A verziókövetés alapjai téma

Programozóként szinte minden esetben verziókövető rendszereket használunk a munkánk mentésére.

Ha szoktunk számítógépen játszani, akkor minden bizonnyal gyakran használjuk a játékok mentés funkcióját. Ha egy nehezebb részhez érünk, vagy egyszerűen csak ki szeretnénk valamit próbálni, elmentjük az aktuális állást. Ha nem a tervek szerint haladtak a dolgok, akkor pedig visszatöltjük azt.

A verziókövető rendszerek is hasonló elven működnek, csak itt nem játékállást, hanem fájlokat, fájlváltozásokat mentünk el. A verziókövetők az alábbiakban támogatják a munkánkat:

* Visszaállhatunk egy fájlt egy régebbi verzióra
* Az egész projektünket visszaállíthatjuk egy régebbi verzióra
* Összehasonlíthatunk fájlokat időben
* Megnézhetjük, ki, hogyan és miért módosított egy fájlt
* Ezáltal megtalálhatjuk a hibák okozóit

Tárolók

A verziókövető rendszerek tárolókat (repository vagy repo) használnak a fájlok tárolására. Ezeket mi tekinthetjük az egyszerűség kedvéért mappáknak.

Közös munka

A közös munka egyik alapvető eleme a verziókövetők használata. Amikor végzünk egy adott feladattal, feltöltjük azt egy közös verziókövető **szerverre**. Ezáltal mindenki számára elérhetőve válik a munkánk, és fordítva, a többiek munkáját is tudjuk használni.

Íme a tipikus munkafolyamat, zárójelben az idevágó **git** paranccsal:

1. Ellenőrizzük, hogy vannak-e változások a verziókövető szerveren (**fetch**).
2. Amennyiben vannak, letöltjük őket (**pull**).
3. Helyi környezetünkben integráljuk a változásokat a saját fájlainkkal (**merge**).
4. Elvégezzük a munkánkat a saját gépünkön (**commit**).
5. Ismét ellenőrizzük a szerver változásait (**fetch**).
6. Ismét letöltjük a változásokat, ha vannak (**pull**).
7. Ismét integrálunk (**merge**).
8. Kipróbáljuk a saját gépünkön, hogy működik-e minden.
9. Ha igen, feltöltjük a munkánkat (**push**).

További olvasnivalók:

<https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>