



1. Írja le, mit jelent memória esetén a dinamikus allokáció!

- Dinamikus allokáció:
 - Ami memóriát nem használ a vendég, azt ne is kapja meg
 - Gyakorlati haszna önmagában elenyésző, a legtöbb OS az összes szabad memóriát disk cache-nek használja
 - Háttértárra swappelhetők a lapok a vendég OS tudta nélkül

2. Írja le a Bináris fordítás működését!

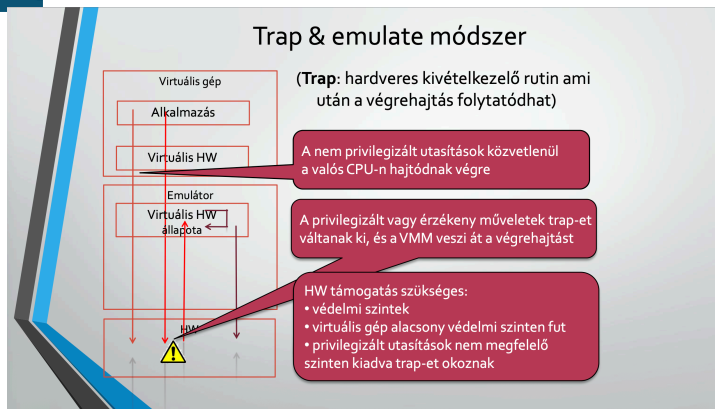
Bináris fordítás (binary translation)

- Mit tegyünk a problémás utasításokkal?
- Szoftveres virtualizáció **bináris fordítással** (VMware megoldása)
 - Egy JIT fordító a végrehajtás előtt végignézi a kód szegmenst és kicseréli a problémás utasításokat pl. SYSCALL-ra, vagy valami egyéb kódrészletre
 - Kicserélhet egyéb, amúgy elfogható utasításokat is rögtön a kezelő kódrészlettel, hogy elkerülje a felesleges hívást a VMM-be (inline translation)
 - Mivel a kód hossza megváltozhat ezért a kódra mutató pointereket (jump, branch utasítások) is módosítani kell
 - A végrehajtás menete: elő-fordítás + virtualizált végrehajtás elfogással + elfogott utasítások emulációja
- Teljesítmény?
 - Nem teljes fordítást végez, az utasítások többségét változatlanul hagyja
 - A JIT fordító minden kódrészletet csak az első futáskor jár be, ismételt futtatáskor már cache-eli a már módosított kódrészleteket
 - Optimalizálással csökkenthető a VMM-be történő hívások száma

3. Mit jelent a JIT, és mi a feladata?

- „Éppen időben” fordító (JIT compiler „just in time”) – végrehajtás előtt egy fordító feldolgozza a virtuális gép soron következő utasításait, és kódot generál belőle, ami az eredetivel ekvivalens viselkedést mutat
 - Közvetlenül futtatható kódot generál, cache-elési eljárásokkal gyors lehet
 - Nehéz implementálni, nem hordozható
 - Izolációt nem automatikusan biztosítja
 - Példa: QEMU x86 emulátor (és Java VM, .NET CLR is ilyen)

4. Írja le mi a lényege a Trap&Emulate módszernek!



5. Mi a három tanult lehetőség virtualizációra? Röviden miben különböznek, milyen konkrét módszert, lépéseket használnak?

- **Szoftveres virtualizáció**
 - (Trap and Emulate + bináris fordítás)
- **Paravirtualizáció**
 - (vendég OS kódját módosítjuk, hogy tudjon a virtualizációról, és hívhassa a VMM-et)
- **Hardveres virtualizáció**
 - (Trap and Emulate, teljesen hardveres támogatással)

6. Mik a VSAN előnyei, hátránya egy hagyományos infrastruktúrával szemben?

- **Azért ez nem olcsó!**
- ## VSAN
- **De! (pros)**
 - Szoftveres megoldás, könnyen frissíthető (új feature-ök)
 - Ugyanazon felületet alatt kezelhető
 - Könnyen bővíthető
 - Kevesebb eszköz -> kevesebb költség
 - Különböző policy-k (védelem (protection, security), performance)



(apróbb hátránynak ide mondható az is, hogy vannak hardverbeli kritériumai, minimum miből mennyi kell (azt a részt vágtam le az előző diáról, nehogy valaki azt higgye, hogy azt az angol felsorolást vártam el ide. De összefoglalásképpen hátránynak elfogadható az is))

7. Egy hagyományos üzemeltetési környezettel szemben a WSL miben különbözik? Mind az ötöt sorolja fel!

WSL

- Hagományos üzemeltetési környezettel szemben:
 - Egy könnyed utility VM automatikusan kezeli itt az erőforrásokat
 - Adott esetben a WSL automatikusan leállhat... (pl.: webserver)
 - Alapértelmezetten elérhetőek a host Windows fájljai (/mnt directory)
 - WSL Windows exe-t is tud kezelni
 - Kernel automatikusan frissítve van
 - Máshogy érhető el a GPU, mint egy szokásos Linux rendszeren

8. Mi az a Jumbo Frame, mire kell figyelni ennek használatakor?

Jumbo Frame

- Hardveres támogatás kell hozzá
- Az ethernet keret 1500 bájtos max payloadját kitolja 9000-ig
- Az eszközökön a jumbo frame engedélyezése, ha elfelejtjük el fogják dobni az eszközök mivel nem az ethernet szabványának megfelelően van formázva a csomag
- Operációs rendszerekben MTU átállítása (Max Transmission Unit)

9. Mik a konténerek "hátrányai"? írjon 3-at!



“Hátrányok”

- Docker csak egy processzt használ (multiprocess bukta)*
- Több konténer egy szolgáltatáshoz
 - Nginx konténer
 - Mysql konténer
 - PHP konténer stb...
- Nincs login script, SSH daemon, monitoring agent stb.
- Read-only konténer (snapshot jellegű működés)

10. Mi a hasonlóság és a különbség a hibrid és a közösségi felhő között?

Hibrid Felhő

- A hibrid felhő, mint neve is mutatja, **megpróbálja a két előző fajta legjobb tulajdonságait egyesíteni**. Lényege, hogy egy adott vállalat rendelkezik egy adott felépítésű privát felhővel, és amennyiben van olyan **időszak**, illetve élethelyzet, **amikor az átlagnál jóval magasabb a terhelés**, nem szükséges drága eszközök beszerzése.
- Megoldásként, ha ezek a terheléstöbbletek ismét mintát követnek, úgy erre az időszakra bérbe vehet privát felhőből erőforrásokat, melyeket a saját rendszerhez illesztve a megfelelő API-n keresztül, akár megtöbbszörözheti a privát felhő erőforrásait. Kérdésként itt is felmerülhet a biztonság, de az illesztés során meg lehet adni olyan policy-kat melyek megakadályozzák a kritikus adatok privát felhőbe vándorlását.
- Tipikus ilyen helyzet például az egyetemek tantárgy- illetve vizsgajelentkezései. Hiszen sok évi átlag szolgálat mintának akár időpontokra, akár a terhelés mértékére, így nagy segítséget nyújthat az IT részlegnek a kiegészítő erőforrások mennyiségének meghatározásához.

Közösségi Felhő

- Nem összetévesztendő a közösségi hálózattal.
- Valójában logikailag a privát és publikus felhő között helyezkedik el. Lényege, hogy **több cég** (hasonló informatikai irányelvekkel) **összefog** és **saját privát felhőjük erőforrásait megosztja a többiekkel**.
- Így amennyiben nem ugyanabban az időpillanatban vannak a rendszerben többleterőforrás igények, úgy **egy másik cég privát felhőjéből „kölcsönözhetünk” erőforrásokat**. A biztonság magasabb fokú, mint egy átlagos publikus felhőben, mégis rendelkezik annak a skálázhatóságával. Feltéve természetesen, ha a privát felhők teljesítménye közel azonos.