# 19. Tétel

Milyen különbségek vannak a Linux kernel modulok és alkalmazások között? (felépítés, fordítás, hibakezelés, további megkötések)

## Linux kernel

* monolitikus modul
  + egyben minden funkció
  + adatstruktúra
* más megoldás:
  + mikrokernel: minden funkcionális elemet külön egységben tartalmaz + jól meghatározott kommunikáció
* monolitikus modul következménye: új komponens, funkció hozzáadása bonyolultabb
  + korábban újrakonfigurálást, és újrafordítást igényel
  + még most is az nVidia CUDA képes GPU-inál
* 1.2-es verzió óta betölthető modulok suppportja
  + dinamikus hozzáadás, és elvétel a modultól
  + eszközvezérlők

## Fordítás

* saját makefile-ok
* c fordítás

## Betöltés – eltávolítás

* modul betöltése: insmod hellomodule.ko
  + hozzálinkeli a kernelhez
  + inicializálja
* modul eltávolítása: lsmod hellomodule
  + meghívódik a modul tisztító függvény
  + unlinkeli a programot
  + törli a memóriából
* betöltött modulok listázása: lsmod
* modprobe:
  + modul 🡪 /lib/modules/<kernelverzió>/
  + depmod – modulok függőségi listájának előállítása
  + modprobe – betölti a megnevezett modult és a függőségeit is

## Felhasználói mód – kernel mód

* programok: felhasználói módban
  + megvédi a rendszert az illegális hozzáféréseket az erőforrásokhoz
* modulok: kernel módban
  + bármi megengedett 🡪 kernel modulokat csak root tölthet be
* memóriahasználat
* eszközökhöz való hozzáférés

## Különbségek alkalmazások és modulok között

|  |  |  |
| --- | --- | --- |
| **modulok** | **szempont** | **alkalmazások** |
| csak C | nyelv | bármi, amire van fordítód az adott Linux distrora |
| két belépési pont:   1. betöltéskor inicializálódik csak, majd várja, hogy használja valaki 2. tisztogató: eltávolítás előtt aktivizálódik | belépési pontok, és végrehajtódás | main() függvénynél beugrik, és dolgozik, amíg a kilépési feltétel nem adott |
| csak a kernellel linkelődik 🡪 csak olyan fv. használható, ami már le van implementálva a kernelben 🡪 nem használunk fejlesztői könyvárakat 🡪 nincsenek headerek a hagyományos értelemben. Ami használható az a linux és az asm alkönyvtárakban van | headerek használata | a program sokszor nem tartalmazza minden adott fv. definícióját 🡪 libary-k, headerek  fordítási fázisban a linkelés során oldódnak meg ezek a linkek |
| a kernelnek mindig a memóriában kell lennie, nemkerülhet ki lapcsere során a merevlemezre 🡪 spórolni kell a hellyel | memóriakezelés | alkalmazástól függően fizikai memóriában, swap-ban, vagy a merevlemezis használható |
| fixpontos (tizedes pont előre meghatározott fix helyen + előjelbit) | számábrázolás | ami tetszik  (lebegőpontos: értékes számjegy + kitevő) |
| komoly problémák, akár rendszerösszeomlás  bonyolult hibakezelés | hibakezelés, segmentation fault (valami olyasmihez való hozzáférés, ami nem a tied, pl. túlindexelés. Már valaki használja azt a memóriaterületet) | nem veszélyes, kiszűrhető  elterjedt hibakezelő megoldások |