

## A verziókövetés → Git

### • Repository

- Adatházis, lokálisan vagy egy távoli serveren → Fájlok, kódok, stb...

### • Server/origin

- Távoli server, ami tárolja a repository-t

### • Branch

- Létrehozhatunk "különböző" brancheket, például ahova az új feature-öt töltjük fel
- Alapértelmezett fő branch → Master, main

### • Commit

- Üzenetet csatolunk a módosított kódhoz

### • Push

- Fájlokat töltünk fel a távoli serverre.

### • Merge

- Kódok összekendelése, ahol konfliktus is létrejuthat
- Konfliktusokat kezelhetjük automatikusan vagy manuálisan

### • Fetch

- Frissítések letöltése a távoli repository-ból, merge nélkül

### • Pull request

- Frissítések letöltése és merge-je

### • Tag

- Címke, például verziót tudunk jelölni

### • Revision number

- Egy hash függvény segítségével ellátott fájl, ez alapján tudunk visszamenni korábbi verzióra

## • Head

- Legutolsó revision number hivatkozó kifejezése

## • Kliensek

- Visual Studio / Code
- GitKraken
- Sourcetree

## • Checkout

- Branchek közötti váltás

## • .gitignore

- Fájl, amelyet nem szeretnénk feltölteni.
- .dot fájl

## • Fájlrendszeri modellezés

- Gitflow → Branch ágak létrehozása specifikusan

1. develop ág → fejlesztések
2. feature ág → új funkciók
3. release ág → új verzió, ami végleges
4. hotfix ág → javítások, patch-ek

## • Trunk-based

- Minden fejlesztés lokálisan fejleszt, majd legalább naponta 1x merge-lik
- Merge-ek attól függően meg kell történnie, hogy kész van-e vagy sem
- Kevesebb ág, kevesebb merge conflict
- CI/CD
- Kisebb teamben

## • Linux-kernel

- Fejlesztő közös fejlesztési erőfeszítéseinek eredménye
- Kis létszámú maintainerokból álló team, akik felügyelik a hozzájárulásokat és biztosítják, hogy azok összhangban legyenek a projekt általános céljaival.
- Folyamata:
  1. Ötlet/idea
  2. Fejlesztés/development → teszteléssel
  3. Benyújtás/submission → fejlesztő benyújtja a kódot, hogy a maintainer felülvizsgálja
  4. Értékelés/review → Maintainer ad egy visszajelzést, hogy min kellene még változtatni
  5. Tesztelés/testing → Maintainer és más fejlesztők tesztelik a kódot, hogy stabil-e
  6. Integráció/integration → Ha átment a teszten és a maintainer elfogadta, akkor beintegrálják a kernel kódbázisába.
  7. Release → új verzió nyilvánosságra kerül és ezzel a ciklus kezdődik elölről