra

```
PS /home/mnagy> gci /
```



Könyvtár struktúra

```
PS /home/mnagy> gci /
```

```
Directory: /
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-r---	2016. 08. 03. 13:12		acct
d-r---	2017. 10. 24. 11:35		bin
d-r---	2016. 03. 23. 21:54		boot
d-----	1970. 01. 01. 1:00		cache
d-----	1970. 01. 01. 1:00		data
d-r---	2017. 11. 16. 14:45		dev
d-r---	2017. 10. 27. 12:10		etc
d-r---	2014. 04. 11. 0:12		home
d-r---	2017. 10. 24. 11:44		lib
d-r---	2017. 08. 21. 10:54		lib32
d-r---	2017. 06. 27. 13:13		lib64



Könyvtár struktúra

```
PS C:\Users\Matty> ls C:\
```



Könyvtár struktúra

```
PS C:\Users\Matty> ls C:\
```

Directory: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-----	2016. 05. 02. 11:43		Chocolatey
d-----	2017. 10. 27. 14:38		Jatekok
da----	2017. 11. 05. 17:36		Kellekek
d-----	2016. 05. 22. 23:54		Logs
d-----	2016. 07. 15. 15:13		Microsoft
d-----	2017. 06. 25. 9:08		PerfLogs
d-r---	2017. 11. 04. 13:04		Program Files
d-r---	2017. 11. 04. 13:06		Program Files (x86)
d-r---	2017. 10. 10. 18:48		Users
d-----	2017. 05. 06. 22:05		Virtual Machines
d-----	2017. 11. 10. 11:52		windows



Könyvtár struktúra

```
PS /home/mnagy> ls /
```



Könyvtár struktúra

```
PS /home/mnagy> ls /  
acct  boot  data  etc   init  lib32  lost+found  mnt  proc  run  srv  tmp  var  
bin   cache dev   home  lib   lib64  media       opt  root  sbin  sys  usr
```



```
PS /home/mnagy> ls /  
acct  boot  data  etc  init  lib32  lost+found  mnt  proc  run  srv  tmp  var  
bin   cache dev  home lib  lib64  media      opt  root  sbin sys  usr
```

- A PowerShell Core csapatának egy nehéz döntést kellett meghoznia:
 - Mi legyen azokkal a parancsokkal, amiket ~10 éve szándékosan azért neveztünk úgy el, mint a hasonló funkciójú *nix megfelelője, hogy kézre álljon?
 - Nem fedjük el őket
 - Nem jó, mert ha vegyes rendszereket kell kezelni, akkor barátságot kell kötni a nem ütköző nevekkel
 - Jó, mert a régi szkriptek (ls -al) továbbra is működni fognak
- Az `ls` és a `gci` hogy „ugyanazok”?



Parancsok felfedezése

```
PS C:\Users\Matty> Get-Command ls
```

CommandType	Name	Version	Source
-----	----	-----	-----
Alias	ls -> Get-ChildItem		

```
PS C:\Users\Matty> Get-Command ls,gci
```

CommandType	Name	Version	Source
-----	----	-----	-----
Alias	ls -> Get-ChildItem		
Alias	gci -> Get-ChildItem		

- Ja, hogy ezek aliasok?
- Ja, hogy a gci az Get-ChildItem?



Parancsok felfedezése

```
PS C:\Users\Matty> Get-Alias
```

CommandType	Name	Version	Source
-----	----	-----	-----
Alias	% -> ForEach-Object		
Alias	? -> Where-Object		
Alias	ac -> Add-Content		
Alias	cat -> Get-Content		
Alias	cd -> Set-Location		
Alias	chdir -> Set-Location		
Alias	clc -> Clear-Content		
Alias	clear -> Clear-Host		
Alias	clhy -> Clear-History		
Alias	cli -> Clear-Item		
Alias	clp -> Clear-ItemProperty		
Alias	cls -> Clear-Host		
Alias	clv -> Clear-Variable		
Alias	cmar -> Compress-Archive		
Alias	cnsn -> Connect-PSSession		



Parancsok felfedezése

```
PS C:\Users\Matty> Get-Command git.exe
```

CommandType	Name	Version	Source
Application	git.exe	2.14.1.1	C:\Kellekek\Git\2.8.1\cmd\git.exe

```
PS C:\Users\Matty> Get-Command cmake.exe
```

CommandType	Name	Version	Source
Application	cmake.exe	0.0.0.0	C:\Kellekek\Kitware\CMake\3.9.5\bin\cmake.exe

- Linuxről ismert `which` parancs megfelelője
 - Ha rendesen lett megírva, akkor a verziót is meg tudja mondani
 - Linuxon klasszikusan `util --version` és string parse



Parancsok felfedezése

```
PS C:\Users\Matty> Get-Help gci
```

NAME

Get-ChildItem

SYNOPSIS

Gets the items and child items in one or more specified locations.

SYNTAX

```
Get-ChildItem [[-Path] <String[]>] [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] [-UseTransaction <SwitchParameter>] [<CommonParameters>]
```

```
Get-ChildItem [[-Filter] <String>] [-Exclude <String[]>] [-Force] [-Include <String[]>] [-Name] [-Recurse] -LiteralPath <String[]> > [-UseTransaction <SwitchParameter>] [<CommonParameters>]
```



Parancsok felfedezése

DESCRIPTION

The `Get-ChildItem` cmdlet gets the items in one or more specified locations. If the item is a container, it gets the items inside the container, known as child items. You can use the `Recurse` parameter to get items in all child containers.

A location can be a file system location, such as a directory, or a location exposed by a different Windows PowerShell provider, such as a registry hive or a certificate store.

RELATED LINKS

Online Version: <http://go.microsoft.com/fwlink/?LinkId=821580>

`Get-Item`

`Get-Location`

`Get-Process`

`Get-PSPProvider`

REMARKS

To see the examples, type: `"get-help Get-ChildItem -examples"`.



Változók

```
PS C:\Users\Matty> $a = "valami"  
PS C:\Users\Matty>
```



Változók

```
PS C:\Users\Matty> $a = "valami"  
PS C:\Users\Matty> $a  
valami  
PS C:\Users\Matty>
```



Változók

```
PS C:\Users\Matty> $a = "valami"
```

```
PS C:\Users\Matty> $a
```

```
valami
```

```
PS C:\Users\Matty> $a.GetType()
```

```
IsPublic IsSerial Name
```

```
-----
```

```
True      True      String
```

```
BaseType
```

```
-----
```

```
System.Object
```

```
PS C:\Users\Matty>
```



Változók

```
PS C:\Users\Matty> $a = "valami"  
PS C:\Users\Matty> $a  
valami  
PS C:\Users\Matty> $a.GetType()
```

IsPublic	IsSerial	Name
True	True	String

BaseType
System.Object

```
PS C:\Users\Matty> $b = 5  
PS C:\Users\Matty> $b  
5  
PS C:\Users\Matty> $b.GetType()
```

IsPublic	IsSerial	Name
True	True	Int32

BaseType
System.ValueType



Változók

```
PS C:\Users\Matty> [Int32]$c = 5  
PS C:\Users\Matty> [Int32]$d = "mas"
```



Változók

```
PS C:\Users\Matty> [Int32]$c = 5
PS C:\Users\Matty> [Int32]$d = "mas"
Cannot convert value "mas" to type "System.Int32". Error: "Input string was not in a
correct format."
At line:1 char:1
+ [Int32]$d = "mas"
+ ~~~~~
+ CategoryInfo          : MetadataError: (:) [],
ArgumentTransformationMetadataException
+ FullyQualifiedErrorId : RuntimeException

PS C:\Users\Matty>
```



Változók

```
PS C:\Users\Matty> [Int32]$c = 5
PS C:\Users\Matty> [Int32]$d = "mas"
Cannot convert value "mas" to type "System.Int32". Error: "Input string was not in a
correct format."
At line:1 char:1
+ [Int32]$d = "mas"
+ ~~~~~
+ CategoryInfo          : MetadataError: (:) [],
ArgumentTransformationMetadataException
+ FullyQualifiedErrorId : RuntimeException

PS C:\Users\Matty> [Int32]$d = "5"
PS C:\Users\Matty> $d
5
PS C:\Users\Matty> $d.GetType()

IsPublic IsSerial Name                                     BaseType
-----
True     True     Int32                                     System.ValueType
```



Pár fontos változó

```
PS C:\Users\Matty> $PROFILE
C:\Users\Matty\Documents\PowerShell\Microsoft.PowerShell_profile.ps1
PS C:\Users\Matty> Test-Path $PROFILE
False
PS C:\Users\Matty> Test-Path $PSHOME\Microsoft.PowerShell_profile.ps1
False
```

	Bash	PowerShell
Rendszer	/etc/profile	\$PSHOME\Microsoft.PowerShell_profile.ps1
Felhasználó	~/.profile	\$PROFILE

- Két előre definiált változó segítségével lehet megtalálni a felhasználó saját, és a rendszer profil fájlokat
- Ezek szkriptek, amik mindig lefutnak induláskor



Pár fontos változó

```
PS C:\Users\Matty> [Int32]$c = 5
PS C:\Users\Matty> [Int32]$d = "mas"
Cannot convert value "mas" to type "System.Int32". Error: "Input string was not in a
correct format."
At line:1 char:1
+ [Int32]$d = "mas"
+ ~~~~~
+ CategoryInfo          : MetadataError: (:) [],
ArgumentTransformationMetadataException
+ FullyQualifiedErrorId : RuntimeException

PS C:\Users\Matty>
```



Pipeline

```
PS C:\WINDOWS\system32> Get-Disk
```

Number	Friendly Name	Serial Number	HealthStatus	OperationalStatus	Total Size	Partition Style
0	Samsung SS...	S1DHNSBF633364L	Healthy	Online	465.76 GB	GPT
2	ST1000LM 0...	9C000000000050506102	Healthy	Online	931.51 GB	GPT
1	SMI USB DISK	AA000000000000489	Healthy	Online	15.23 GB	MBR

- Emlékezzünk vissza, hogyan készítettünk telepítő pendrive-ot!



```
PS C:\WINDOWS\system32> Get-Disk -SerialNumber AA00000000000489 |  
>> Clear-Disk -RemoveData -Confirm:$false -PassThru |  
>> New-Partition -UseMaximumSize -IsActive:$true -AssignDriveLetter |  
>> Format-Volume -FileSystem NTFS -NewFileSystemLabel 'Windows 10 Installer'
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
E	Windows 10 Ins...	NTFS	Removable	OK	15.19 GB	15.23 GB

- Mi a ``-PassThru`` kapcsoló?
 - A `Get-`, `New-`, és hasonló parancsok (függvények) visszatérési értékei az objektumok, amiket létrehoztak
 - A `Set-`, `Clear-`, és hasonló „csak” módosító parancsoknak nincs visszatérési értékük, csak módosítják az objektum állapotát
 - Tipikusan pipeline-ok végén szoktak szerepelni
 - Ha mégis szeretnénk a módosított objektumot továbbadni, `-PassThru`



Pipeline

```
PS C:\WINDOWS\system32> Get-Disk -SerialNumber AA00000000000489 |  
>> Clear-Disk -RemoveData -Confirm:$false -PassThru |  
>> New-Partition -UseMaximumSize -IsActive:$true -AssignDriveLetter |  
>> Format-Volume -FileSystem NTFS -NewFileSystemLabel 'Windows 10 Installer'
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
E	Windows 10 Ins...	NTFS	Removable	OK	15.19 GB	15.23 GB

- Ezt az egész parancs láncot azért adtuk ki, hogy legyen egy megformázott, másolásra kész célterületünk
- Feltétlen konzolból kell másolni a meghajtó betűjelet?
- Zárójelekkel kiértékeltehetünk alkifejezéseket

```
PS C:\Users\Matty> (Get-Disk -Number 0 | Get-Partition -PartitionNumber 4).DriveLetter  
C  
PS C:\Users\Matty>
```

Pipeline

```
PS C:\WINDOWS\system32> Get-Disk -SerialNumber AA00000000000489 |  
>> Clear-Disk -RemoveData -Confirm:$false -PassThru |  
>> New-Partition -UseMaximumSize -IsActive:$true -AssignDriveLetter |  
>> Format-Volume -FileSystem NTFS -NewFileSystemLabel 'Windows 10 Installer'
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
E	Windows 10 Ins...	NTFS	Removable	OK	15.19 GB	15.23 GB

```
PS C:\WINDOWS\system32> Mount-DiskImage -ImagePath F:\win10.ISO -PassThru | Get-Volume
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
G	J_CES_X64...	Unknown	CD-ROM	OK	0 B	3.35 GB

```
PS C:\WINDOWS\system32> Copy-Item -Recurse G:\* E:\  
PS C:\WINDOWS\system32> Set-Location G:\boot\  
PS G:\boot> .\bootsect.exe /nt60 E:
```



Pipeline

```
PS C:\WINDOWS\system32> Get-Disk -SerialNumber AA00000000000489 |  
>> Clear-Disk -RemoveData -Confirm:$false -PassThru |  
>> New-Partition -UseMaximumSize -IsActive:$true -AssignDriveLetter |  
>> Format-Volume -FileSystem NTFS -NewFileSystemLabel 'Windows 10 Installer'
```

DriveLetter	FriendlyName	FileSystemType	DriveType	OperationalStatus	SizeRemaining	Size
E	Windows 10 Ins...	NTFS	Removable	OK	15.19 GB	15.23 GB

```
PS C:\WINDOWS\system32> Mount-DiskImage -ImagePath F:\win10.ISO -PassThru | Get-Volume
```

DriveLetter	FriendlyName	SystemType	DriveType	OperationalStatus	SizeRemaining	Size
G	J_CES_X64...	Unknown	CD-ROM	OK	0 B	3.35 GB

```
PS C:\WINDOWS\system32> Copy-Item -Recurse G:\* E:\  
PS C:\WINDOWS\system32> Set-Location G:\boot\  
PS G:\boot> .\bootsect.exe /nt60 E:
```



Miért beszéltem erről ennyit?



Python

- Történelmi okoknál fogva a tudományos világban vert gyökeret
- Később az adatbányászat is felkarolta
- C-ben íródott
 - C-ben írt könyvtárakkal kis fájdalommal interfészelhető
 - Bármilyen mással már macerásabb



PowerShell

- Történelmi okoknál fogva a rendszer üzemeltetői világban vert gyökeret
- Később az üzleti világ is felkarolta
- C#-ban íródott
 - .NET könyvtárakat használni gyerekjáték
 - C-ben írt könyvtárakat macerásabb



- A PowerShell egy .NET-re épülő interaktív shell és szkript nyelv
- Nem HPC-re való
 - Ahogy Python sem
- A 6-os verziója óta robbanásszerű fejlődésen megy keresztül
- Rendszergazdai feladatokra kiválóan alkalmas
- Expand-, Compress-Archive
 - unzip, zip
- Invoke-WebRequest
 - wget
- Select-String
 - grep
- Measure-Command
 - time
- Get-Process
 - ps





Telepítsünk programokat

Már amire nincsen app

Telepítők, csomag formátumok, stb.

- Windowson sajnos a programok telepítése egy hosszú és szomorú történet
 - Szoftver honlapjára felmegy
 - Telepítőt letölt (32/64-bit? , .msi/.exe?)
 - Telepít
 - Accept, Next, Next, Finish
 - Ismételve meg 30X
- A Microsoft egyik legnagyobb bűne, hogy ez a Windows 95 óta így megy
 - A Linux felhasználók meg röhögnek rajtunk, teljes joggal
- Az első érdemi változást a Win8 app formátuma, az [.appx](#) hozta
 - Röviden: ZIP+Web+XML= Open Packaging Conventions, ISO szabvány
 - Trusted kiterjesztése a [WSA csomagok](#)





Chocolatey is a package manager for Windows (like apt-get or yum but for Windows).

A csokról, röviden

- Csomagkezelő Windowsra
 - Megőrökli a formátumokat
- PowerShellre épül
- Közösségi összefogás
 - Bárki hozzájárulhat
- A leggyakoribb programok kényelmesen telepíthetők
- A csomagok lehetnek MSI, zip, NuGet, stb.
- <https://chocolatey.org/packages>



Hogyan tegyük fel?

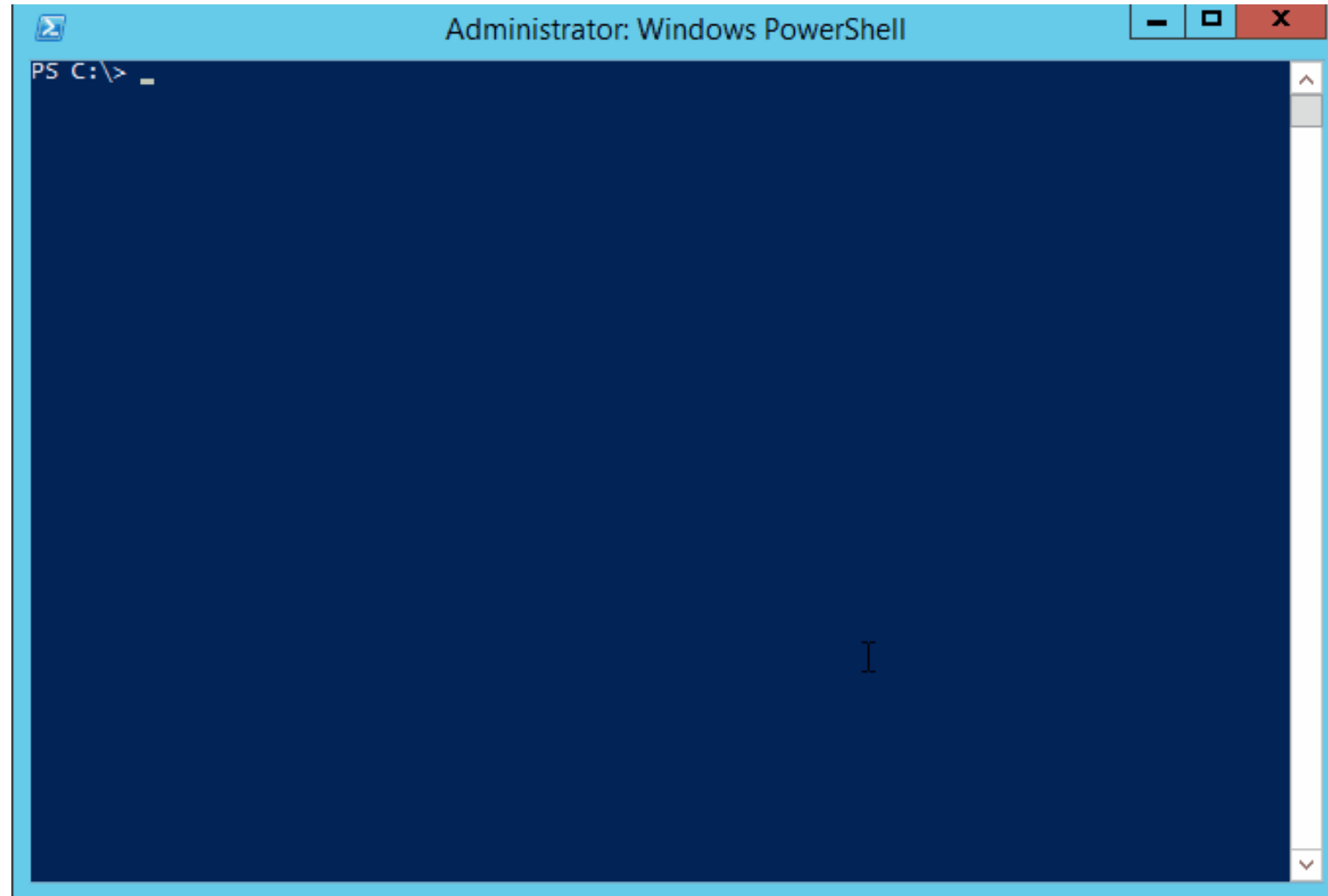
```
PS C:\Users\Matty> iwr https://chocolatey.org/install.ps1 | iex
```

- A telepítés nemigen lehetne ennél egyszerűbb
 - Egy rendszergazdai parancssorból adjuk ki a fenti egy soros parancsot
 - Mit csinál a parancs?
 - Az iwr egy alias az Invoke-WebRequest parancsra (wget is ugyanaz)
 - Az eredmény át van pipe-olva az Invoke-Expression parancsra
- Off-topic: Hogyan mentjük lemezre a telepítőt?
 - Ahelyett, hogy elindítanánk a scriptet, lemezre írjuk.
- Ha a szkriptek futtatása tiltott a gépen, ez a parancs telepít.

```
PS C:\Users\Matty> wget https://chocolatey.org/install.ps1 | Out-File install.ps1
```



- choco install
 - git.portable
 - poshgit
 - cmake.portable
 - ninja
 - visualstudiocode
 - gnuplot.portable
 - [openssh](#)
 - miktex.portable
 - vim
 - 7zip(.portable)





Kettő az egyben

Linux alrendszer (Windows Subsystem for Linux)

Windows Subsystem for Linux



Windows Subsystem for Linux

- Az NT kernel eredetileg egy mikrokernél, tehát több OS-t is támogathatna
 - A „véletlen” műve, hogy csak Win32 alkalmazásokat támogatott eddig
- Az utolsó userspace réteg a kernel előtt Linuxon a `syscall`-ok
 - Ezek lettek implementálva az NT kernelen belül
 - A linuxos Wine projekt fordította, csak kernel szinten implementálva
- Linuxos ELF binárisok futnak az NT kernelen!
 - A binárisok egyenesen az archive.ubuntu.com-ról jönnek
 - Nem csalás, nem ámtítás



Hogyan telepítsük?

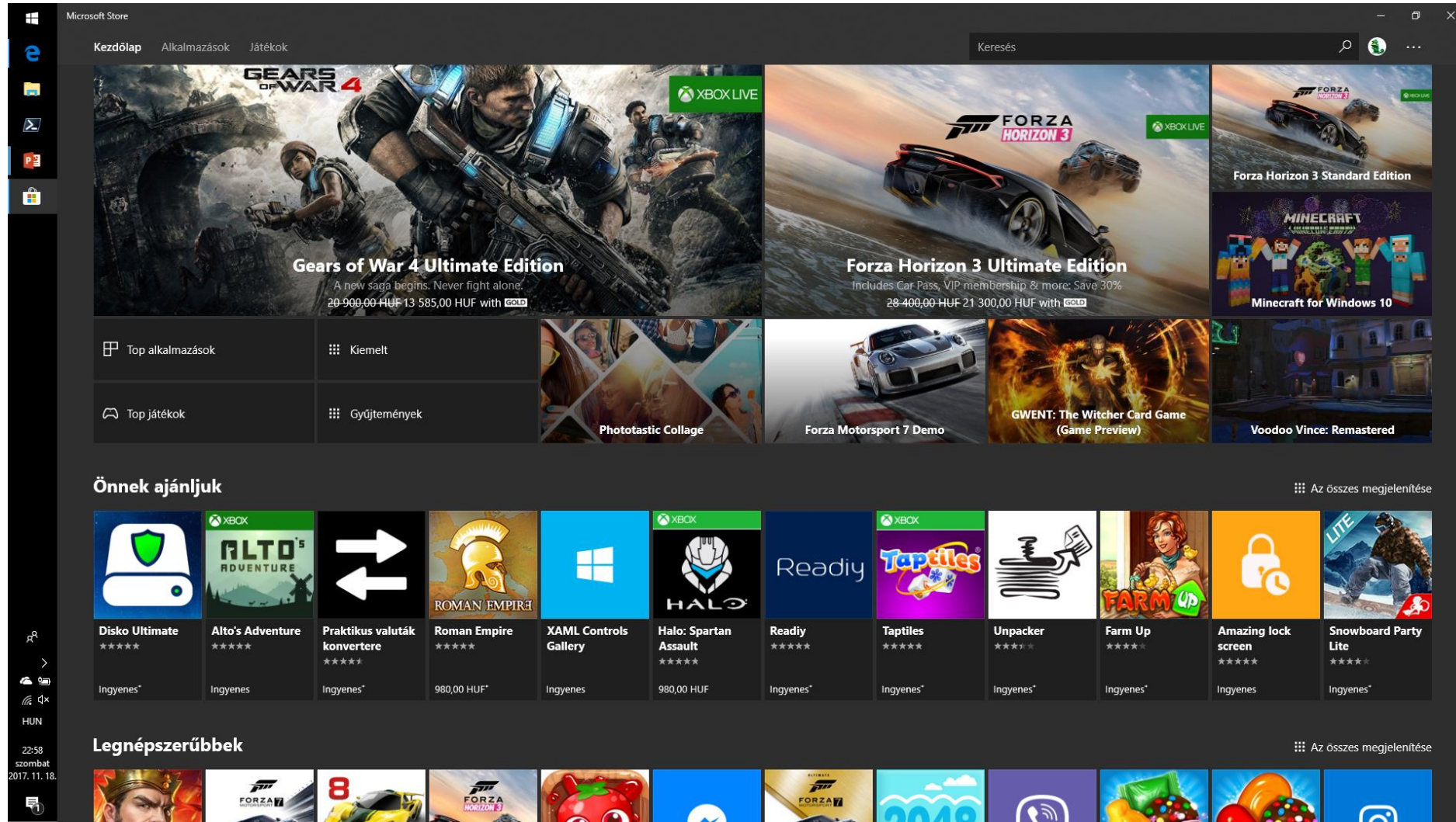
- Nyissunk egy rendszergazdai Windows PowerShellt

```
Windows PowerShell  
Copyright (C) 2016 Microsoft Corporation. All rights reserved.  
  
PS C:\WINDOWS\system32> Enable-WindowsOptionalFeature -FeatureName Microsoft-Windows-Subsystem-Linux
```

- Vezérlőpult >> Programok >> Programok és szolgáltatások
 - Windows szolgáltatások be- és kikapcsolása
 - Linuxos Windows alrendszer telepítése



Hogyan telepítsük?

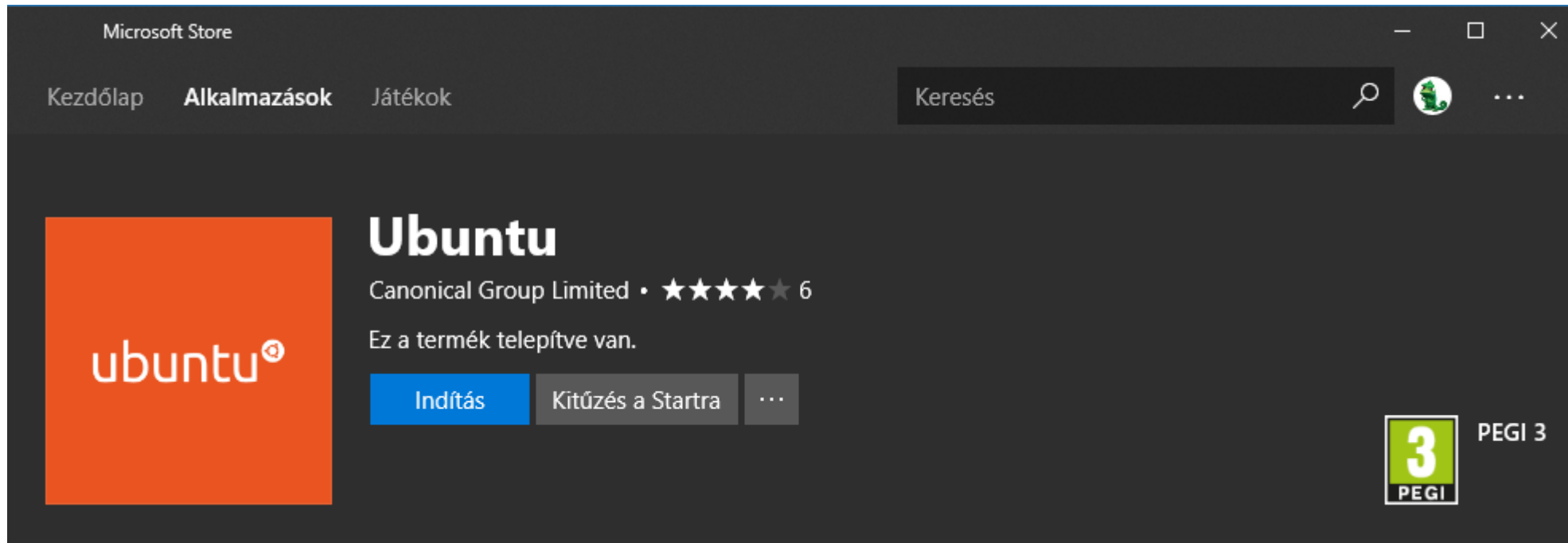


Hogyan telepítsük?

- Microsoft Store
- Kereső
 - Ubuntu
 - Suse
 - Mint



Hogyan telepítsük?



- Indítani
 - Start, Bash on Ubuntu on Windows
 - cmd.exe, bash.exe
- Első indításkor kér majd
 - unix felhasználó nevet
 - ez lesz társítva az aktuálisan bejelentkezett Windows felhasználóhoz
 - unix jelszót



Gyakorlatban

```
PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 2003ms.
PS C:\Users\Matty> bash.exe
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mnagy@MATTY-Z50-75:/mnt/c/Users/Matty$ cd
mnagy@MATTY-Z50-75:~$ cat /etc/issue
Ubuntu 16.04.1 LTS \n \l

mnagy@MATTY-Z50-75:~$ which cat
/bin/cat
mnagy@MATTY-Z50-75:~$ ldd /bin/cat
        linux-vdso.so.1 => (0x00007ffffde3a6000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f06be7f0000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f06bec00000)
```



Gyakorlatban

```
PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

Loading personal and system profiles took 2003ms.
PS C:\Users\Matty> bash.exe
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

mnagy@MATTY-Z50-75:/mnt/c/Users/Matty$ cd
mnagy@MATTY-Z50-75:~$ cat /etc/issue
Ubuntu 16.04.1 LTS \n \l

mnagy@MATTY-Z50-75:~$ which cat
/bin/cat
mnagy@MATTY-Z50-75:~$ ldd /bin/cat
linux-vdso.so.1 => (0x00007ffffde3a6000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f06be7f0000)
/lib64/ld-linux-x86-64.so.2 (0x00007f06bec00000)
```



Minimalista

Unix top

```
mnagy@MATTY-GL702ZC: /mnt/c/Users/nagy-  
top - 23:16:55 up 1 min, 0 users, load average: 0.52, 0.58, 0.59  
Tasks: 3 total, 1 running, 2 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 0.1 us, 0.2 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st  
KiB Mem : 16707892 total, 12253712 free, 4217704 used, 236476 buff/cache  
KiB Swap: 31119484 total, 31111780 free, 7704 used. 12349332 avail Mem  
  
  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND  
    1 root        20   0   10440   584   552 S   0.0   0.0   0:00.10 init  
    2 mnagy      20   0   25820   3600 3504 S   0.0   0.0   0:00.07 bash  
   19 mnagy      20   0   41720   1948 1420 R   0.0   0.0   0:00.10 top
```

Windows Feladatkezelő

Feladatkezelő						
Fájl Beállítások Nézet						
Folyamatok Teljesítmény Alkalmazáselelmények Indítás Felhasználók Részletek Szolgáltatások						
Név	0% Processzor	26% Memória	0% Lemez	0% Hálózat	0% GPU	GPU-motor
Windows Sneh Experience Host ...	0%	32,4 MB	0 MB/s	0 MB/s	0%	
Keresés (2)	0%	41,3 MB	0 MB/s	0 Mb/s	0%	
Posta (2)	0%	1,7 MB	0 MB/s	0 Mb/s	0%	
Microsoft Outlook Communica...	0%	14,6 MB	0 MB/s	0 Mb/s	0%	
Gépház	0%	0,1 MB	0 MB/s	0 Mb/s	0%	
Microsoft Store (2)	0%	3,3 MB	0 MB/s	0 Mb/s	0%	
Facebook (2)	0%	4,7 MB	0 MB/s	0 Mb/s	0%	
Twitter (2)	0%	0,9 MB	0 MB/s	0 Mb/s	0%	
pwsh.exe (3)	0,1%	35,4 MB	0 MB/s	0 Mb/s	0%	
Rendszermegszakítások	0%	0 MB	0 MB/s	0 Mb/s	0%	
Szolgáltatásgazda: Csempék ad...	0%	1,4 MB	0 MB/s	0 Mb/s	0%	
Top	0%	0,5 MB	0 MB/s	0 Mb/s	0%	
Bash	0%	0,1 MB	0 MB/s	0 Mb/s	0%	
WMI Provider Host	0%	1,7 MB	0 MB/s	0 Mb/s	0%	
Init	0%	0,1 MB	0 MB/s	0 Mb/s	0%	
Szolgáltatásgazda: Helyi rendszer	0%	0,9 MB	0 MB/s	0 Mb/s	0%	
Windows Defender SmartScreen	0%	4,3 MB	0 MB/s	0 Mb/s	0%	
Microsoft Windows Search Filte...	0%	3,3 MB	0 MB/s	0 Mb/s	0%	



Az apróbetűs rész

- Mi az, amit NEM tud?
 - Semmi nem működik, amihez kernel modul kell
 - Hiszen a kernel nincs ott, nincs mibe beépülni
 - Eszközöket nagyon limitáltan lehet csak elérni
 - A hálózat természetesen működik és a Windowstól kölcsönzi
 - USB Windows 10 1709 óta működik (IoT fejlesztők kérték)
 - GPU és GPGPU nincs az alrendszerben (eszköz elérés)
- Amire figyelni kell!
 - Fájrendszer között korlátozott átjárhatóság
 - Az alrendszer /mnt/c és társai utakon eléri a Windowsos partíciókat írni/olvasni
 - Windowson az egész alrendszer egy mappába van betéve, DE NEM ÍRHATÓ!
 - A javasolt munkafolyamat: minden dokumentumunk mezei NTFS doksik legyenek.
 - A fájlrendszer teljesítménye messze alul múlja a natívét





Félelem az ismeretlentől

„A félelem haragot szül, a harag gyűlöletet, a gyűlölet kint és szenvvedééést...” - Yoda mester

Félelem az ismeretlentől

- Mi is az az IDE?
 - Integrated Development Environment, avagy integrált fejlesztői környezet
 - Olyan eszközök gyűjteménye, melyek összhangban vannak egymással, megkönnyítve a szoftverfejlesztés különböző lépéseit úgy, hogy maximalizálják a munka hatékonyságát
- Milyen különböző lépések ezek?
 - Forrás fájl szerkesztés, fordítás automatizálása, hibakeresés, verzió követés, tesztelés, profilozás, telepítő készítés és publikálás, stb.
- Bakker, de én egy fizikus vagyok, miért foglalkozzam ezzel?
 - Pont azért!



Előnyök, hátrányok

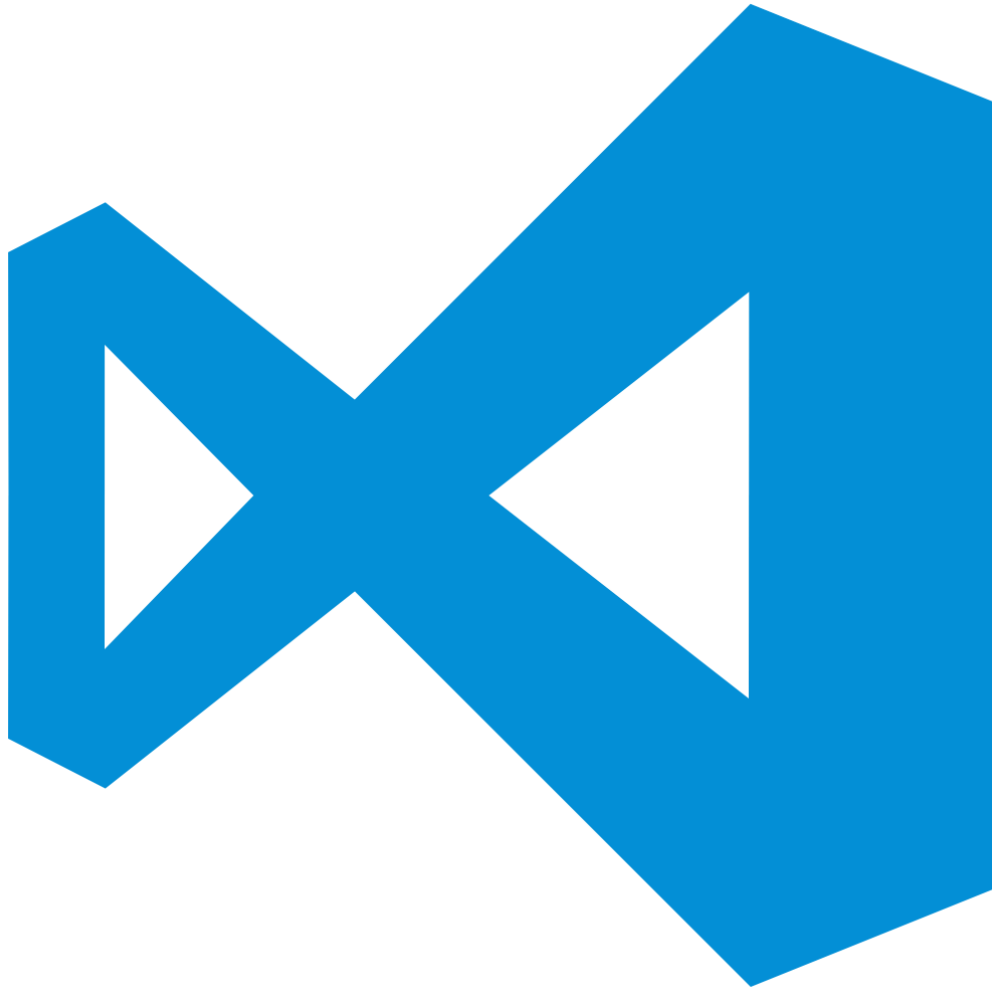
Integrált fejlesztői környezet

- Pro
 - Elejétől a végéig automatizált
 - Természetes a munkafolyamat
 - Könnyű megtanulni, „nehéz” tökélyre vinni
- Con
 - Avval főzünk, amink van
 - Az IDE megválasztása fontossá válik
 - A kiterjeszthetőség fontos

Eszközlánc

- Con
 - Külön eszköz mindenre
 - Interfészelni kell őket
 - Nem mindig zökkenőmentes
 - Nehéz megtanulni, nehéz tökélyre vinni
 - Kialakulnak munkafolyamatok (szkriptelés?)
- Pro
 - Mindenből csak a legjobbat!

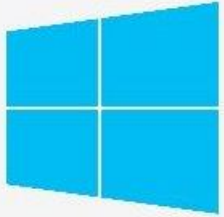




- A Visual Studio Code fiatal
 - Első kiadása: 2015 április 29.
 - Nyílt forrás: 2015 november 18.
- Kereszt-platform támogatás
 - Windows, Linux, OSX
 - JavaScript és TypeScript alapok
- Kezdetben szövegszerkesztő
 - Közösségi nyomásra megnyitották a beépülő modulok API-ját
 - Azóta elképesztő fluxussal érkeznek a beépülő modulok



<https://code.visualstudio.com>



↓ Windows

Windows 7, 8, 10

.zip



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, CentOS

.tar.gz | 32 bit versions



↓ Mac

macOS 10.X

```
mnagy@MATTY-Z50-75:~$ sudo dpkg -i code_1.6.1-1476373175_amd64.deb
```

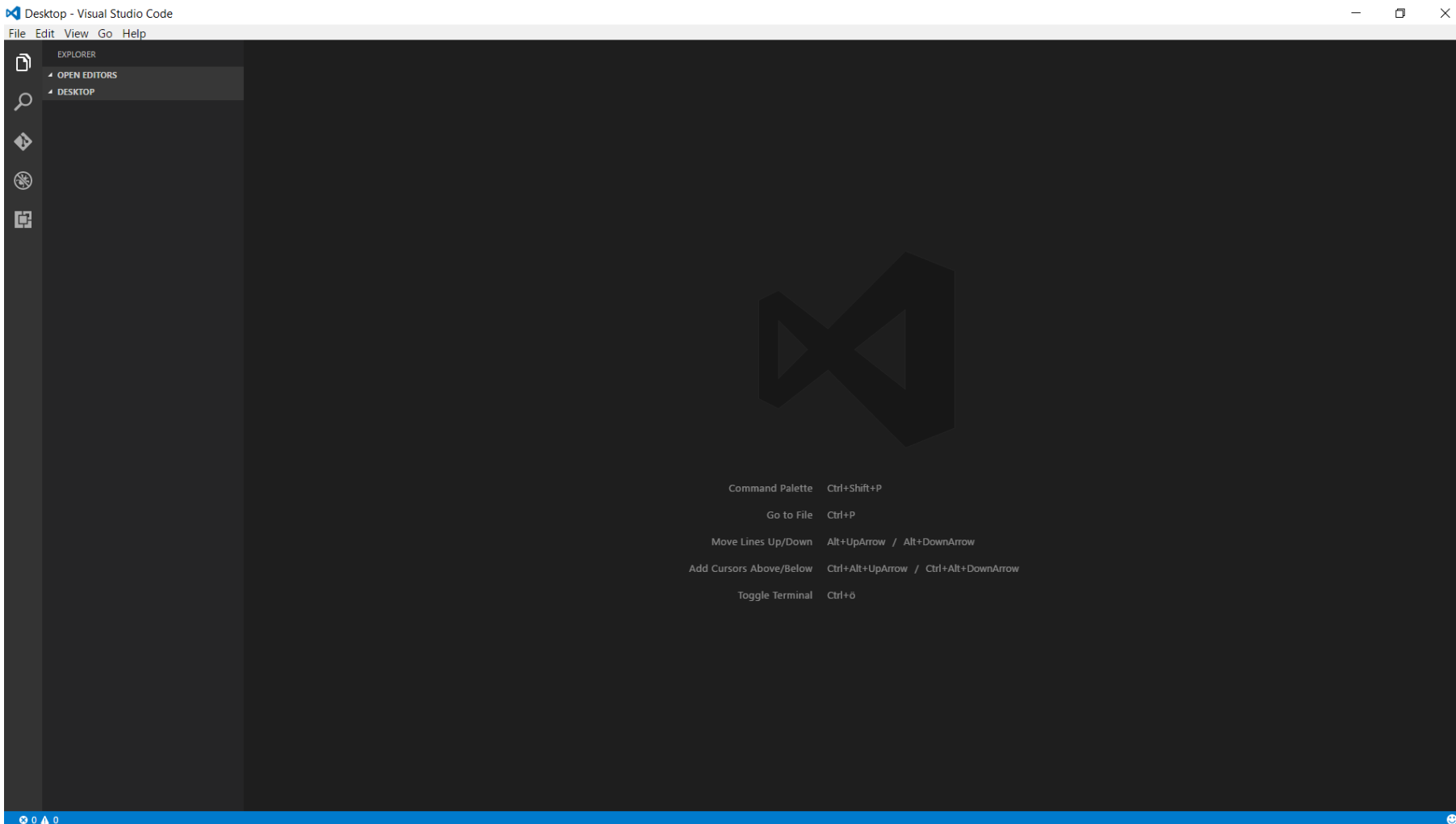
<https://code.visualstudio.com/docs/setup/linux>

```
PS /home/mnagy> sudo apt install code
```

```
PS C:\Users\mnagy> choco install visualstudiocode
```



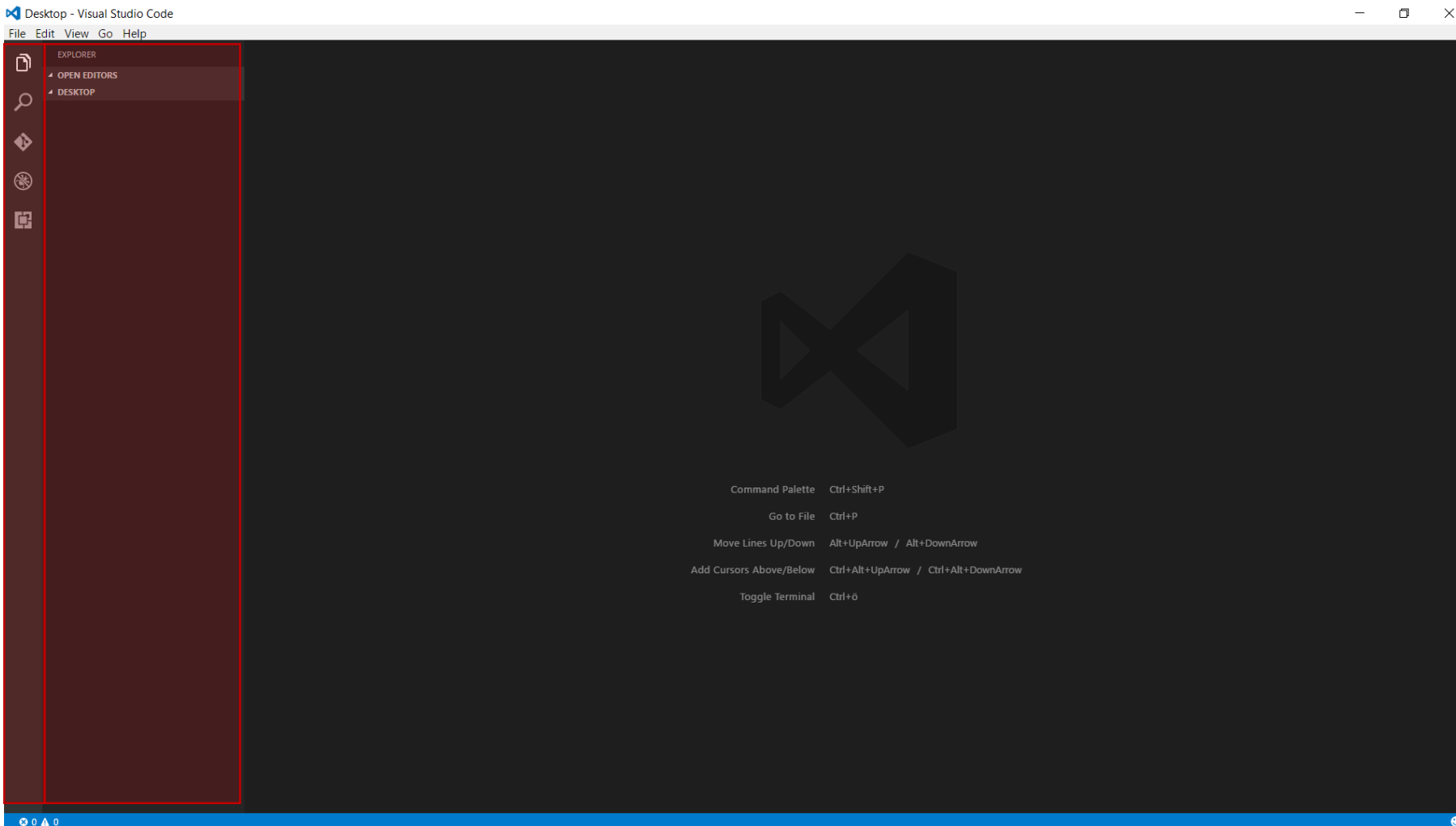
Első pillantásra



Első pillantásra



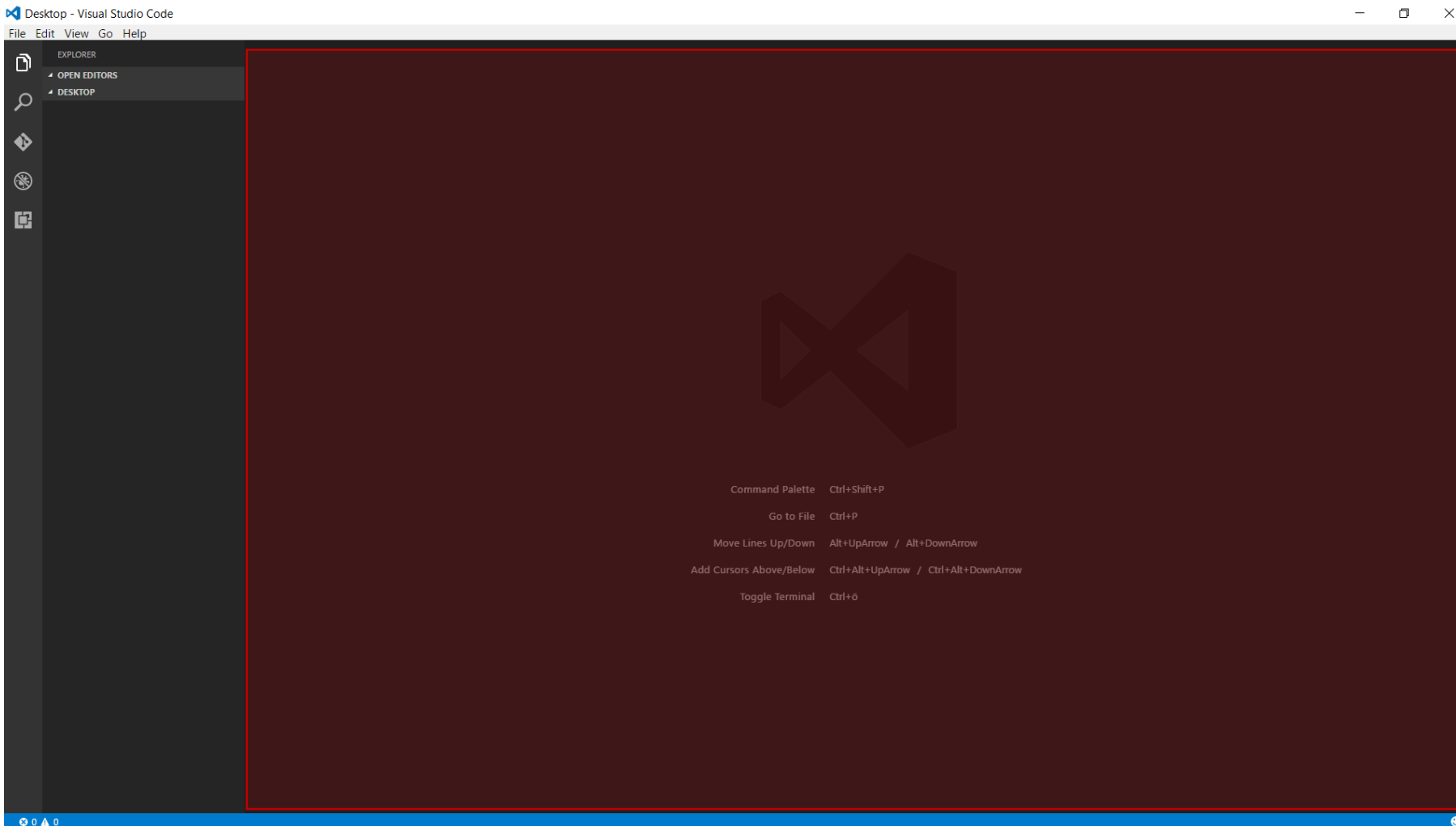
- Oldalsáv
- Beépített funkciók gombjai találhatóak itt
 - Fájl nézet
 - Keresés
 - Git
 - Debug
 - Bővítmények



Első pillantásra

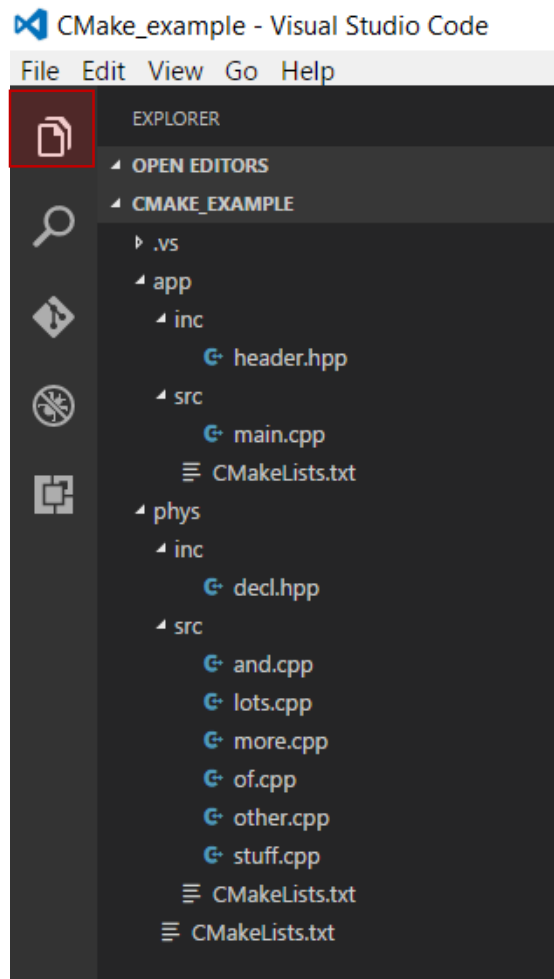


- Szerkesztő panel
- Itt töltjük a legtöbb fejlesztési időt
- Felosztható
 - Fejléc
 - Forrás
- Diff nézet

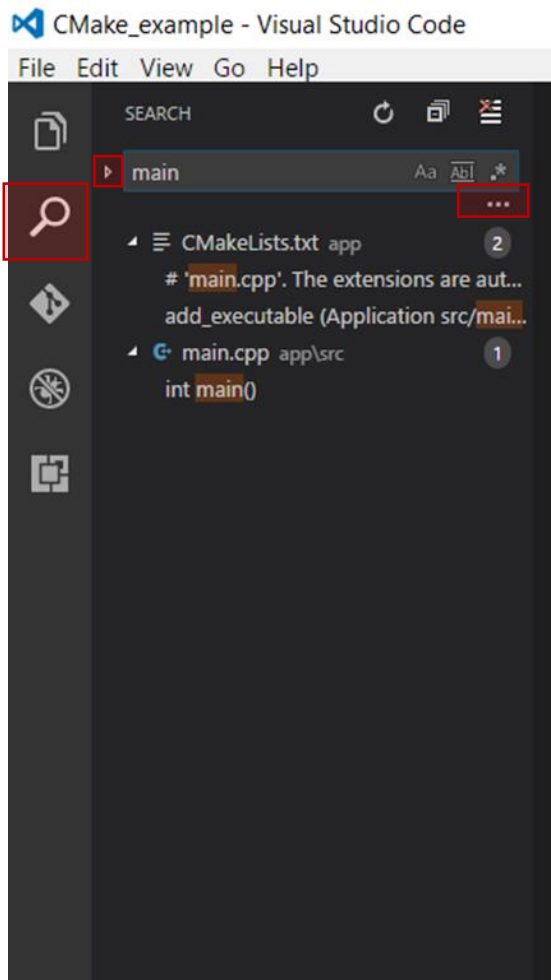


- Használható egyszerű szövegszerkesztőként is
 - Fájlt megnyit, szerkeszt, ment, kilép
- Használható, mint egy nagy mértékben testreszabható, teljes értékű fejlesztői környezet
 - Egy teljes mappa is megnyitható vele
 - A globális beállítások platform-specifikus helyen tároltak
 - A munkaterület-specifikus beállítások (a mappák, amiket megnyitottunk) helyben tároltak egy `.vscode` nevű mappában
 - Akinek ez nem felel meg, az kövesse [ezt](#) a Github bejegyzést

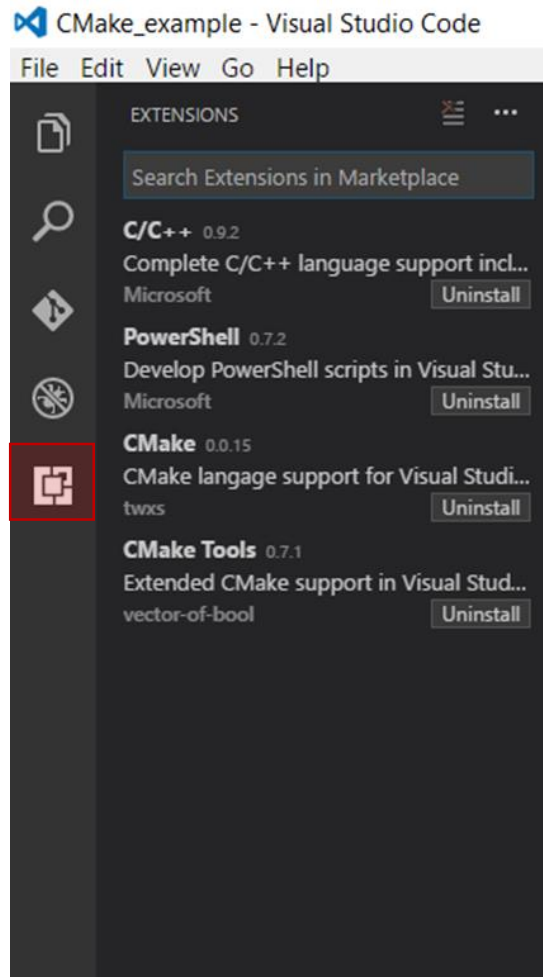




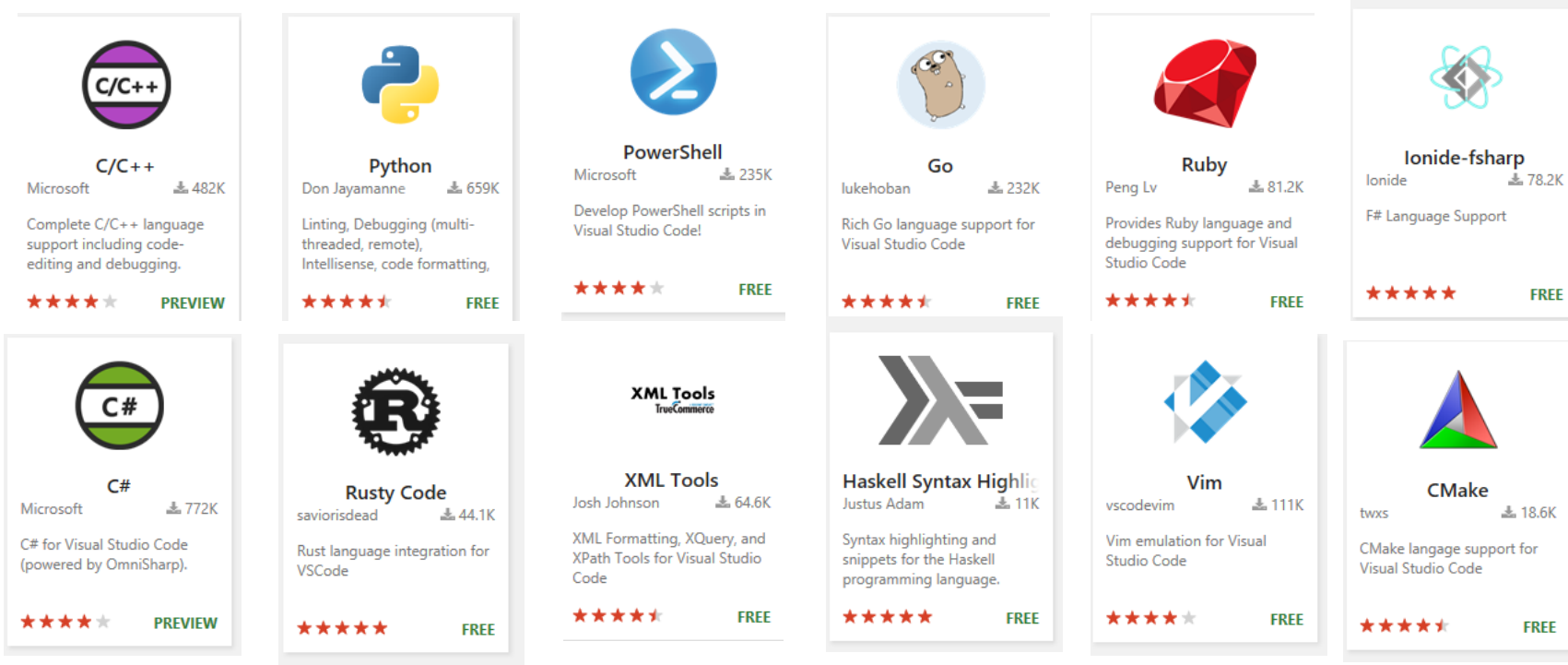
- A fájl nézet gomb egy fa struktúrában megjeleníti a megnyitott mappa tartalmát
- A gyors azonosíthatóság jegyében ikon csomagok is letölthetők
 - Fájl, Beállítások, Fájl ikon témák, Seti (pl.)
 - CMake ikonokért [ez](#) a Github bejegyzés követhető
- Mappák összecsuksukhatók/lenyithatók igény szerint
- Dupla-klikkelve bármelyik fájl megnyitható a szerkesztőben

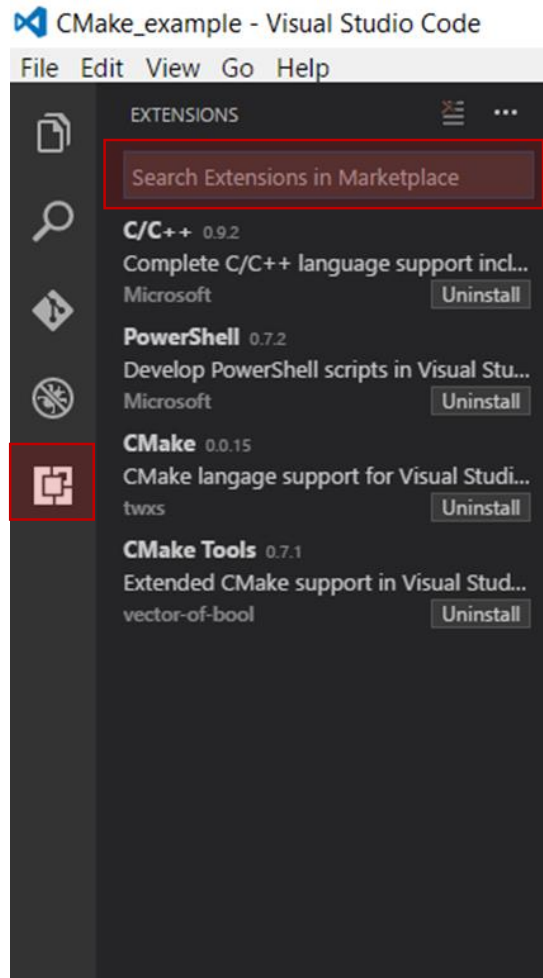


- A kereső gombbal kereshetünk a fájlok tartalmában (grep) akár reguláris kifejezések használatával is
- Írjunk be a kereső kifejezést tartalmat, hogy kapjunk egy listát minden fájlal, amely eleget tesz neki
- A kicserélés funkció a lenyitható háromszöggel hívható elő
- Az ... ikon a szövegdoboz alatt további keresési opciókat hoz elő



- A bővítmények gombbal:
 - kereshetünk a piactérről új bővítményeket
 - kezelhetjük a már telepített bővítményeket





- To install an extension:
 - Click the search bar
 - Type the name of the extension
 - Click on the **Install** button
 - When it's done, click on the **Reload** button
- This tutorial will show the synergy of these:
 - C/C++
 - CMake
 - CMake Tools



- Linuxon

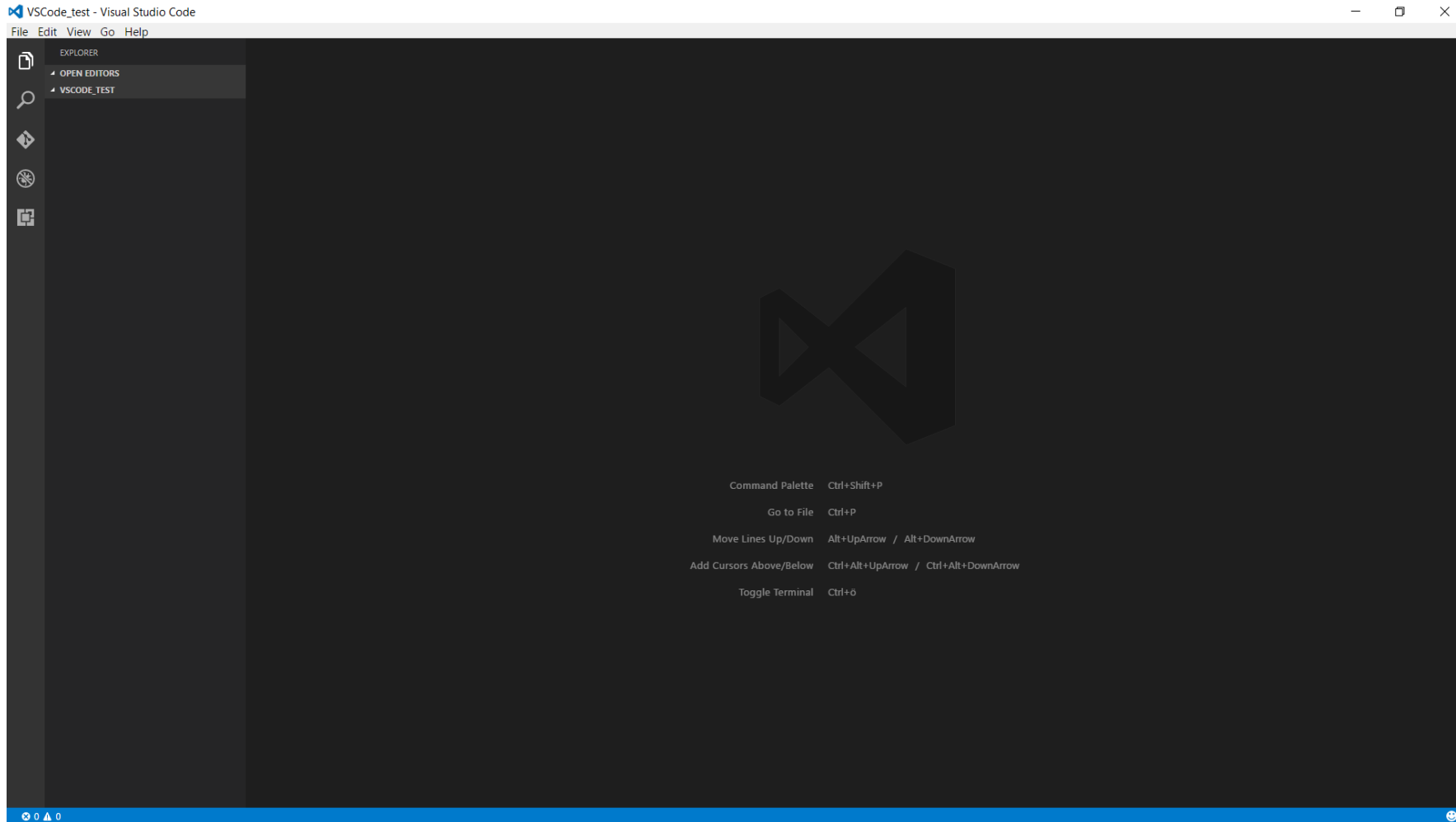
- Feltételezem, hogy egy működő C/C++ fordító és linker már telepítve van a gépre és elérhető a \$PATH-ból
 - gcc/g++
 - clang/clang++ illetve lld

- Windowson

- Feltételezem, hogy egy működő C/C++ fordító és linker már telepítve van a gépre
 - [Standalone Visual C++ Build Tools](#) (msvc)
 - [Visual Studio 2017 Community Edition](#) (a nagytestvér Visual Studio + msvc)
 - clang-cl illetve lld (letölthető [innen](#))



Lássunk hozzá...



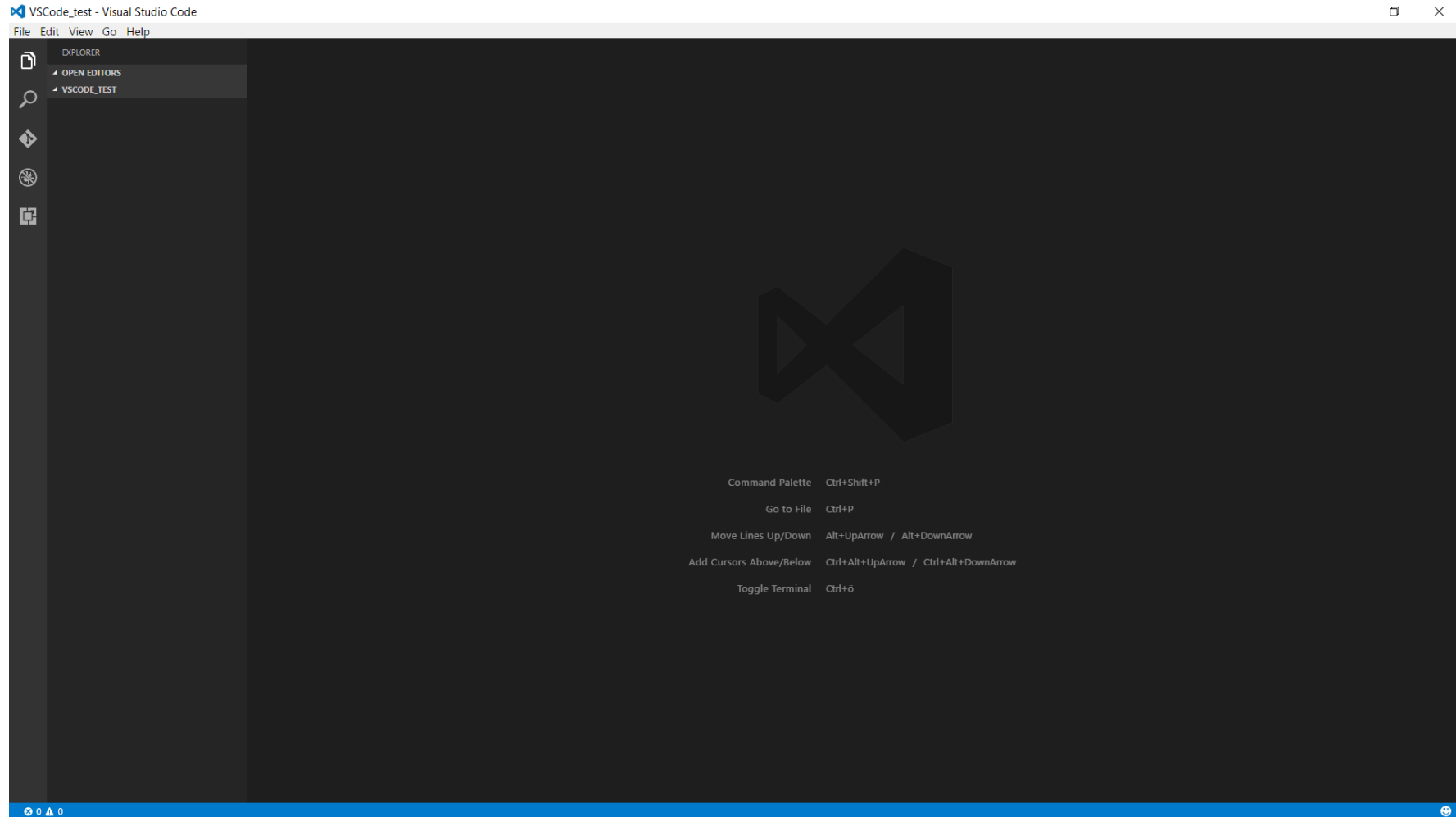
Lássunk hozzá...

- Csináljunk egy üres mappát, amiben majd fejleszteni kezdünk
 - Windows-on a GUI shell kontextus menüjébe beépül VS Code, és tetszőleges mappa ikonjára vagy egy mappában üres területre jobbklikkelve megjelenik egy „Open with Code” kontextus menü
 - Linuxon parancssorból a ``code /path/to/folder`` szintaxissal nyitható meg egy mappa
 - Programon belülről a nyitóképernyőről, illetve a Fájl menüből is nyitható mappa
- Kezdetben egy üres IDE fog bennünket fogadni, az egyes bővítmények gombjai akkor jelennek csak meg, ha megfelelő fájlok vannak a megnyitott mappában, vagy ha végrehajtjuk egy parancsát
- Először jóváhagyjuk a CMake Tools automatikus konfigurálását.



Lássunk hozzá...

- A parancs palettát megnyitni
 - Ctrl+Shift+P
 - View menü majd Show Command Palette
- Ez megjeleníti majd az összes elérhető parancsot valamennyi bővítményből
- Írjuk be; „CMake”
- Utána válasszuk a „Edit Kits file” gombot



Lássunk hozzá...

Info CMake Tools has scanned for available kits and saved them to a file. Would you like to edit the Kits file?

Yes No

- Amikor a CMake kiterjesztés először aktiválódik, neki áll „Kit”-eket keresni
- Egy [Kit](#) az egy konform, CMake által ismert fordító N-esből és a hozzájuk kapcsolódó linkerekből, egyéb eszközökből áll.
- A bővítmény megkérdezi, hogy szeretnénk-e megnézni azt a JSON fájlt, amibe összegyűjtötte az általa talált Kiteket. Mondhatunk igen és nemet is
 - Később a parancs palettáról ugyanúgy nyithatjuk meg ezt a fájlt a „CMake: Edit CMake Kits” parancs kiválasztásával
- Ha jóváhagytuk az eredményt a „CMake: Quick Start” parancsot választjuk és konfiguráljuk az első programunk.



Egy kicsi hasonlítás

Eredeti (csak MSVC detektálva)

```
[
  {
    "name": "Visual Studio Build Tools 2017 - amd64",
    "visualStudio": "66d7a7eb",
    "visualStudioArchitecture": "amd64",
    "preferredGenerator": {
      "name": "Visual Studio 15 2017",
      "platform": "x64"
    }
  },
  {
    "name": "Visual Studio Build Tools 2017 - x86",
    "visualStudio": "66d7a7eb",
    "visualStudioArchitecture": "x86",
    "preferredGenerator": {
      "name": "Visual Studio 15 2017"
    }
  }
]
```

Módosít (32-bitre nem fordítunk)

```
[
  {
    "name": „MSVC 15.5.2”,
    "visualStudio": "66d7a7eb",
    "visualStudioArchitecture": "amd64",
    "preferredGenerator": {
      "name": "Visual Studio 15 2017",
      "platform": "x64"
    }
  }
]
```



Lássunk hozzá...

- Eztán a kiterjesztés három keresztkérdést fog feltenni, tudjuk-e mit is szeretnénk valójában
 - Mi legyen a neve a kimeneti binárisnak?
 - Könyvtár vagy futtatható program-e a kimenet?
 - Debug, Release vagy valami cselesebb fordítási módot szeretnénk-e alkalmazni?

VSCode_test

Enter a name for the new project (Press 'Enter' to confirm or 'Escape' to cancel)

Library Create a library

Executable Create an executable

Debug Emit debug information without performing optimizations

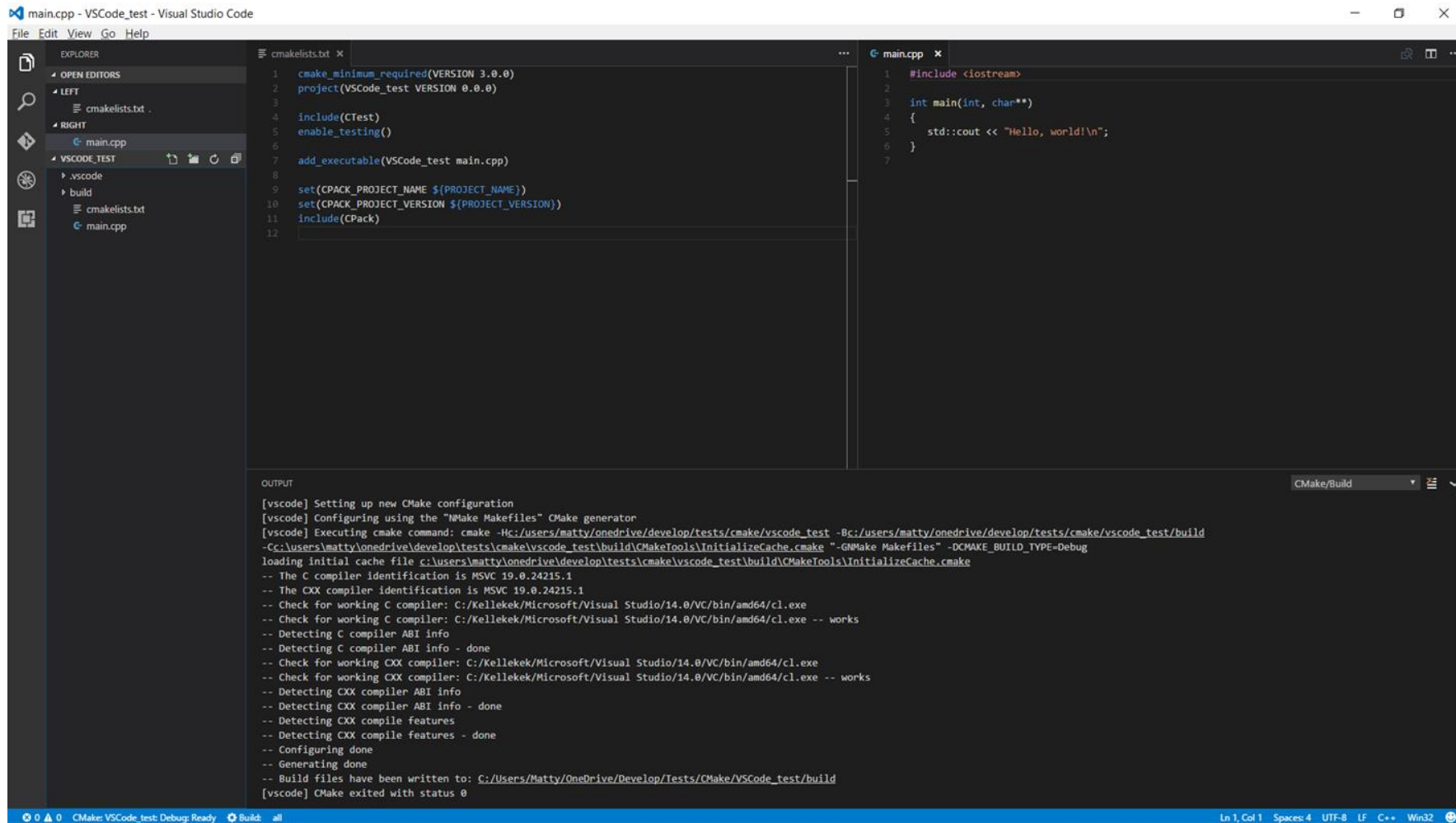
Release Enable optimizations, omit debug info

MinSizeRel Optimize for smallest binary size

RelWithDebInfo Perform optimizations AND include debugging information



Repülő rajt



```
main.cpp - VSCode_test - Visual Studio Code
File Edit View Go Help

EXPLORER
  OPEN EDITORS
  LEFT
    cmakelists.txt
  RIGHT
    main.cpp
  VS_CODE_TEST
    .vscode
    build
    cmakelists.txt
    main.cpp

cmakelists.txt
1 cmake_minimum_required(VERSION 3.0.0)
2 project(VSCode_test VERSION 0.0.0)
3
4 include(CTest)
5 enable_testing()
6
7 add_executable(VSCode_test main.cpp)
8
9 set(CPACK_PROJECT_NAME ${PROJECT_NAME})
10 set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
11 include(CPack)
12

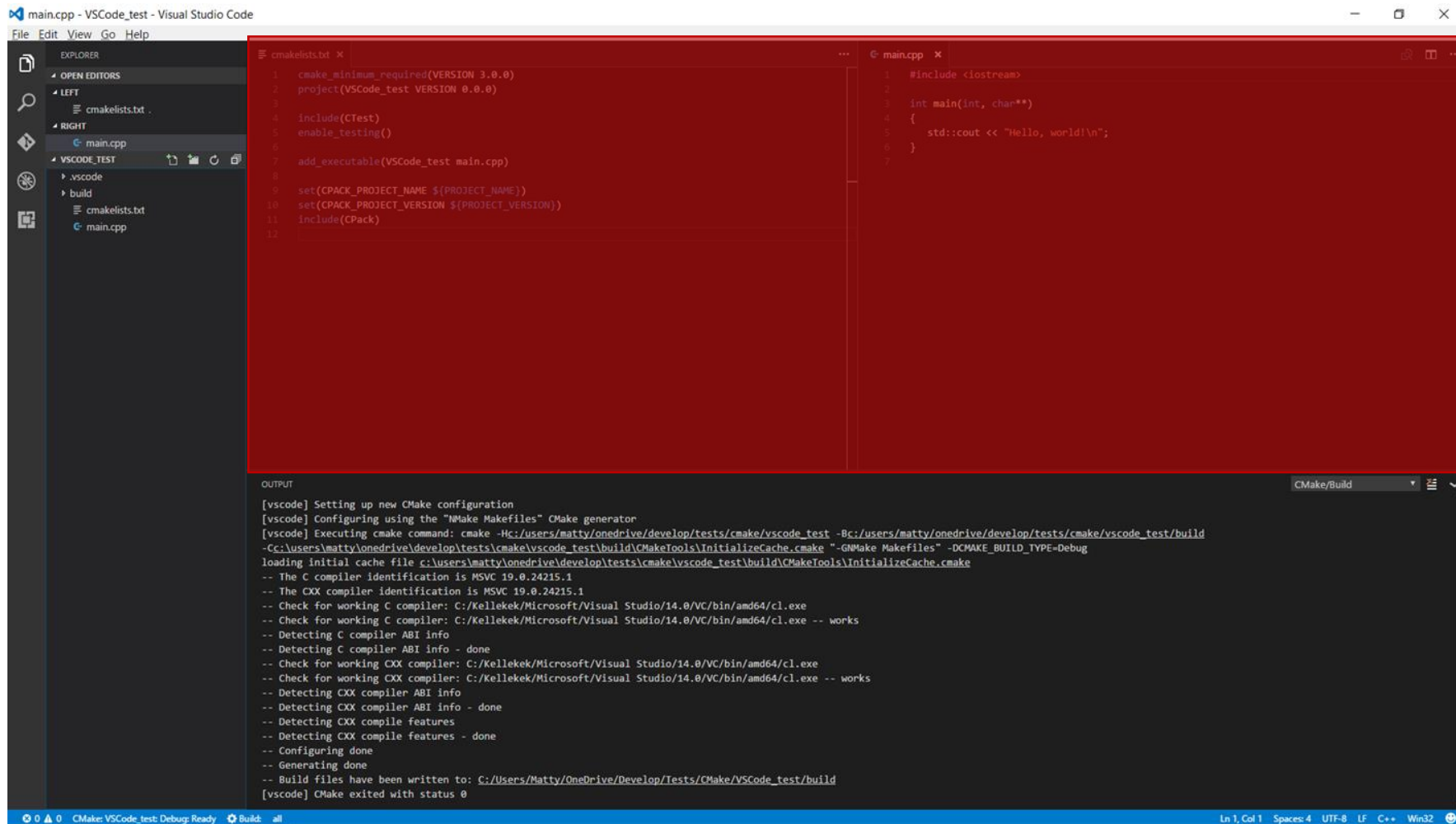
main.cpp
1 #include <iostream>
2
3 int main(int, char**)
4 {
5     std::cout << "Hello, world!\n";
6 }
7

OUTPUT
[vscode] Setting up new CMake configuration
[vscode] Configuring using the "NMake Makefiles" CMake generator
[vscode] Executing cmake command: cmake -Hc:/users/matty/onedrive/develop/tests/cmake/vscode_test -Bc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build
-Cc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake "-GNMake Makefiles" -DCMAKE_BUILD_TYPE=Debug
loading initial cache file c:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake
-- The C compiler identification is MSVC 19.0.24215.1
-- The CXX compiler identification is MSVC 19.0.24215.1
-- Check for working C compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe
-- Check for working C compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe
-- Check for working CXX compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/Matty/OneDrive/Develop/Tests/CMake/VSCode_test/build
[vscode] CMake exited with status 0
```

• Mi történt?



Repülő rajt



The screenshot shows the Visual Studio Code interface with the following content:

```
cmakeLists.txt
1 cmake_minimum_required(VERSION 3.0.0)
2 project(VSCode_test VERSION 0.0.0)
3
4 include(CTest)
5 enable_testing()
6
7 add_executable(VSCode_test main.cpp)
8
9 set(CPACK_PROJECT_NAME ${PROJECT_NAME})
10 set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
11 include(CPack)
12
```

```
main.cpp
1 #include <iostream>
2
3 int main(int, char**)
4 {
5     std::cout << "Hello, world!\n";
6 }
7
```

OUTPUT

```
[vscod] Setting up new CMake configuration
[vscod] Configuring using the "NMake Makefiles" CMake generator
[vscod] Executing cmake command: cmake -Hc:/users/matty/onedrive/develop/tests/cmake/vscode_test -Bc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build
-Cc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake "-GNMake Makefiles" -DCMAKE_BUILD_TYPE=Debug
loading initial cache file c:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake
-- The C compiler identification is MSVC 19.0.24215.1
-- The CXX compiler identification is MSVC 19.0.24215.1
-- Check for working C compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe
-- Check for working C compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe
-- Check for working CXX compiler: C:/Kellekek/Microsoft/Visual Studio/14.0/VC/bin/amd64/cl.exe -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/Matty/OneDrive/Develop/Tests/CMake/VSCode_test/build
[vscod] CMake exited with status 0
```

- Mi történt?
- A CMake Tools generált nekünk
 - egy egyszerű main.cpp fájlt
 - egy minimalista fordító szkriptet

Repülő rajt

cmakelists.txt

```
cmake_minimum_required(VERSION
3.0.0)

project(VSCode_test VERSION 0.0.0)

include(CTest)
enable_testing()

add_executable(VSCode_test main.cpp)
set(CPACK_PROJECT_NAME
${PROJECT_NAME})
set(CPACK_PROJECT_VERSION
${PROJECT_VERSION})
include(CPack)
```

main.cpp

```
#include <iostream>

int main(int, char**)
{
    std::cout << "Hello, world!\n";
}
```



Repülő rajt

cmakelists.txt

```
cmake_minimum_required(VERSION
3.0.0)

project(VSCode_test VERSION 0.0.0)

include(CTest)
enable_testing()

add_executable(VSCode_test main.cpp)
set(CPACK_PROJECT_NAME
${PROJECT_NAME})
set(CPACK_PROJECT_VERSION
${PROJECT_VERSION})
include(CPack)
```

main.cpp

```
#include <iostream>

int main(int, char**)
{
    std::cout << "Hello, world!\n";
}
```

F7



Fordítás



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows the project structure with files like `cmakelists.txt`, `main.cpp`, and build directories.
- cmakelists.txt:**

```
1 cmake_minimum_required(VERSION 3.0.0)
2 project(VSCode_test VERSION 0.0.0)
3
4 include(CTest)
5 enable_testing()
6
7 add_executable(VSCode_test main.cpp)
8
9 set(CPACK_PROJECT_NAME ${PROJECT_NAME})
10 set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
11 include(CPack)
12
```
- main.cpp:**

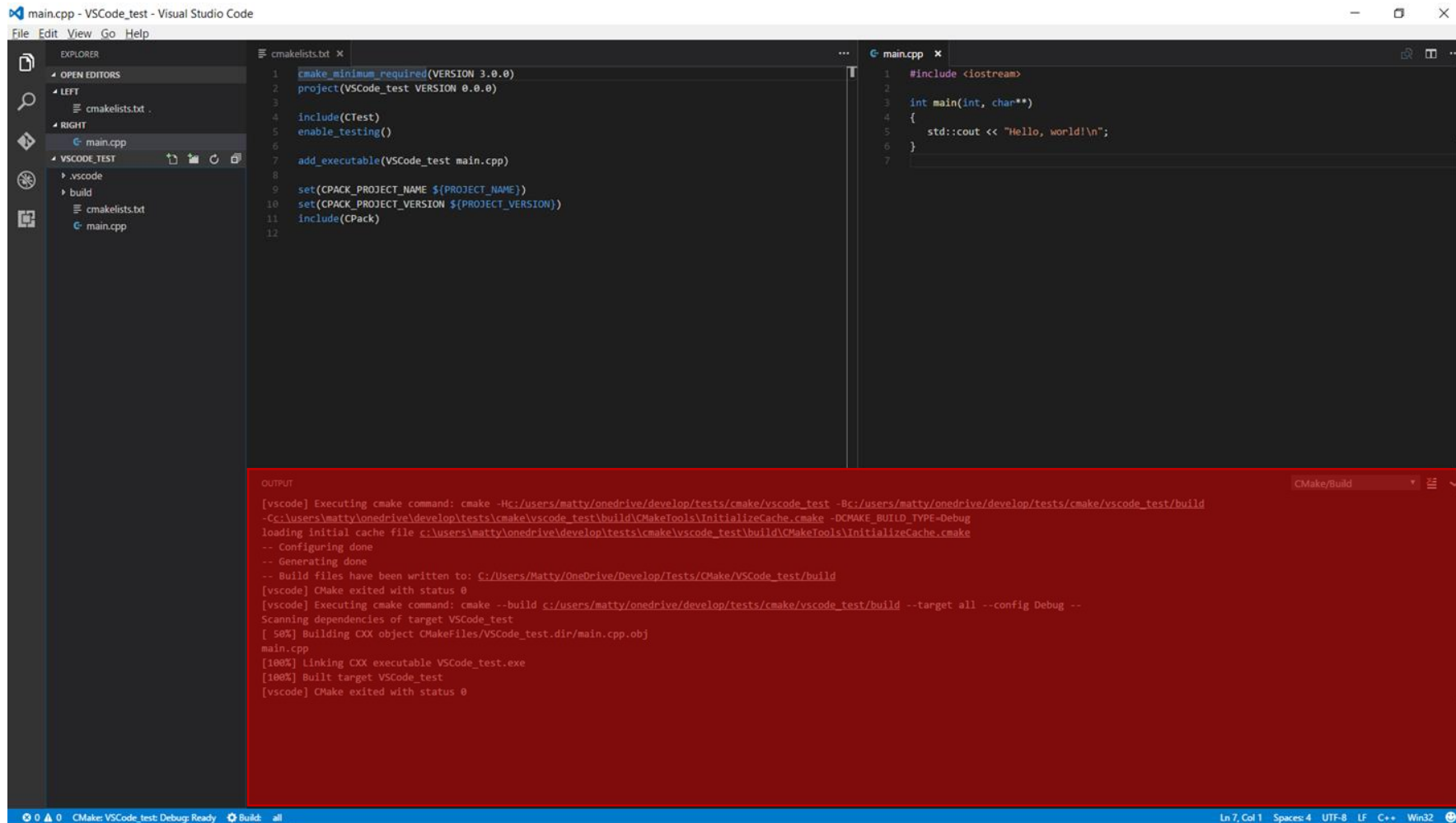
```
1 #include <iostream>
2
3 int main(int, char**)
4 {
5     std::cout << "Hello, world!\n";
6 }
7
```
- OUTPUT:**

```
[vscode] Executing cmake command: cmake -Hc:/users/matty/onedrive/develop/tests/cmake/vscode_test -Bc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build
-Cc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake -DCMAKE_BUILD_TYPE=Debug
loading initial cache file c:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/Matty/OneDrive/Develop/Tests/CMake/VSCode_test/build
[vscode] CMake exited with status 0
[vscode] Executing cmake command: cmake --build c:/users/matty/onedrive/develop/tests/cmake/vscode_test/build --target all --config Debug --
Scanning dependencies of target VSCode_test
[ 50%] Building CXX object CMakeFiles/VSCode_test.dir/main.cpp.obj
main.cpp
[100%] Linking CXX executable VSCode_test.exe
[100%] Built target VSCode_test
[vscode] CMake exited with status 0
```

• Mi történt?



- Mi történt?
- A CMake Tools
 - Meghívta helyettünk a build systemet
- Ugyanazt a kimenetet látjuk, mint ha mi magunk csináltuk volna



The screenshot shows the Visual Studio Code interface with the CMake Tools extension. The Explorer sidebar on the left shows the project structure with files like `cmakelists.txt` and `main.cpp`. The main editor area displays the `cmakelists.txt` file, which configures a C++ project named `VSCode_test` using CMake. The `main.cpp` file is also visible, containing a simple "Hello, world!" program. The Output window at the bottom shows the CMake build process, including the execution of the `cmake` command, the generation of build files, and the successful compilation and linking of the `VSCode_test.exe` executable.

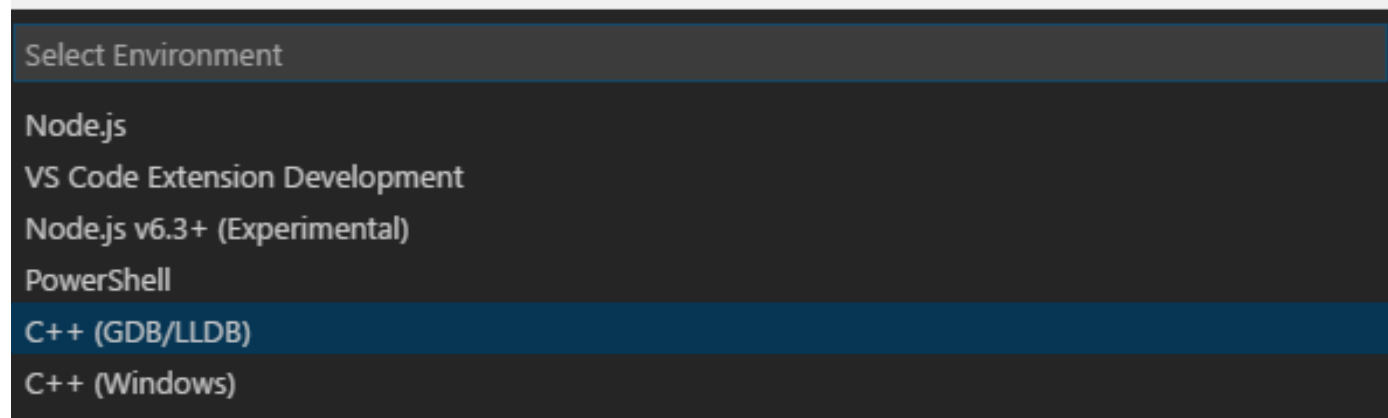
```
cmakelists.txt
1 cmake_minimum_required(VERSION 3.0.0)
2 project(VSCode_test VERSION 0.0.0)
3
4 include(CTest)
5 enable_testing()
6
7 add_executable(VSCode_test main.cpp)
8
9 set(CPACK_PROJECT_NAME ${PROJECT_NAME})
10 set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
11 include(CPack)
12
```

```
main.cpp
1 #include <iostream>
2
3 int main(int, char**)
4 {
5     std::cout << "Hello, world!\n";
6 }
7
```

```
OUTPUT
[vscode] Executing cmake command: cmake -Hc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build
-Cc:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake -DCMAKE_BUILD_TYPE=Debug
loading initial cache file c:/users/matty/onedrive/develop/tests/cmake/vscode_test/build/CMakeTools/InitializeCache.cmake
-- Configuring done
-- Generating done
-- Build files have been written to: C:/Users/Matty/OneDrive/Develop/Tests/CMake/VSCode_test/build
[vscode] CMake exited with status 0
[vscode] Executing cmake command: cmake --build c:/users/matty/onedrive/develop/tests/cmake/vscode_test/build --target all --config Debug --
Scanning dependencies of target VSCode_test
[ 50%] Building CXX object CMakeFiles/VSCode_test.dir/main.cpp.obj
main.cpp
[100%] Linking CXX executable VSCode_test.exe
[100%] Built target VSCode_test
[vscode] CMake exited with status 0
```

Hiba keresés (Cpp Tools módra)

- Nyomjuk meg az F5-öt
- Minden bővítmény és beépített képesség megkérdni, mit szeretnénk tenni a gyorsbillentyű lenyomásakor
- Linuxon
 - Válasszuk a „C++ (GDB/LLDB)” lehetőséget GCC vagy Clang használatához
- Windowson
 - Válasszuk a „C++ (Windows)” lehetőséget MSVC használatához
 - Válasszuk a „C++ (GDB/LLDB)” lehetőséget Clang használatához



Hiba keresés (Cpp Tools módra)



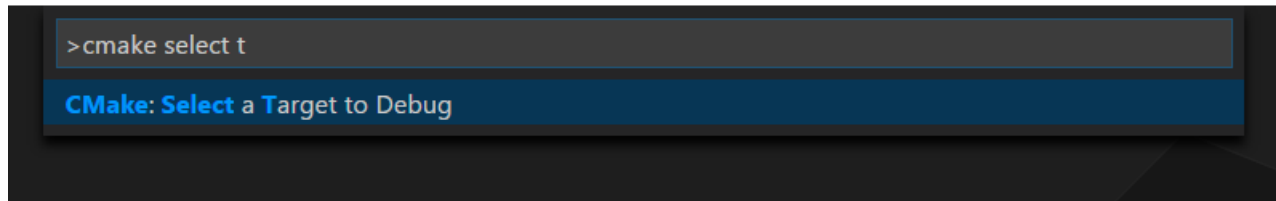
- Amikor először szeretnénk egy programot elindítani hibakeresés céljából, meg kell mondani a C++ bővítménynek hova került az futtatható állomány
 - Ez a rész fölösleges, ha magával a CMake Tools-zal indítjuk a hiba keresést (köv dia)
- Alapértelmezetten

`${workspaceRoot}/build/<project_name>`

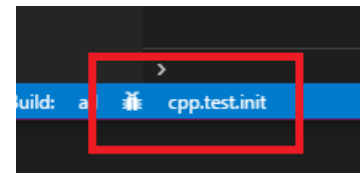
```
cmakelists.txt  launch.json x
1 {
2   "version": "0.2.0",
3   "configurations": [
4     {
5       "name": "C++ Launch (Windows)",
6       "type": "cppvsdbg",
7       "request": "launch",
8       "program": "${workspaceRoot}/build/VSCode_test.exe",
9       "args": [],
10      "stopAtEntry": false,
11      "cwd": "${workspaceRoot}",
12      "environment": [],
13      "externalConsole": false
14    },
15    {
16      "name": "C++ Attach (Windows)",
17      "type": "cppvsdbg",
18      "request": "attach",
19      "processId": "${command.pickProcess}"
20    }
21  ]
22 }
```

Hiba keresés (CMake Tools módra)

- Mivel a CMake Tools tudja hova kerülnek a binárisok és azt is tudja milyen fordítóval fordítottunk (tehát azt is, milyen hiba kereső képes a binárisokban turkálni), ezért képes a korábbi beállításokat megspórolni nekünk
 - Bővítmények közti kommunikációt használ
- A parancs palettáról hívjuk elő a következő parancsot

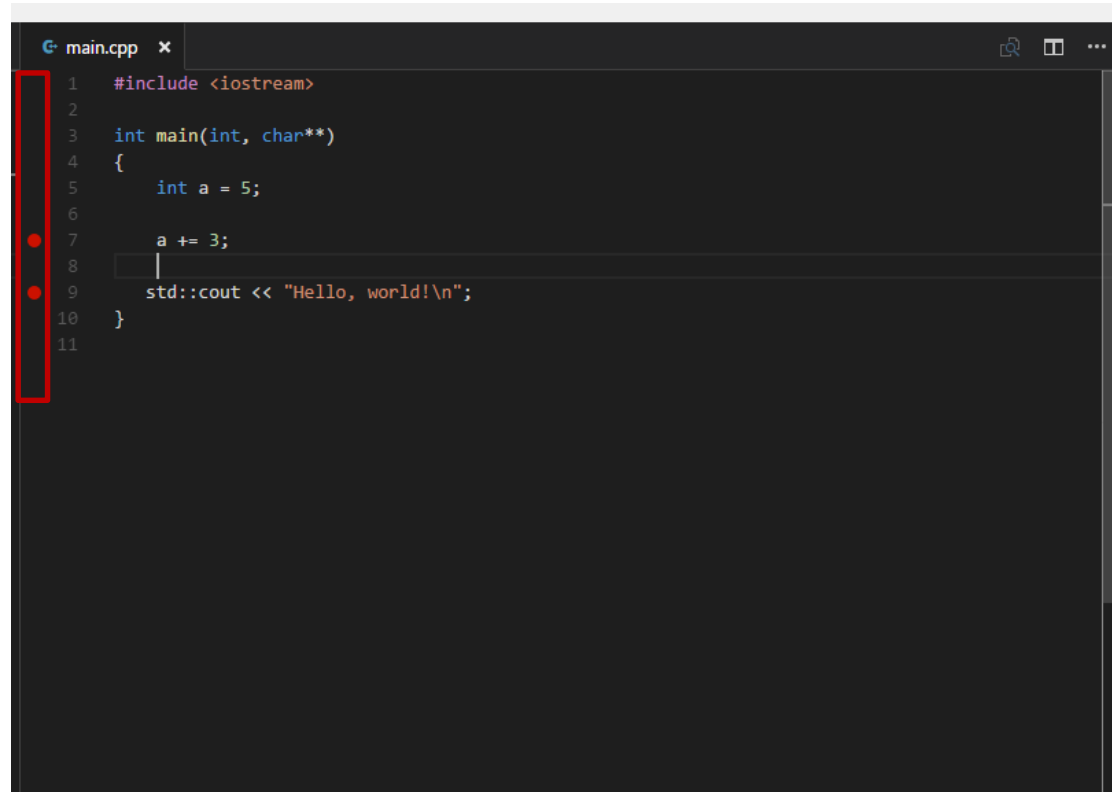


- Eztán ha kiválasztottuk az egyetlen futtatható állományunkat, megjelenik egy gomb a státusz sávon
 - Erre nyomva elindul a hiba keresés



Hiba keresés

- Pár sor kódot hozzáadva a forráshoz izgalmasabb lesz a hiba keresés
- A sor számok bal oldalára kattintva megakasztási pontok helyezhetők el
- Amikor a program végrehajtása ezekhez a pontokhoz ér megakad, hogy szemügyre vehessük az állapotát

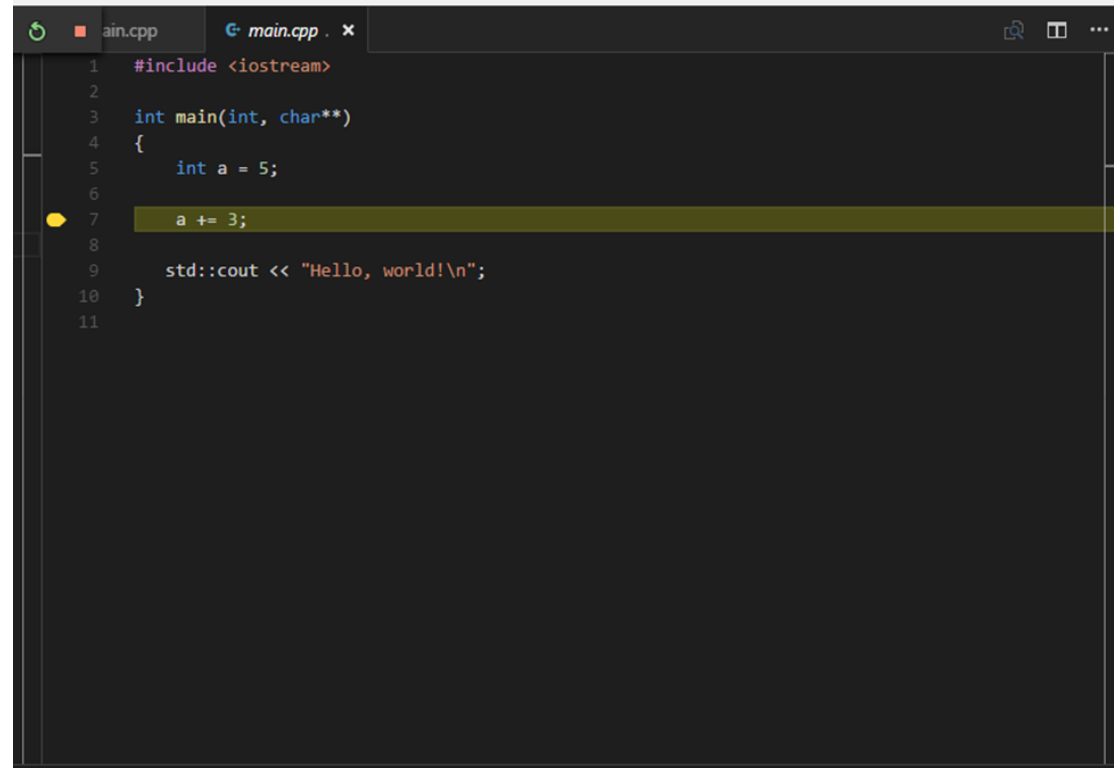


```
main.cpp x
1  #include <iostream>
2
3  int main(int, char**)
4  {
5      int a = 5;
6
7      a += 3;
8
9      std::cout << "Hello, world!\n";
10 }
11
```



Hiba keresés

- Ha újból megnyomjuk az „F5” gombot, akkor most már (és ezután mindig) rendszeren elindul a hiba keresés
- Ha az egeret változók fölé visszük, megtekinthetjük azok aktuális értékét

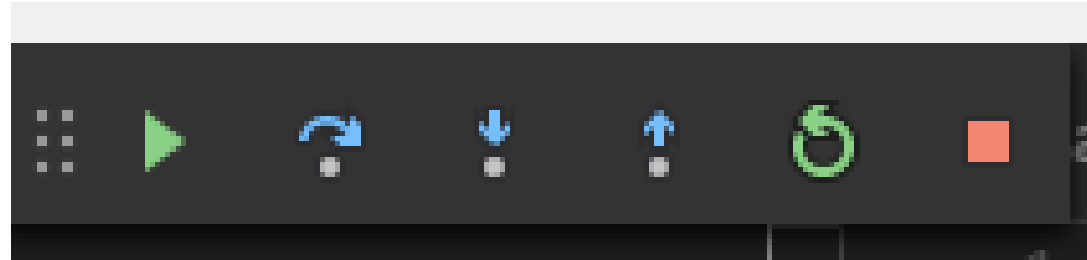


```
1  #include <iostream>
2
3  int main(int, char**)
4  {
5      int a = 5;
6
7      a += 3;
8
9      std::cout << "Hello, world!\n";
10 }
11
```



Hiba keresés

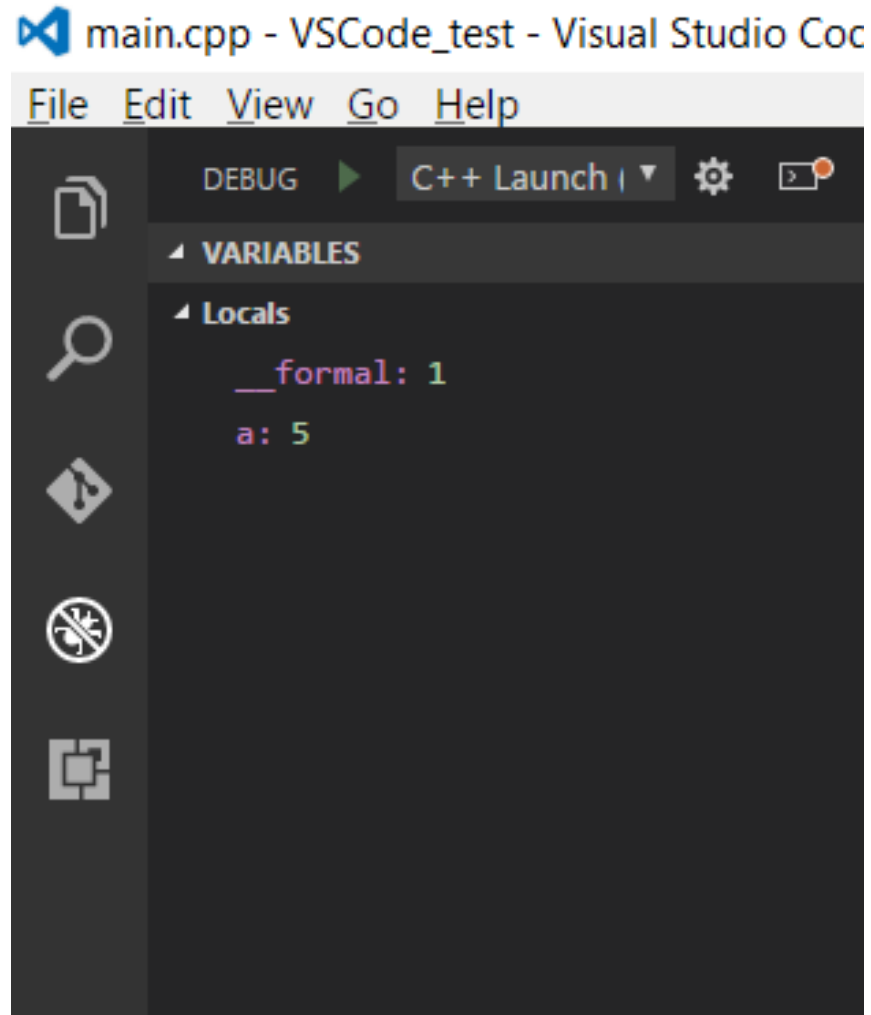
- Ha újból megnyomjuk az „F5” gombot, akkor most már (és ezután mindig) rendszeren elindul a hiba keresés
- Ha az egeret változók fölé visszük, megtekinthetjük azok aktuális értékét
- A középső panelen a következő parancsok láthatók



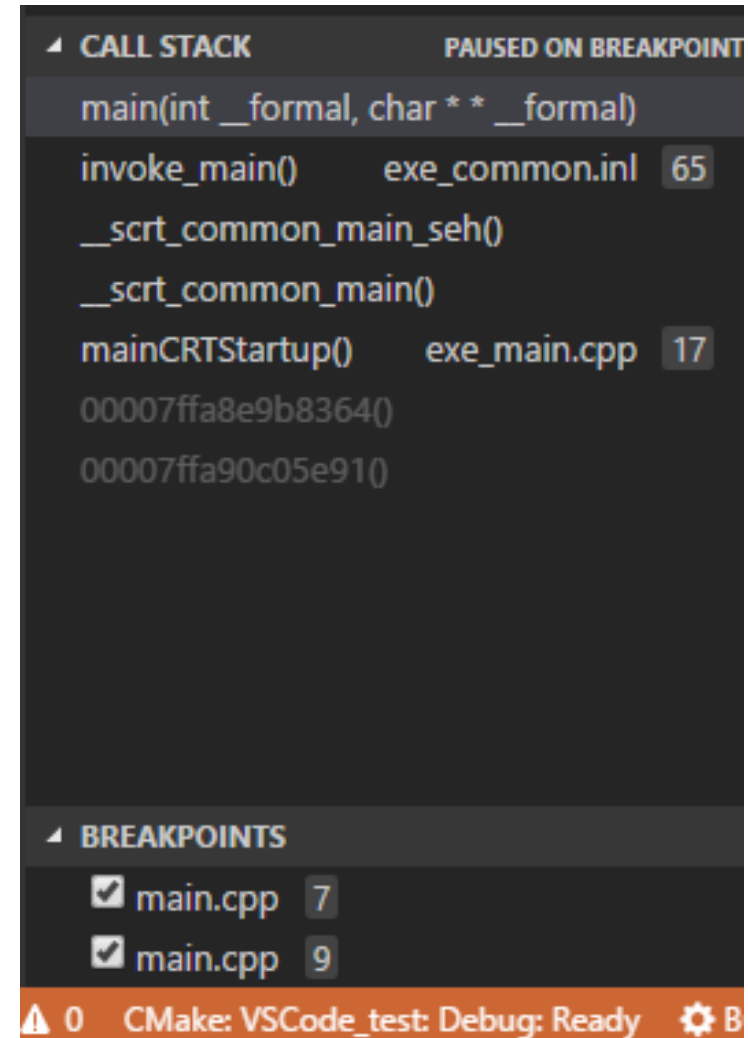
- Ugrás a következő pontra (F5)
- Állítás átlépése (F10)
- Állításba lépés (F11)
- Állításból kilépés (Shift+F11)
- Újraindít (Ctrl+F5)
- Menet leállítása (Shift+F5)

Hiba keresés

Helyi
változók



Hívás
verem

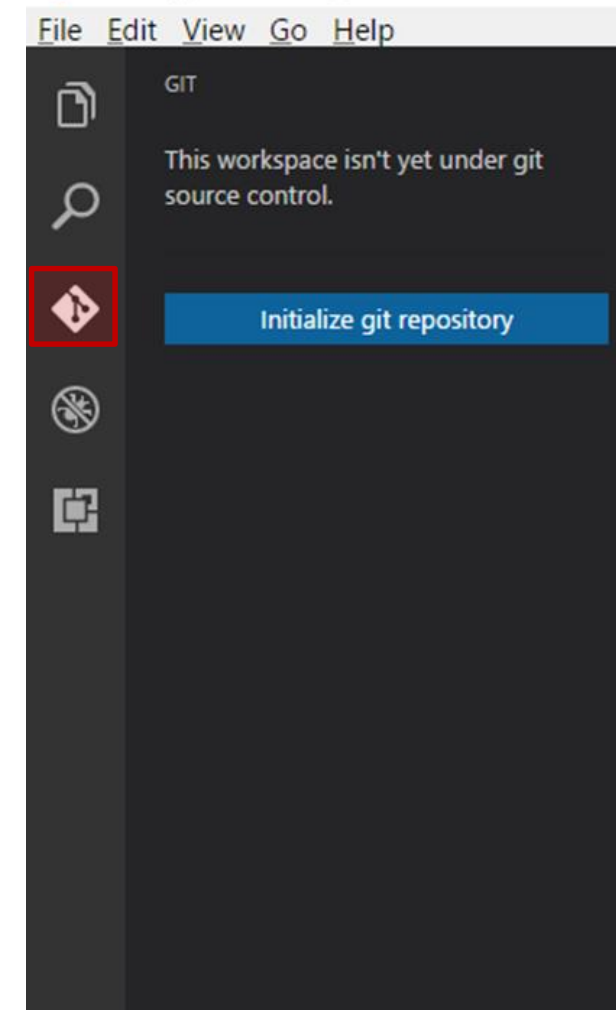


- A Git a legelterjedtebb verziókövető rendszer napjainkban
- Kifejezetten hasznos, ha szeretnénk
 - visszatérni a programkód egy korábbi változatára
 - használni egy működő verziót, amíg fejlesztünk egy új képességet
 - együttműködni másokkal ugyanazon a kódbázison
- A Git részletesebb bemutatását a tavalyi diákon már megtettük.



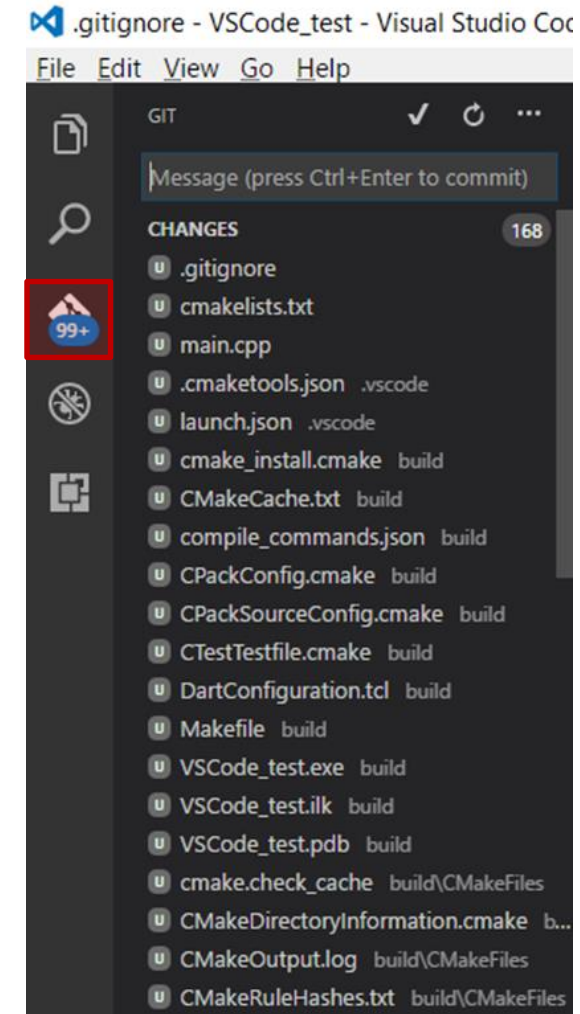
Verzió követés

- A Gitet olyan sok programnyelv művelője használja, hogy beépített funkció a Visual Studio Code-ban
- A CMake Quick Start projektünket eddig nem követtük Gittel
- Ha megnyitjuk a verziókövető menüpontot, git repot csinálhatunk a gyöker mappából
- Kezdetben nincs verziókövetés
 - Kattintsunk a **Initialize git repository** gombra



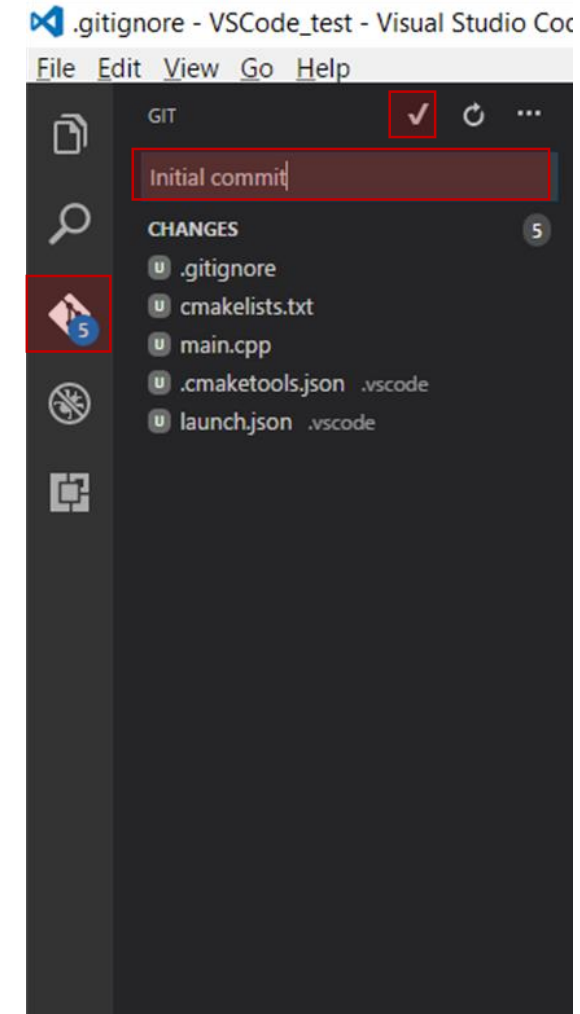
Verzió követés

- Ezt követően kismillió fájlt detektál majd a Git, ami még nem szerepel a repóban
 - Ez azért van, mert az alapértelmezett fordítási helyet használtuk, és a gyökérmappában lévő „build” mappába került minden fordítási mellékhatás
 - Ugyan a keletkező binárisok nem teljesen kerülnek átfedésbe a forrás fájlokkal (in-source fordítás), mégsem a legszebb gyakorlat
 - Szebb volna egy teljesen külön helyre fordítani (out-of-source fordítás)



Verzió követés

- Visszatérve a verziókövető fülre, most már csak az általunk létrehozott fájlok szerepelnek, mint új fájlok
 - Amennyiben másokkal együtt kollaborálunk vagy szeretnénk közzétenni a munkánkat, úgy a környezet-specifikus fájlokat is ki illik zárni a verziókövetésből
 - .vscode/
 - .gitignore (magát a .gitignore-it is ki lehet zárni)
 - Adjunk egy rövid megjegyzést a commithoz
 - A ✓ gomb megnyomásával kész az első commit
- Gratulálunk, kész az első Git repo Code-



- A legtöbb Git parancs elérhető a Parancs Palettáról
 - Ha beírjuk, hogy „Git”, akkor kilistázódik az összes Git parancs, amit a szerkesztő ki tud adni helyettünk
- A tavalyi diákon megtalálható a tipikus Git munkafolyamat és mikor érdemes használni azt
 - A többletmunka amit a Git jelent már közepes méretű projekteknél is megtérül
- Ha bizton szeretnénk tudni a kódot
 - Tartsuk a repot olyan mappában, amit egy felhős tárhelyszolgáltató is követ
 - Időnként toljuk ki a munkát egy megbízható távoli kiszolgálóra



Priváts vs. Publikus

Wigner Git

- A Wignernek saját GitLab kiszolgálója van
 - Központi Wigner hitelesítés (spam szűrő jelszó)
 - A GitLab a legelterjedtebb, ingyenes és privát alternatívája a GitHub-nak

Publikus Git

- A világ túlnyomó része a GitHub ingyenes szolgáltatásait használja
 - Könnyen megosztható másokkal

