



# Virtualizálás Linux alapon

Farkas Attila

# Tartalomjegyzék

1 Célok.....	1
2 Feltételek.....	1
3 Környezet felépítése.....	2
3.1 A rendszer konfigurálása.....	2
3.2 Virtualizáló környezet telepítése.....	3
3.2.a KVM.....	3
3.2.b XEN.....	3
3.2.c KVM környezet telepítése, gép létrehozása, konfigurálása.....	3
3.3 A vendég OS indítása automatikusan.....	7
3.4 Statikus IP cím Windows guest számára.....	8
3.5 Guest védelme tűzfalal.....	8
4 Windows és Linux összehangolása.....	9
4.1 Driverelek telepítése.....	9
4.2 C könyvtár megosztása.....	9
4.3 Megosztott C könyvtár felcsatolása Ubuntu alatt.....	10
4.4 Csatolási pont megosztása SAMBA szolgáltatással Ubuntu alatt.....	11

# 1 Célok

Az OmegaByte egyik feladata, hogy az általuk felügyelt számítógépek, amelyeken termelés szempontjából kritikus alkalmazások / mérések működnek, mindig üzemképesek legyenek. Ez adott esetben nem csak egy-egy kártya cseréjét, hanem a komplett alaplap újítását is jelentheti. A jövőben több gép esetén is várható ez utóbbi, ami sajnálatos módon megnehezíti a szóban forgó eszközök működtetését: ezek a gépek több évtizede működnek, tehát az alaplap cseréje az arra ültetett, ahhoz szorosan kapcsolódó periféria cseréjét is jelenti. Ráadásul a bennük elhelyezett bővítőkártyák több esetben nem támogatják a Windows XP-nél újabb operációs rendszereket, hiszen azok még valamilyenkor a Win95-Win98-as időszakban kerültek beszerzésre. További problémát okoz, hogy ezen kártyák cseréjéhez a komplett gyártósort kellene újraprogramozni, ami igen költséges és időigényes – hozzáátéve, hogy mindezt egyetlen kieső számítógép miatt kellene megtenni.

A helyzet megoldásaként egy virtuális környezetben futtatott XP került a döntéshozók látókörébe, ami hosszútávon is működőképes lehet: virtualizálni majdnem mindenféle eszközt lehet és ráadásul egy-két feltételt figyelemmel kísérve gyakorlatilag akár mikor cserélhető alatta a fizikai réteg.

## A virtuális gépeknek

- automatikusan el kell indulnia a számítógép bekapcsolásakor teljes képernyős üzemmódban,
- XP esetén külön védeni kell a vendég OS-t a különböző kibertámadásoktól
- a C meghajtót átlátszóvá kell tenni a Linux számára (írasi jogosultsággal).

# 2 Feltételek

A virtualizálásra szánt HOST számítógépnek meg kell felelnie a következő feltételeknek:

- **A processzor támogassa a VT-d technológiát**

*Ez a technológia szükséges a fizikai PCI és egyéb IO eszközök csatolására a guest felé*

- A számítógép rendelkezzen a szükséges erőforrásokkal, hogy a HOST OS-t és a Guest OS-t is ki tudja szolgálni!

*Az alaprendszer fogyasztása nagyban függ annak típusától is. Egy grafikai szobba megjelenésű verzió többet fogyaszt a memóriából és jobban használja a videokártyát is, míg egy szerényebb kinézetű esetén több erőforrást tudunk megtakarítani a vendég OS számára.*

## 3 Környezet felépítése

Az első és legfontosabb kérdés, hogy milyen disztribúciót válasszunk. Jelen esetben az Ubuntu alapú OS-ek között válogattam, mivel ehhez található meg a legtöbb információ az interneten. Először a LinuxMint Cinamon legújabb (20.02) verziójával kezdtem, de itt nem sikerült a PCI perifériákat csatolni – *az, hogy ez ténylegesen a LinuxMint hibája volt-e, nem lett feltárva*. Az OS hiba kiszűrése végett telepítettem a Xubuntu-t, azaz az Ubuntu Xfce felületével rendelkező verzióját. Ennek előnye a szerényebb felület és az ezzel járó alacsonyabb erőforrásigény.

**A lent leírt folyamatok eltérhetnek különböző disztribúciók esetén, de az Ubuntu alapú rendszerek esetén (LinuxMint is ilyen) ezen lépéseknek meg kell egyeznie!**

### 3.1 A rendszer konfigurálása

Az operációs rendszer alapértelmezetten nem támogatja a VT-d technológiát. Ahhoz, hogy ezt használni tudjuk, ahhoz engedélyoznünk kell:

1. A `/etc/default/grub` fájlt szerkesszük a következők szerint:
  - (a) Keressük meg az alábbi változót: `GRUB_CMDLINE_LINUX_DEFAULT`
  - (b) Az értékét szerkesszük úgy, hogy hozzá fűzzük az „`intel_iommu=on`” kifejezést.
  - (c) A változónk értéke tehát – saját gép példáján: `"quiet splash intel_iommu=on"`
2. Futtassuk le a grub konfigurációt frissítő alkalmazást, aminek köszönhetően az általunk beírt kernel paraméter bekerül a grub által ténylegesen használt konfigurációs állományába: `/boot/grub/grub.cfg`. Ezt a fájlt kézzel nem kell szerkesztenünk!

A futtatandó script: `update-grub`

3. A rendszer újraindítását követően a `/sys/kernel/iommu_groups` tartalmazza azokat az eszközöket, amelyeket megoszthatunk virtuális gépekkel.

**Fontos:** Az VT-d technológia támogatja egy eszköz kiosztását egyidejűleg több virtuális gép számára. Azonban ezt a kártyának is támogatnia kell, úgynevezett többfelhasználós módban kell működni. A régi kártyák erre nem képesek!

## 3.2 Virtualizáló környezet telepítése

A környezet tekintetében két alkalmazás merült fel: KVM és XEN.

### 3.2.a KVM

Ez a technológia a GUEST OS kernel szintjét beépíti a HOST OS kernel szintjébe, aminek köszönhetően a GUEST OS alig veszít a hardverek tényleges teljesítményéből. Egyetlen hátránya, hogy inkább Linuxok virtualizálásával működik, mivel a Windows driverek hiányosak. Legtöbbször a videokártyákkal merülnek fel problémák, így első körben ezzel indultunk el.

### 3.2.b XEN

Ez a technológia a GUEST OS kernel szintjét a HOST OS kernel szintje felett tartja és minden hardvert virtualizál. Ezáltal a GUEST OS kevésbé tudja kihasználni a hardverek teljesítményét, mint a KVM: a kérések folyamatosan egy köztes rétegen kell keresztül menjenek. Előnye, hogy teljes mértékben támogatja a Windowsos rendszereket.

### 3.2.c KVM környezet telepítése, gép létrehozása, konfigurálása

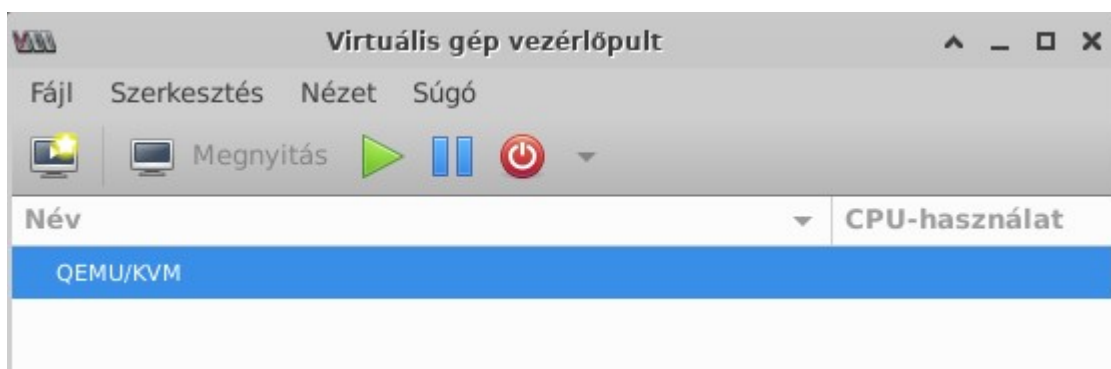
1. Linuxos környezetben a KVM virtualizáláshoz a QEMU emulátoron keresztül férhetünk hozzá. Ezt kell telepítenünk tehát:

```
sudo apt install qemu-kvm
```

2. Ezen kívül érdemes telepíteni a QEMU grafikus paneljét, amin keresztül konfigurálhatjuk és vezérelhetjük a virtuális gépeinket:

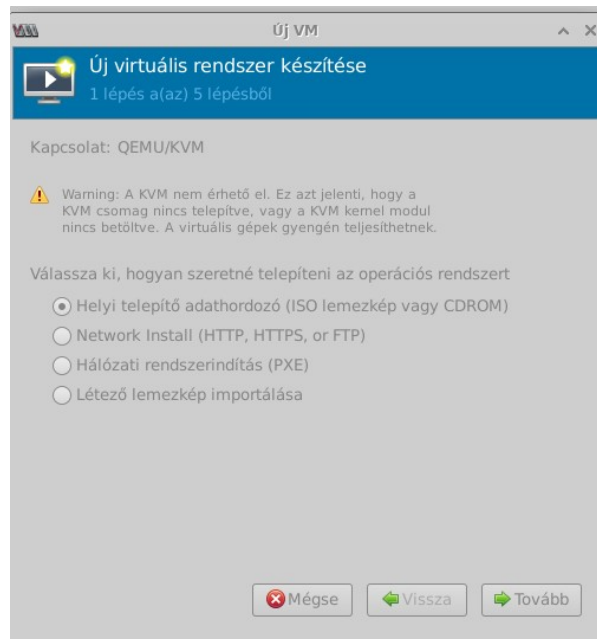
```
sudo apt install virt-manager
```

3. A telepített GUI-t a *Start>Rendszer>Virtuális gép vezérlőpult* menüpontban találjuk meg.



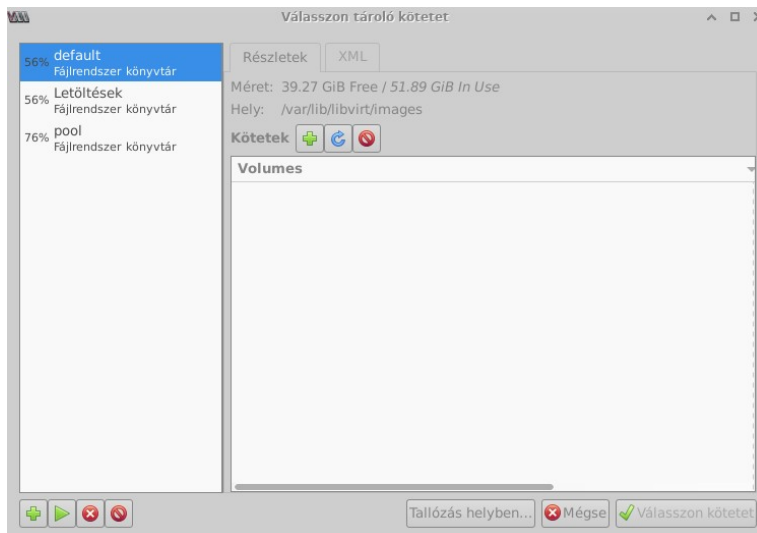
1. Ábra: Az alkalmazás felülete

4. Hozzunk létre új virtuális gépet a *Fájl>Új virtuális gép* menüpontban.



2. Ábra: Új virtuális gép hozzáadása

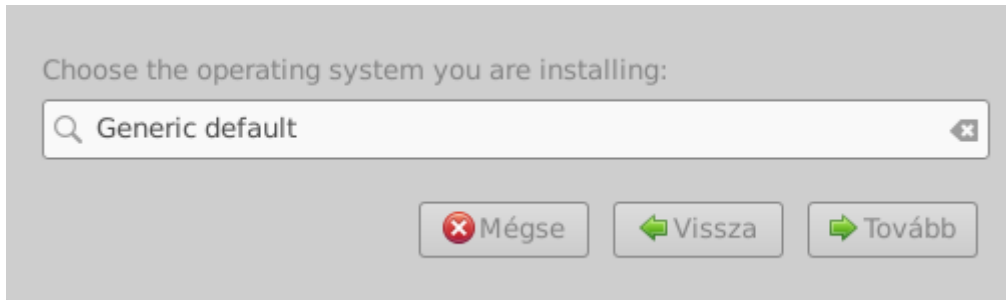
5. Válasszuk a létező merevlemez importálása funkciót, hiszen rendelkezésünkre áll a XP telepített változata. Kattintsunk a tovább gombra.



3. Ábra: Merevlemez képfájl tallózása

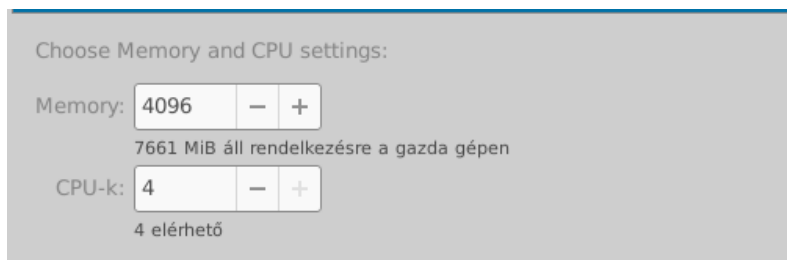
6. Tallózzuk be a XP lemezképfájlját:
1. Válasszuk ki a megfelelő „pool”-t, vagy hozzunk létre újat. Az XP lemezképfájlt másoljuk a megfelelő mappába – vagy hozzunk létre olyan „pool”-t, ami az XP-t tartalmazó könyvtárra mutat.
  2. A jobb oldali ablakban válasszuk ki az XP-hez tartozó merevlemez képfájlt.

7. A képernyő alján válasszuk ki a „Generic default” opciót, mint operációs rendszert.



4. Ábra: Operációs rendszer típusának beállítása

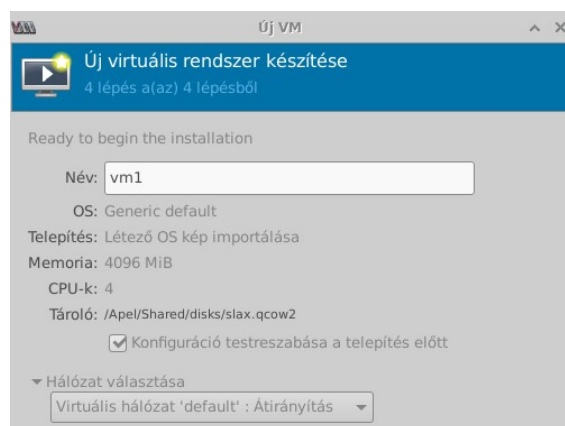
8. A következő oldalon konfiguráljuk az erőforrásokat. Vegyük figyelembe a vendég OS típusát, ne csatoljunk át feleslegesen se memóriát, se processzort. Egy 32 bites rendszer 4096MB (azaz 4GB) memória felett nem tudja kezelni a területet – hibát nem okoz, de kihasználni sem tudja. Tájékozódjunk a vendég OS-ról!



5. Ábra: A vendég PC erőforrásainak konfigurálása

Érdeemes továbbá erőforrásokat hagyni a gazda rendszernek is, hogy a virtuális gép felügyeletét megfelelően el tudja végezni – és az egyéb teendőit is – a vendég OS túlterhelése esetén is.

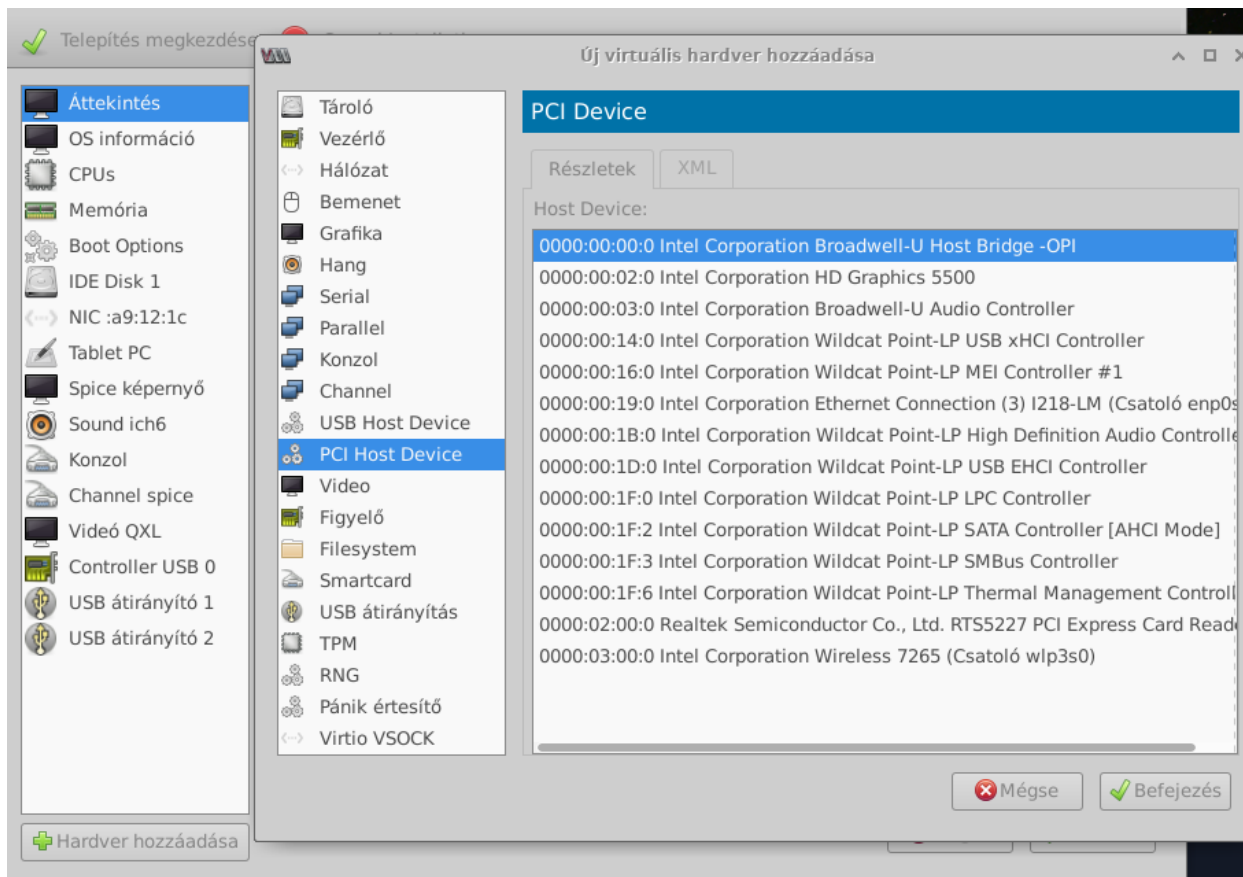
9. A következő oldalon elnevezhetjük a virtuális gépet és ellenőrizhetjük a beállításainkat. Válasszuk az átirányított hálózatot – így felügyelhetjük a vendég gép forgalmát iptables segítségével. Kapcsoljuk be a „Konfiguráció testreszabása a telepítés előtt” funkciót, így a számítógép elindítása előtt további eszközöket csatolhatunk gépünkhöz.



6. Ábra: A beállítások összegzése, elnevezés

10. A hardver hozzáadásnál válasszuk ki a „*PCI Host Device*” opciót, majd a jobb oldali menüben válasszuk ki azt az eszközt, amit a vendég géphez át szeretnénk csatolni.

**Figyelem:** Az át kapcsolt kártyát a host gépen nem tudjuk tovább használni, amennyiben nem támogatja a többfelhasználós módot!



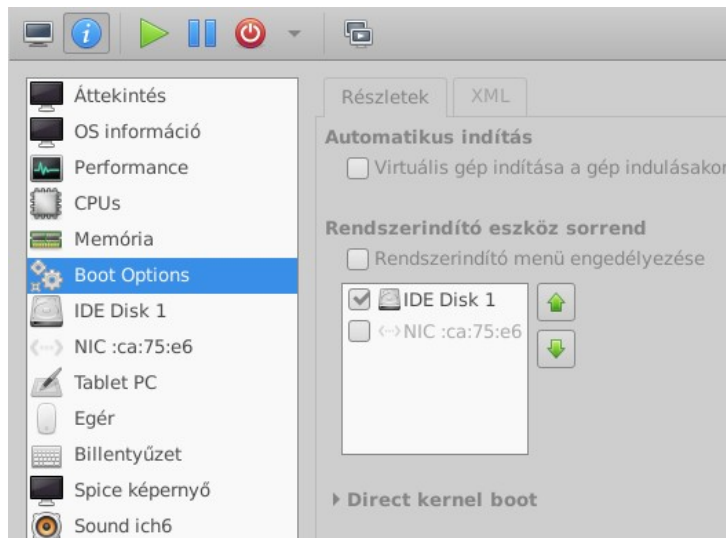
7. Ábra: PCI eszközök továbbítása vendég gép felé

11. A műveletet annyiszor ismételjük meg, ahány kártyát át szeretnénk kapcsolni. Járjunk el figyelemmel: a PCI Host Bridge is átkapcsolható, de ez esetben a host gépen szét eshet a rendszer, mivel ez a PCI vezérlőt egy az egyben át akarja adni. Ne kapcsoljunk át olyat, amiről nem tudjuk, hogy mit csinál.

12. A megfelelő kártyák csatolását követően elindíthatjuk a vendég telepítését a bal felül található „*Telepítés megkezdése*” vagy a jobb alul található „*Alkalmaz*” gomb segítségével.



### 3.3 A vendég OS indítása automatikusan



8. Ábra: A virtuális gép indítása automatikusan

1. A virtuális gép indítását a virt-manager GUI-ban konfigurálhatjuk. Nyissuk meg a létrehozott számítógép konfigurációját (duplakattintással a gépen) és a monitor helyett váltsunk az információs ablakra.
2. Aktiváljuk a „*Boot options*” menüpontban található „*Virtuális gép indítása a gép indulásakor*” funkciót.
3. A VNC megjelenítő automatikus futtatásához hozzuk létre a `/etc/profiles.d/virtviewer.sh` állományt – megtalálható az `OB_FA-tools.iso`-ban.

A script 30 másodperces várakozást követően teljes képernyős módban elindítja a vendég OS megjelenítését – fontos, hogy a gépet a libvirt indítja a rendszer bekapcsolásának megadott szakaszában, így mi gyakorlatilag a monitort kapcsoljuk be ebben a lépésben.

A parancs utolsó paramétere a virtuális gép neve, esetünkben `vm1`.

A várakozási idő szerkeszthető annak függvényében, hogy mennyi idő alatt éri el a rendszer a végleges felbontás állapotát – hibás érték esetén vagy indokolatlanul sokat kell várnunk, hogy használni tudjuk a számítógépet vagy alacsonyabb felbontással kapjuk meg az első képet, így a megjelenítő kilép a teljes képernyős módból – látszódní fog az ablak kerete.

```
sleep 30s && virt-viewer -f vm1 &
```

### 3.4 Statikus IP cím Windows guest számára

Ahhoz, hogy a vendég rendszer megosztott mappáját megbízhatóan fel tudjuk csatolni, érdemes statikus IP-t beállítani a QEMU DHCP szerverében. Ehhez az alábbi lépésekre lesz szükségünk:

1. Másoljuk ki a létrehozott virtuális gép MAC címét a megfelelő NIC-ről
2. Kérdezzük le az elérhető hálózatokat, amelyeket a QEMU kezel

```
virsh net-list
```

3. Válasszuk ki a kezelni kívánt hálózatot – valószínűleg „default”

```
virsh net-edit default
```

4. Az alábbi sort szűrjük be – a range már szerepelni fog a fájlban

```
<range start='192.168.122.100' end='192.168.122.254' />  
<host mac='<mac>' name='<vm_név>' ip='<vm_ip>' />
```

5. Mentsük el a fájlt, majd indítsuk újra a hálózatot

```
virsh net-destroy default  
virsh net-start default
```

**A „default” mindenhol az aktuális rendszernek megfelelően módosítandó a hálózat nevére!**

A virtuális gép újraindítását követően a <vm\_ip>-t fogja megkapni.

### 3.5 Guest védelme tűzfalal

Ahhoz, hogy a vendég OS internetelérést korlátozzuk, a linuxon elérhető iptables csomagra van szükségünk. Ezzel a tűzfal szabályokat tudjuk manipulálni.

Ahhoz, hogy a szabályrendszert vissza tudjuk tölteni újraindításakor, ahhoz az iptables egy kiegészítőjét, az iptables-persistent-et kell telepítenünk.

```
apt install iptables-persistent
```

Meg fogja kérdezni, hogy mentse-e az aktuális beállításokat, feleljünk „Nem”-mel. A OB FA-tools.iso-ban megtalálható a szükséges fájl, amelyet a /etc/iptables/rules.v4 állományba kell menteni.

## 4 Windows és Linux összehangolása

Ha kész a virtuális gép telepítése és elindul a Windows, akkor a qemu által virtualizált eszközök illesztőprogramjait fel kell telepíteni. **Esetünkben az előzetes információgyűjtésnek megfelelően nem minden eszközhöz találtam drivert!** Ezen kívül fel kell telepíteni a fizikai rendszerből átkapcsolt különböző eszközöket is, mielőtt használatba vennénk. Az Én általam használt gépben pl. a profibus kártya drivere megtalálható volt a virtio által kiadott driver pack-ban (PCISerial), de amíg ezt nem telepítettem fel, addig minden indításnál hibát írt ki, mert a mérést végző szoftver nem tudott kommunikálni.

### 4.1 Driverelek telepítése

A driverek telepítésére két megoldási lehetőségünk nyílik, amivel biztosra tudunk menni:

1. A rendszer indításakor felugró illesztőprogramot telepítő alkalmazásban ki tallózzuk a drivert tartalmazó mappát
2. Az eszközközkezelőben kitallózzuk a telepíteni kívánt eszközhöz tartozó illesztőprogramot. Az eszközközkezelő a Winkey+Pause/Break billentyűkombinációval nyitható meg.

Mindkét esetben ugyanaz történik a háttérben!

A PCI Host Bridge, az USB vezérlők, a VGA illesztők telepítése mind sikertelen, de feltehetően ez nem fog a jövőben problémát okozni. (Az USB-s eszközt átcsatolva olvassa és írja azt, a felbontás megfelelő, játék és egyéb komolyabb grafikai processzort igénylő alkalmazás nem lesz rajta).

### 4.2 C könyvtár megosztása

A C könyvtár megosztásához létrehoztam egy felhasználót: **FolderShare**, melynek jelszava: **12345**. Ezzel a felhasználóval tudunk SAMBA klienssel felcsatlakozni, hogy írhasuk és olvashasuk a virtuális gépünk gyökérkönyvtárát.

**Fontos:** Ezen megoldás csak akkor nem jelent biztonsági kockázatot, ha a windows maga nem elérhető az internetről/belső hálózatról. Amennyiben olyan gépet helyezünk el, amely rendelkezik a hálózatról használható IP-vel, különösen ha az internet felől is nyitott (akár publikus) címmel, akkor válasszunk más alkalmazást a SAMBA helyett: SFTP, FTPS stb.

A könyvtár megosztásához az alábbi beállításokat kell végrehajtanunk:

1. Vezérlőpult>Mappa beállítások>„Egyszerű fájlmegosztás használata”  
Vegyük ki a menüpont mellől a pipát
2. Sajátgép>C meghajtó <jobb gomb>, megosztás fül
  1. Hozzuk létre új megosztást – a rendszer automatikusan megosztja C\$ néven – és oszszuk meg „C” néven.

2. Állítsuk a megosztási beállításokat „Mindenki”-re.
3. Navigáljunk ugyanezen ablakon a „Biztonság” fülre – ez akkor jelenik meg, ha az első lépést végre hajtottuk, különben az egyszerű megosztás miatt nem látható.
  1. Kattintsunk az „Új felhasználó” gombra, majd a képernyő alján a „keresés”-re. Ezután válasszuk ki a FolderShare felhasználót és kattintsunk az „Ok” gombra.
  2. Állítsuk be a jogosultságát „teljes”-re és „OK” gombbal aktiváljuk a módosításokat.
  3. A beállítás végrehajtása egy kis időbe kerülhet, mivel ilyenkor az összes olyan könyvtáron szerkeszti ezeket a beállításokat, amelyek ettől öröklik közvetetten vagy közvetve.

Ezt követően csatlakozhatunk a számítógépünkhöz a FolderShare felhasználó segítségével.

**Másik lehetőség,** hogy ez előre elkészített batch fájlt lefuttatjuk rendszergazda jogosultsággal.

### 4.3 Megosztott C könyvtár felcsatolása Ubuntu alatt

A Windows által megosztott könyvtárat SAMBA kliens segítségével mount-oljuk a rendszerbe. Ehhez hozzuk létre a csatolási ponthoz szükséges könyvtárat és telepítsük fel a szükséges szolgáltatásokat. A módosított fájlok megtalálhatók az *OB\_FA-tools.iso* állományban – bemásolni nem szabad, mintának használható.

1. Mappa létrehozása

```
sudo mkdir -p /media/virt/xp01
```

2. Szükséges szoftverek telepítése

```
sudo apt install cifs-utils
```

3. Csatlakozás megosztott könyvtárhoz:

```
sudo mount -t cifs -o  
username=FolderShare,password=12345 //<xp_ip>/C  
/media/virt/xp01
```

4. Amennyiben szeretnénk ezt újraindításkor csatolni:

1. Helyezzük el a következő sort a /etc/fstab állományba:

```
//<xp_ip>/C /media/virt/xp01 cifs  
user=FolderShare,password=12345,vers=1.0,uid=1000,  
file_mode=0700,dir_mode=0700 0 0
```

2. Készítsünk egy cron bejegyzést a /etc/crontab állományban:

```
@reboot root sleep 3m && mount //<xp_ip>/C
```

## 4.4 Csatolási pont megosztása SAMBA szolgáltatással Ubuntu alatt

Erre a szolgáltatásra azért van szükség, hogy a Windows-t magát teljesen el tudjuk szigetelni mind a belső hálózattól, mind az internettől. Így gyakorlatilag a Windows csupán a Linuxos HOST számítógéppel áll kapcsolatban, amit folyamatosan frissítve naprakésznek mondhatunk a különböző támadások ellen. Azonban ahhoz, hogy a Windows fájlrendszerét távolról is lehessen módosítani, ahhoz a Linux mappáját meg kell osztani saját szolgáltatásával, amelybe az xp-s fájlrendszert bele kapcsoljuk.

A megosztáshoz az alábbi lépéseken kell végig mennünk:

1. Telepítenünk kell a SAMBA szerveret, mivel a Windows gépek ezt a szolgáltatást használják. Így Windows-os gépről is fel tudjuk majd csatolni ezt a mappát.

```
sudo apt install samba
```

2. Konfigurálnunk kell a szolgáltatást, hogy ossza meg az XP-s csatolási pontot

A lenti konfigurációnak köszönhetően a megosztás csak a *vm* user-en keresztül érhető el – ezt cserélni kell az aktuális gép felhasználójára –, írási és olvasási jog egyaránt adott. A megosztás neve a szögletes zárójelen belül található: shared.

```
[shared]
comment = XP mount point sharing
browseable = yes
path = /media/virt/xp01
guest ok = no
public = no
writable = yes
read only = no
valid users = vm
```

3. Létre kell hoznunk a megosztási felhasználó számára egy jelszót, amit a SAMBA szolgáltatással használ – nem kell, hogy megegyezzen a login jelszóval.

```
smbpasswd -a test
```

4. Újra kell indítani a szolgáltatást

```
systemctl restart smbd
```

Ha mindent jól csináltunk, akkor a szolgáltatás elérhető és a Windows-os fájlrendszer felcsatolható a <LinuxIP>/C domainen keresztül.