

Mindennapi informatikai alkalmazások – Git, verziókezelés

Csorba Kristóf (kristof@aut.bme.hu)

2018. ősz

“Tudunk gyorsan demózni egyet?”

“Persze, nem gond. Most kicsit darabokban van a firmware, de mindjárt feltöltök egy jót...”

Hajjaj...

20160905.zip

firmware2016szept06.zip

f_szept8.zip

demo_szept06.zip

demo0_javitva.zip

demo1_javitva_final.zip

demo2_felkesz.zip

demo2_ok.zip

demo2_final.zip

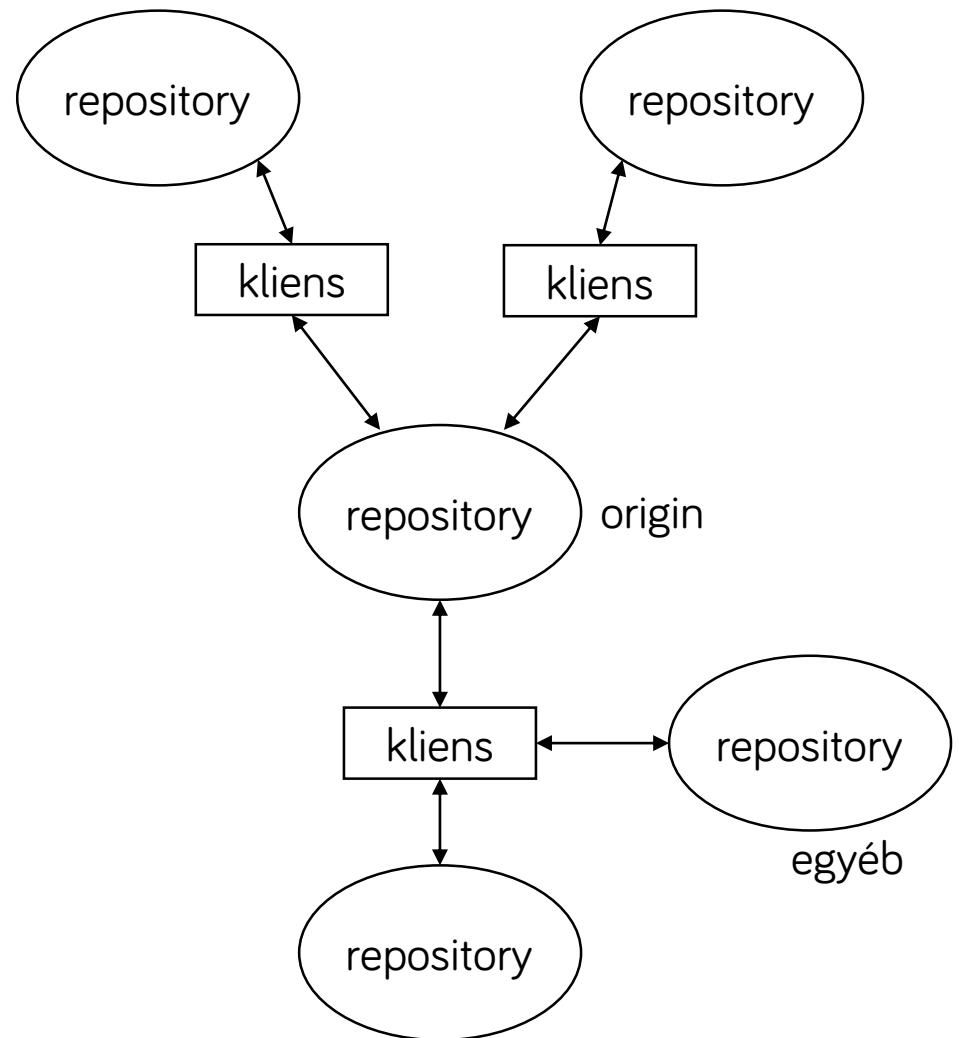
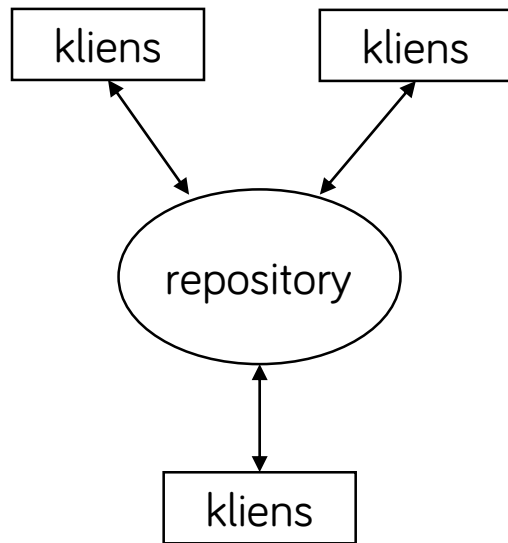
demo2_final_megsem.zip

Mi a verziókezelés?

- Egyéni vagy csapatmunka követése
 - “zip – email – dropbox” helyett...
 - Munkakönyvtár elmentett és megosztott állapotai, vissza lehet állítani
 - > legutolsó jó verzió
 - > “a márciusi demó verziója”
 - > ideiglenesen félrerakott verzió
 - Párhuzamos munka összefésülése
- Git, TFS, SVN, CVS, Perforce

Példa: cv4sensorhub repository





Centralizált vagy decentralizált

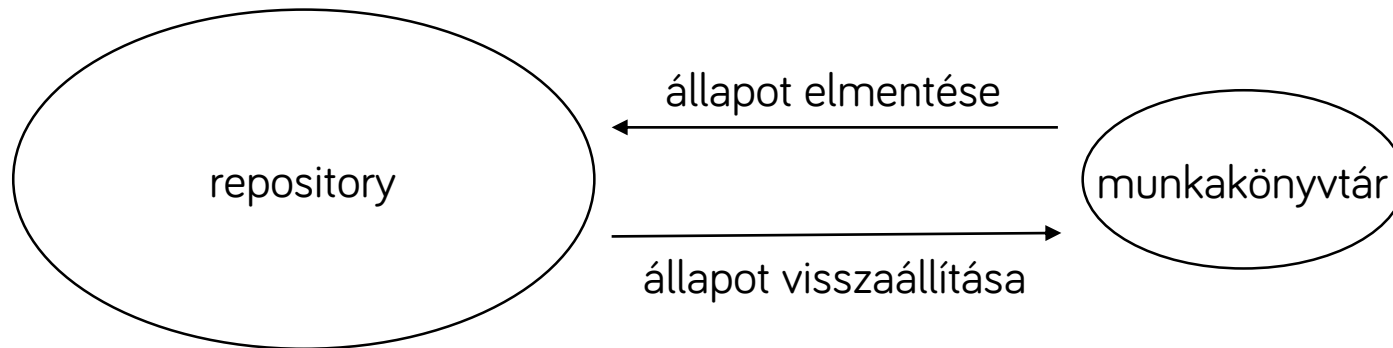


Mit verziókezelünk?

IGEN (ami a fordításhoz kell)	NEM (ami generálódik vagy felhasználó specifikus)
<p>forráskód fájlok (LaTeX diplomaterv) projekt fájlok (pro, pri, sln...) erőforrások: ico, svg, png... .gitignore</p>	<p>fordítás eredményei *.obj, *.o, *.exe (ami generálódik) Felhasználói beállítások: *.user, *.filter</p>

Repository és munkakönyvtár

 ..[..]	<DIR>
 [.git]	<DIR>
 [PanicButton]	<DIR>
 .gitignore	90

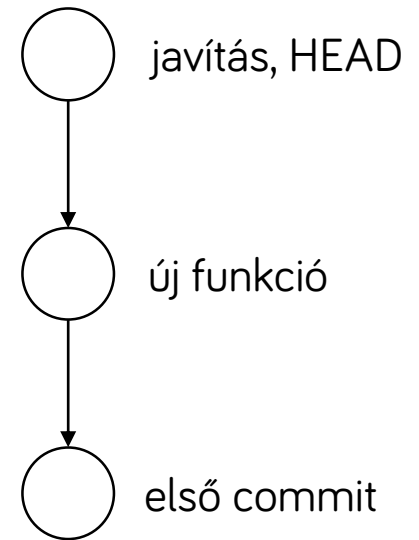


Commitok és gráfjuk

```
diff --git a/PanicButton/state.c b/PanicButton/state.c
index 695d993..413ad58 100644
--- a/PanicButton/state.c
+++ b/PanicButton/state.c
@@ -1,7 +1,7 @@
#include "state.h"

const char *stateStrings[] =
-{"Init", "Normal", "SendSick" };
+{"Init", "Normal", "SendSick", "SendPanic" };

const char *state2text(State currentState)
{
```



history: korábbi commitok sorozata

Hogyan lesz nekem ilyen?

- Git (win: saját shell, parancssoros), GitExtensions, pl. GitHub (student license)

<https://git-scm.com/>

<http://code.google.com/p/gitextensions/>

<https://github.com/>

GitExt



Git ↔ GitHub.com

```
$ git init
Initialized empty Git repository in c:/temp/.git/
```

vagy clone...

Snippetek

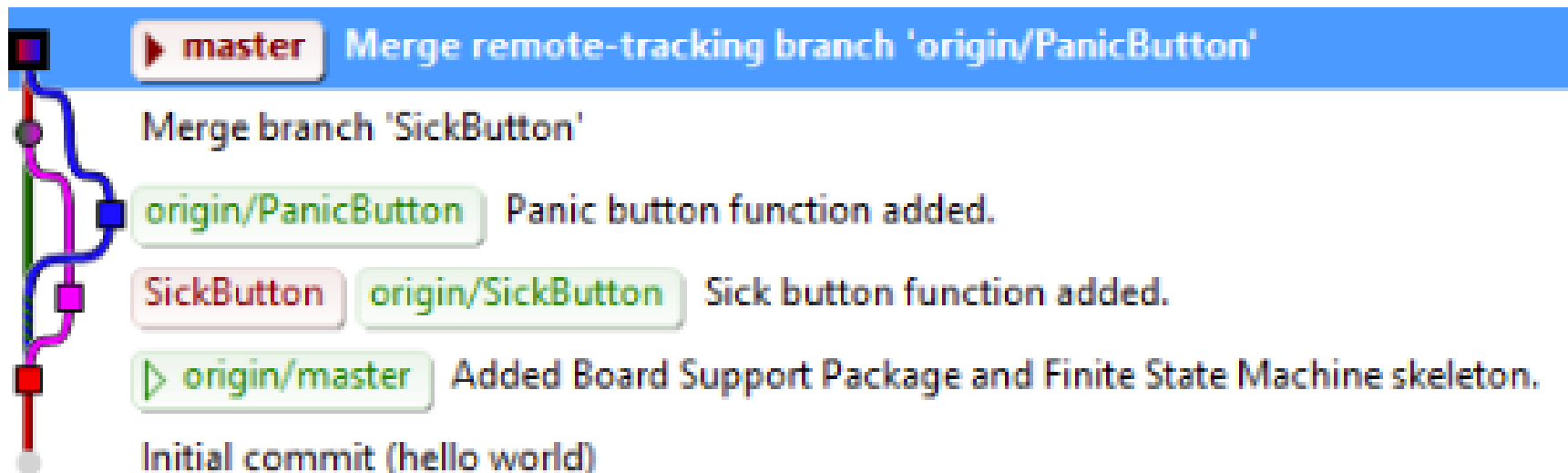
- <http://bmeaut.github.io/snippets/#git>
- Pro.Git könyv
- <http://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>
- Kulcsgenerálás, GitExtensions, github első lépések
<https://www.youtube.com/watch?v=OntibzA4038>
- Demó (színekkel):
<https://youtu.be/OOHnNx1M2aY>

DEMÓ...

A commit gráf, alapsműveletek

A commit gráf (fa, merge commit)

- Commitok és szülő(k)
- Branchek (local, remote)
- Merge commit



Műveletek: Staging, commit (push)

The screenshot shows a Git GUI window titled "Commit to master (E:\Oktatas\AlkFejl\GitPelda\Andezit\)". The window is divided into several sections:

- Working dir changes:** A list of files to be committed, including "PanicButton/main.c.orig", "PanicButton/state.c.orig", and "PanicButton/state.h.orig".
- Diff view:** A diff of the file "state.c" between two versions. The diff shows a change in the "stateStrings" array, adding "SendPanic" to the list of states. The diff is highlighted with green and red colors.
- Staged files:** A list of files that have been staged for commit, including "PanicButton/main.c", "PanicButton/state.c", and "PanicButton/state.h".
- Commit actions:** A set of buttons for committing the changes, including "Commit", "Commit & push", "Reset changes", and "Amend Commit".
- Commit message:** A text area for entering a commit message, currently containing "Merge remote-tracking branch 'origin/PanicButton'".
- Conflicts:** A list of files that have conflicts, including "PanicButton/main.c", "PanicButton/state.c", and "PanicButton/state.h".

```
diff --git a/PanicButton/state.c b/PanicButton/state.c
index 695d993..413ad58 100644
--- a/PanicButton/state.c
+++ b/PanicButton/state.c
@@ -1,7 +1,7 @@
#include "state.h"

const char *stateStrings[] =
-{"Init", "Normal", "SendSick" };
+{"Init", "Normal", "SendSick", "SendPanic" };

const char *state2text(State currentState)
{
```

A commit message...

“töröltem a default.ini-t”

“áthelyeztem az InitSettings osztályt”

“már majdnem kész...”

“javítások”

“small fixes”

“Új kezdőlap”

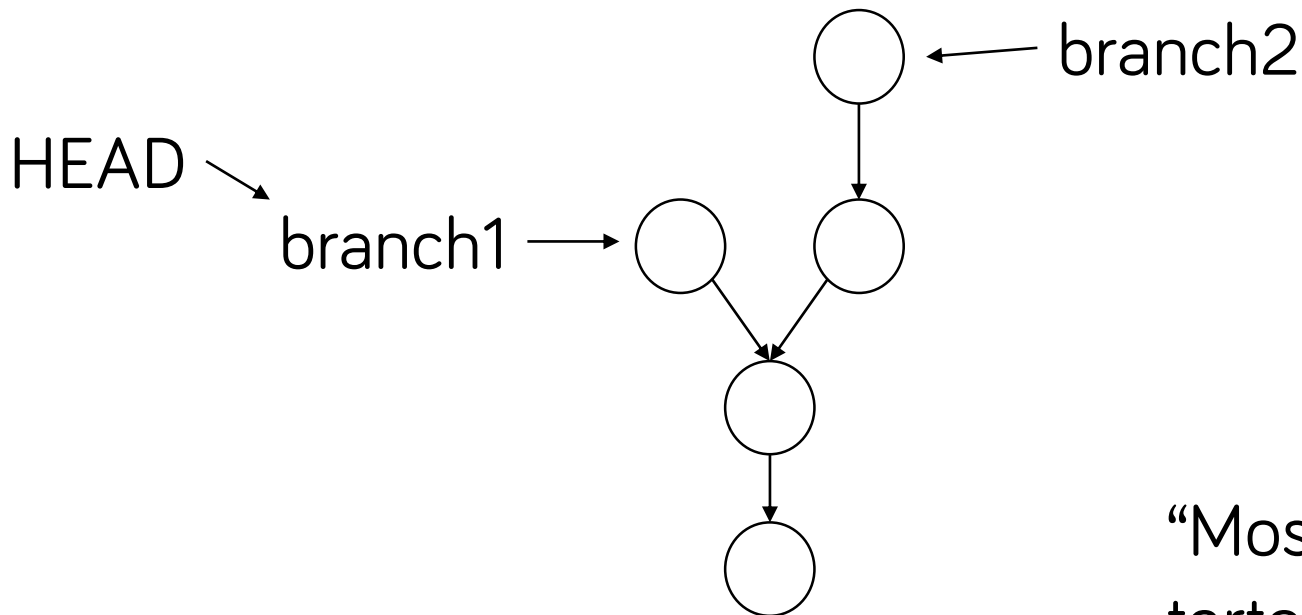
“ADC init hiba javítva”

“#1344 könyvtár jogosultság javítva”

Nem mit (az látszik), hanem miért!

Branchek, vagyis ágak...

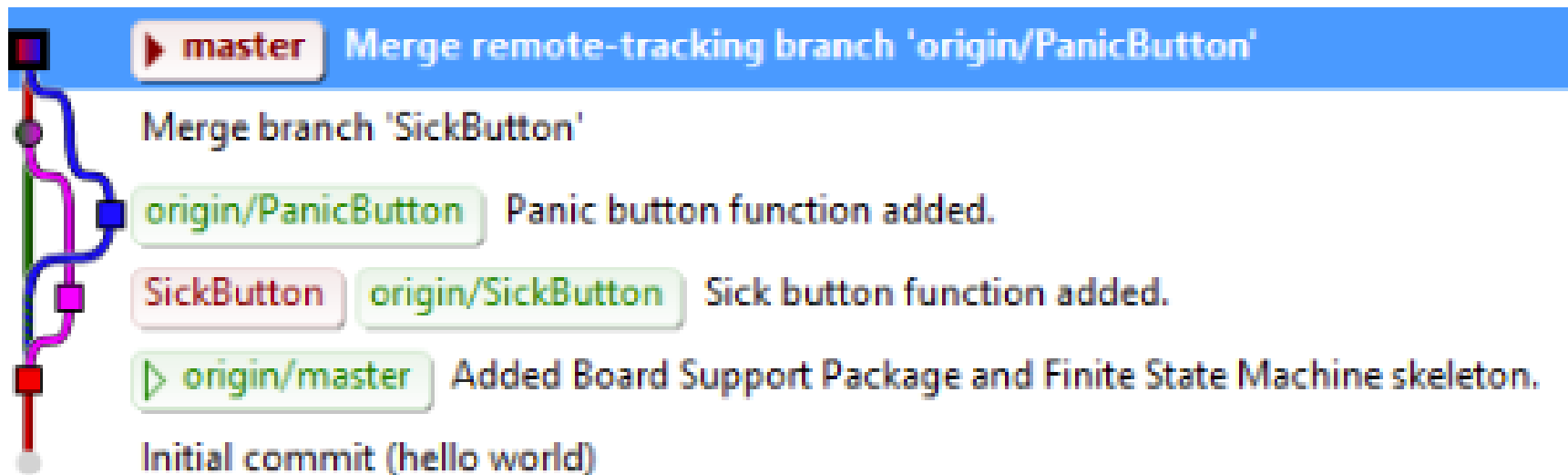
- A branch csak egy hivatkozás egy commit-ra.
- A tag is ilyen, csak nem mozog vele tovább
- HEAD referencia: amire commitolni fogok



“Most akkor hol is tartok/tartunk?”

Műveletek: merge

Merge commit vagy fast forward...



Műveletek: 3-way merge (KDiff3)

The screenshot displays the KDiff3 3-way merge interface. The window title is "state.h.BASE <-> state.h.LOCAL <-> state.h.REMOTE - KDiff3". The interface is divided into three main panes for the source files and one for the merged output.

Pane A (Base): `state.h.BASE`
Top line 1
Encoding: System
Line end style: DOS
Code:

```
/** Finite state machine state definitions
#pragma once
#ifndef STATE_H
#define STATE_H

typedef enum
{
    Init = 0,
    Normal
} State;

const char *state2text(State currentState)

#endif // STATE_H
```

Pane B (LOCAL): `state.h.LOCAL`
Top line 1
Encoding: System
Line end style: DOS
Code:

```
/** Finite state machine state definitions
#pragma once
#ifndef STATE_H
#define STATE_H

typedef enum
{
    Init = 0,
    Normal,
    SendSick
} State;

const char *state2text(State currentState)

#endif // STATE_H
```

Pane C (REMOTE): `state.h.REMOTE`
Top line 1
Encoding: System
Line end style: DOS
Code:

```
/** Finite state machine state definitions
#pragma once
#ifndef STATE_H
#define STATE_H

typedef enum
{
    Init = 0,
    Normal,
    SendPanic
} State;

const char *state2text(State currentState)

#endif // STATE_H
```

Output: `PanicButton\state.h` [Modified]
Encoding for saving: Codec from C: System
Line end style: DOS (A, B, C)
Code:

```
/** Finite state machine state definitions. */
#pragma once
#ifndef STATE_H
#define STATE_H

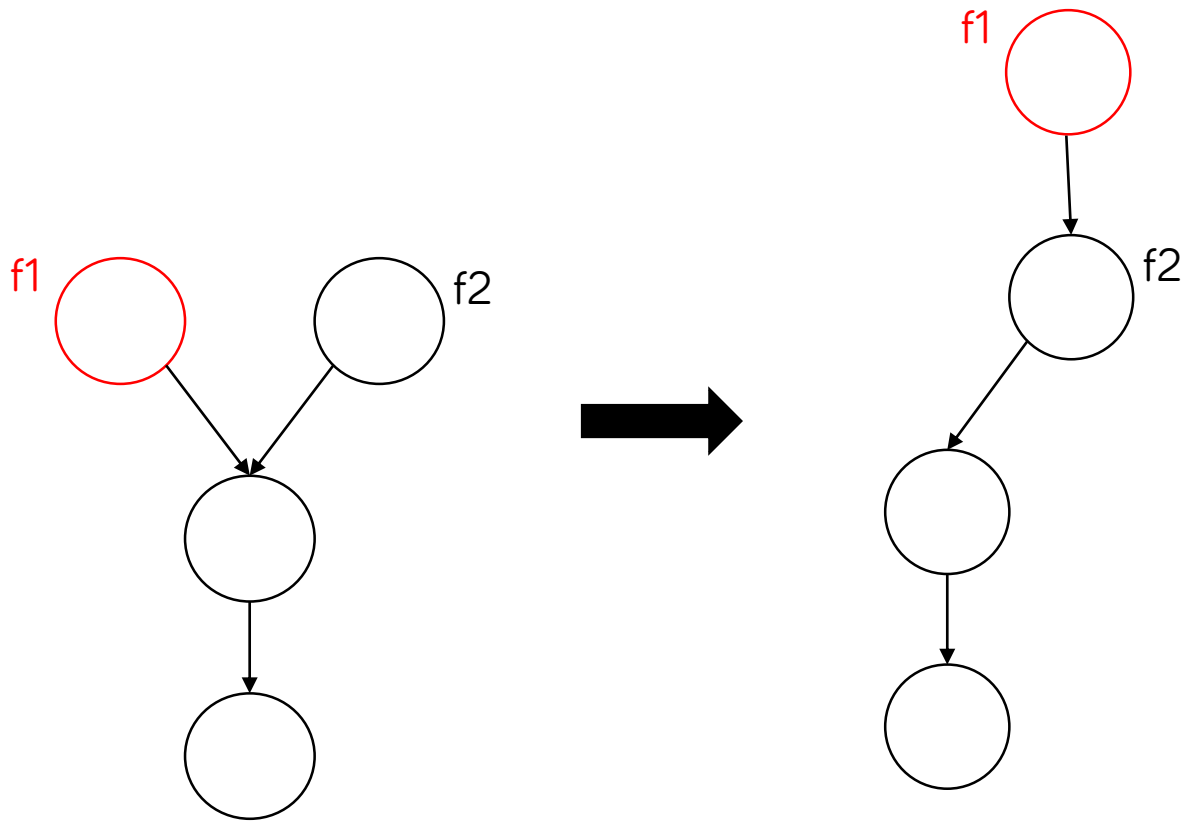
typedef enum
{
    Init = 0,
    Normal,
    SendSick,
    SendPanic
} State;

const char *state2text(State currentState);

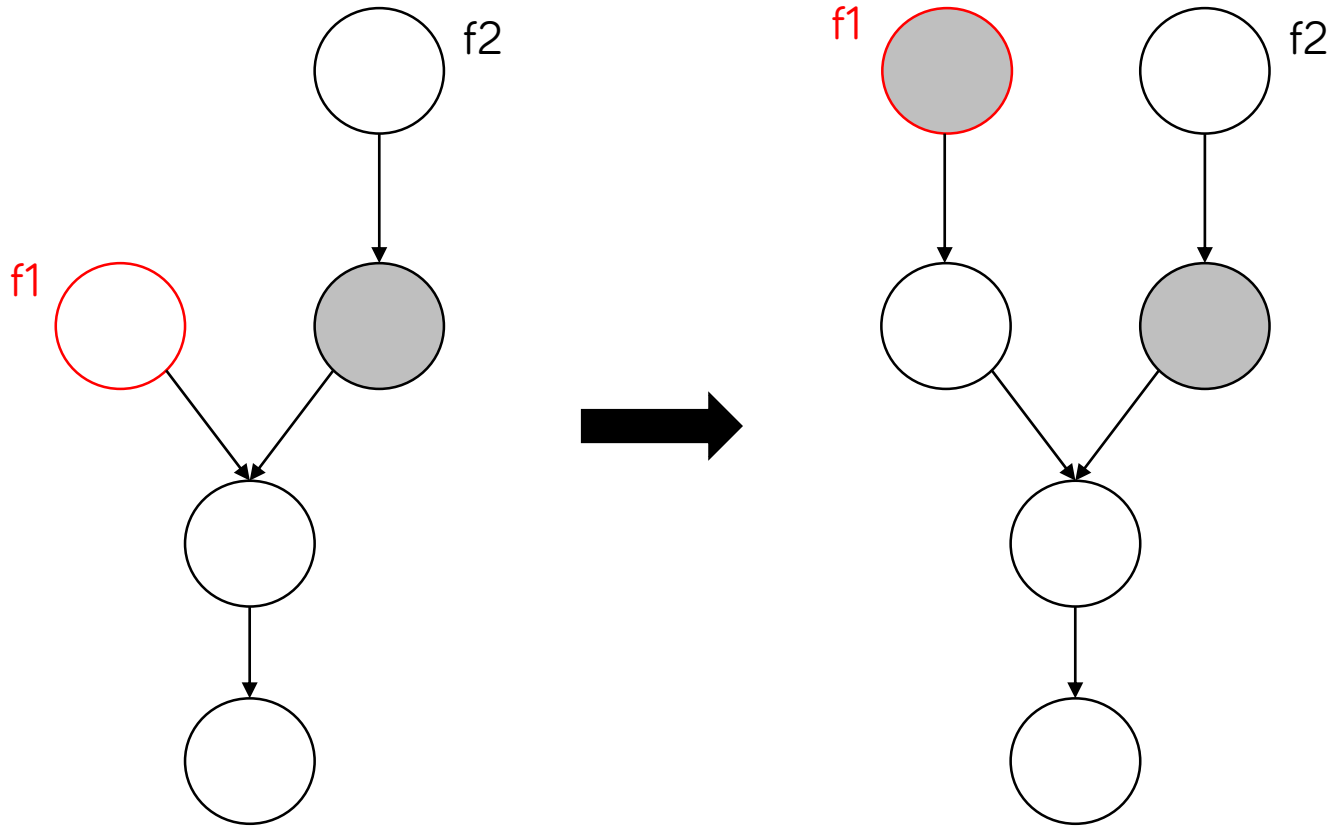
#endif // STATE_H
```

The merged output shows the union of changes from both LOCAL and REMOTE. The LOCAL version's `SendSick` and the REMOTE version's `SendPanic` are both included in the merged output. The output pane also features a vertical column on the left with letters A, B, and C, indicating the source of each line.

Rebase



Cherry picking



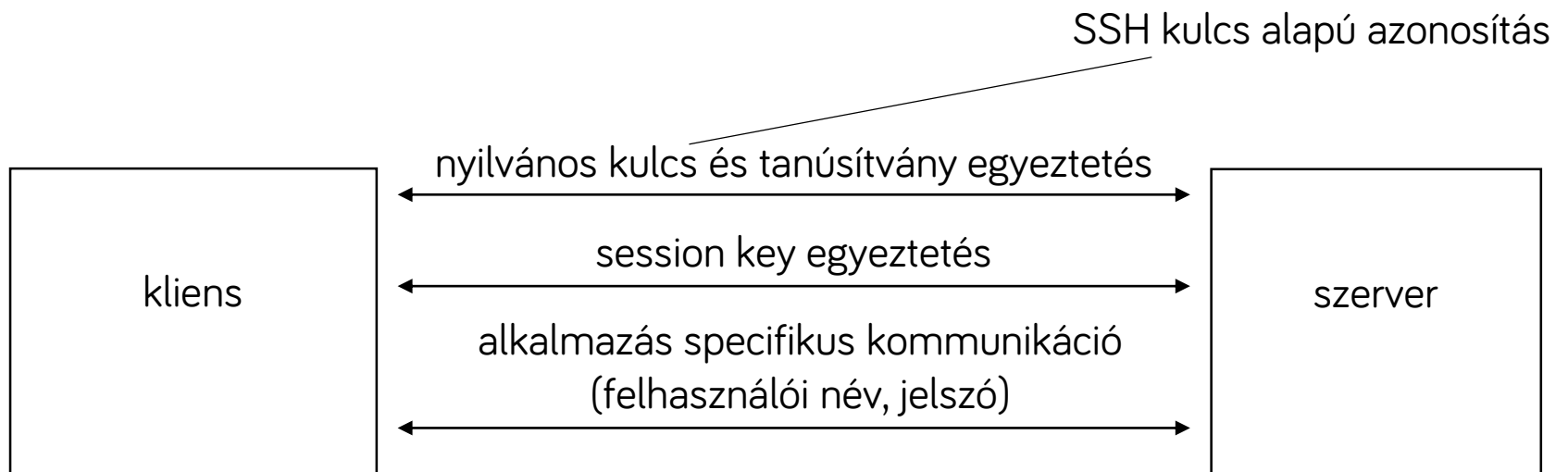
Mi van a .git könyvtárban?

↑ Name	Ext	Size
↑ .. [..]		<DIR>
[hooks]		<DIR>
[info]		<DIR>
[logs]		<DIR>
[modules]		<DIR>
[objects]		<DIR>
[refs]		<DIR>
COMMIT_EDITMSG		43
config		459
description		73
FETCH_HEAD		115
HEAD		23
index		11,192
ORIG_HEAD		41
packed-refs		94

Jogosultság kezelés, remote

Git autentikáció

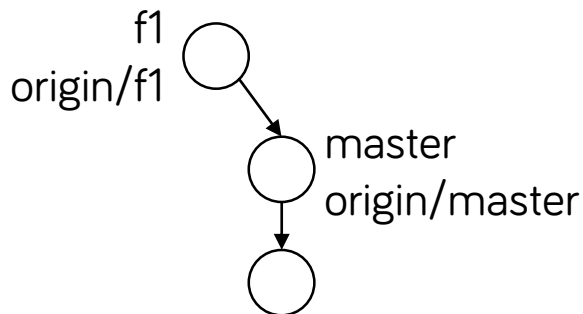
- SSH kulcs alapú
- felhasználói név és jelszó



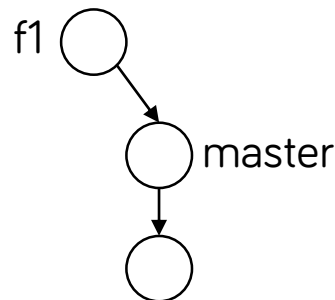
Remote

- Egy vagy több (origin)
- Sima könyvtár (bare)
- remote branch létrehozása és törlése (tracking branch)
- tag és pusholása

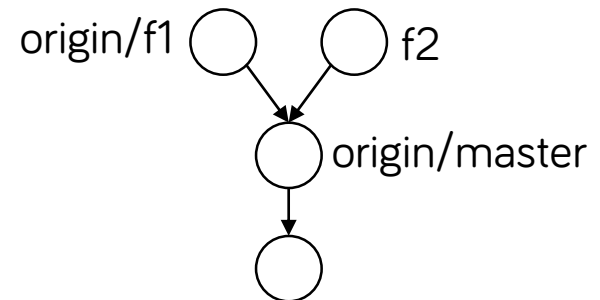
Andezit



origin



Bazalt

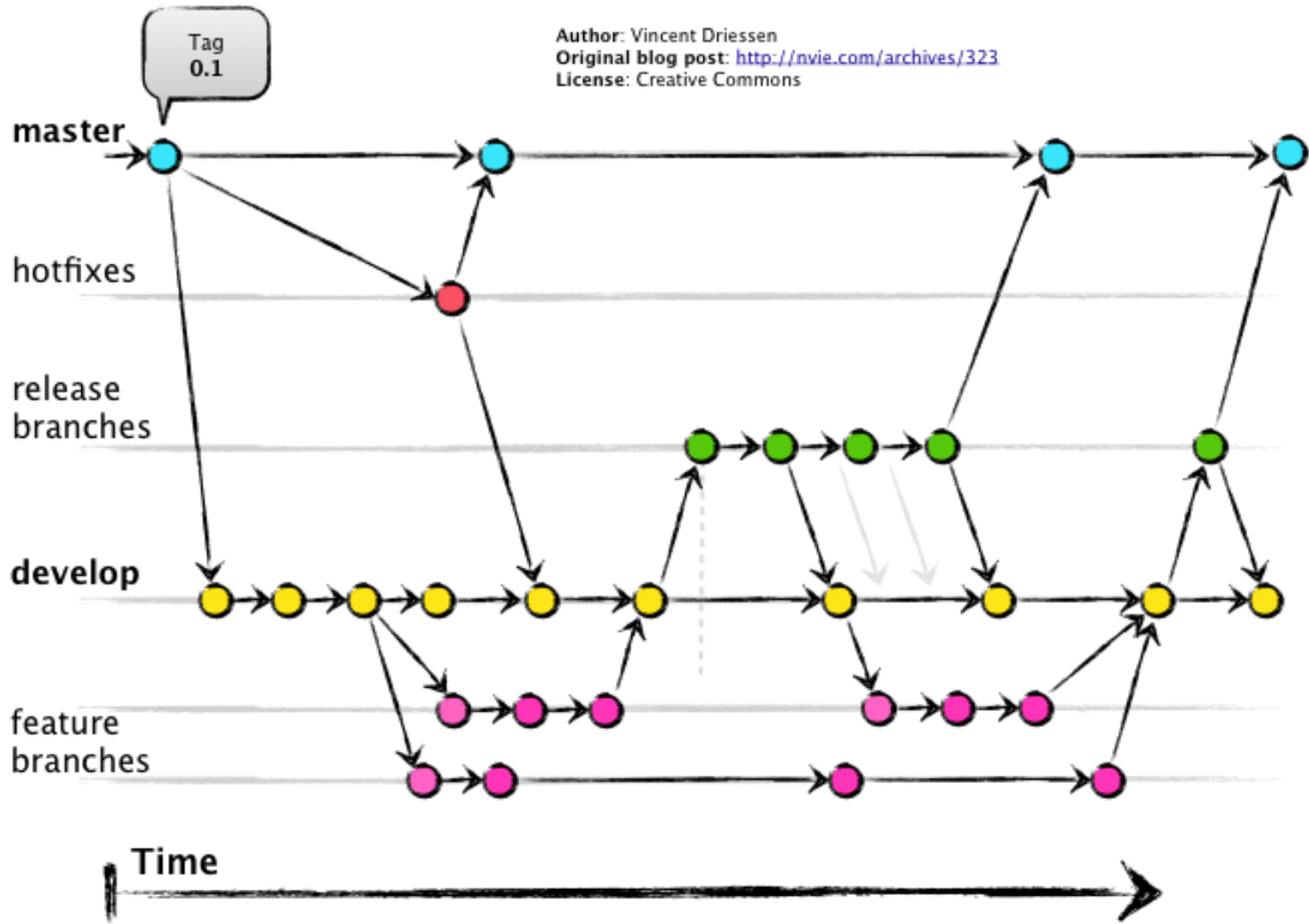


Reset

Reset type

- Soft: leave working dir and index untouched
- Mixed: leave working dir untouched, reset index
- Hard: reset working dir and index (discard ALL local changes, even uncommitted changes)

Git Flow



Feature flagek és master branch

- Minden feature branch a master-re mergelődik.
- Feature flag-ek kikapcsolják, ami még nincs készen.
 - > Feature flag logika, pl. publikus, alfa és beta teszterek, A-B testing

GitHub.com

- Public és private repository (student license)
- Git server
- issue tracker (assignee, labels, milestone)
- Wiki

- fork és pull request
- Markdown támogatás

Snippetek

- 0116_GitAuth
- 0128_GitHalado
- 0129_GiTuzvedelem
- 0137_GithubForkPullReq
- 0138_GitParancssor
- Git első fejlesztési lépések (merge, rebase)
<https://youtu.be/OOHnNx1M2aY>
- Github és ssh kulcs
<https://www.youtube.com/watch?v=OntibzA4038>

miniOHV: <http://goo.gl/forms/dS5aJN6zHT>

Félelmek, és amikor megbomlik a rend

Mikor tűnik el valami véglegesen?

Hova tűnt a tegnapi munkám???

Ezt most miért vontad vissza???

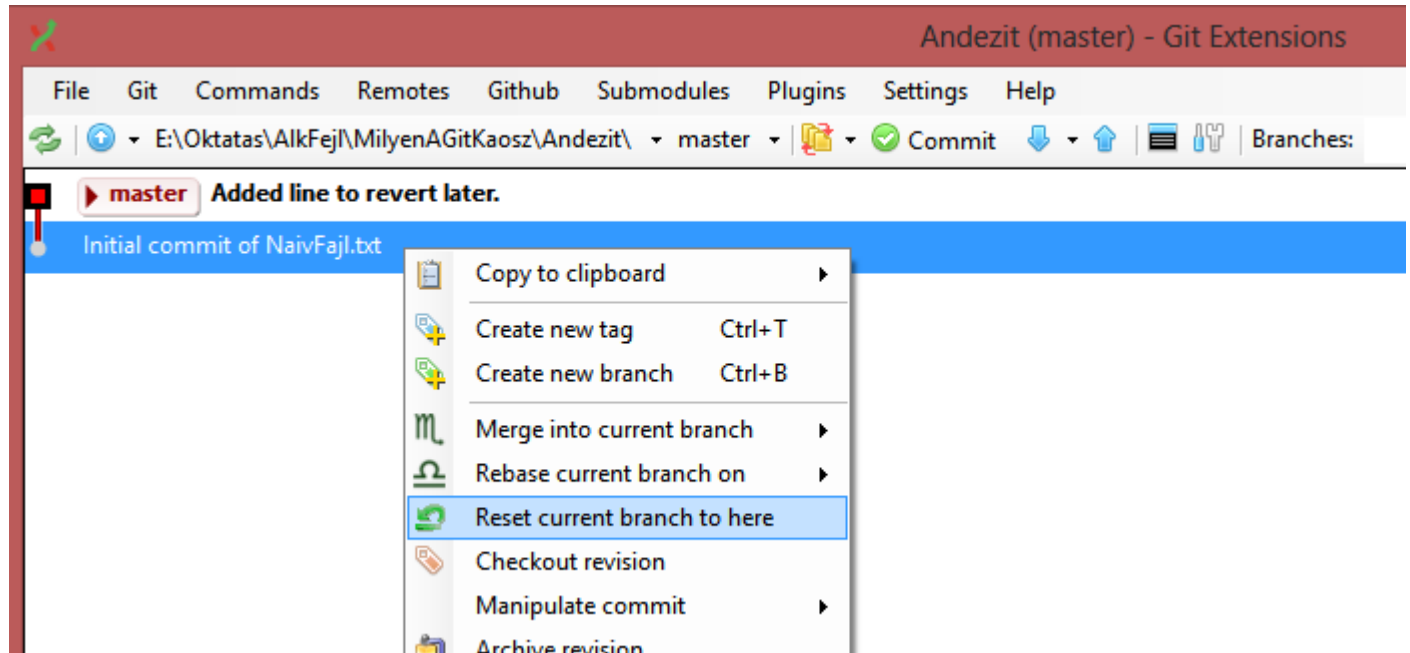
Waaaa, nem megy semmi a merge után!

Ne oda commitolj, te!

Na most mi vaaaaaan???

Lord Meldrum: “A világ megbolondult. Megyek, lefekszem!”

Elveszített commit

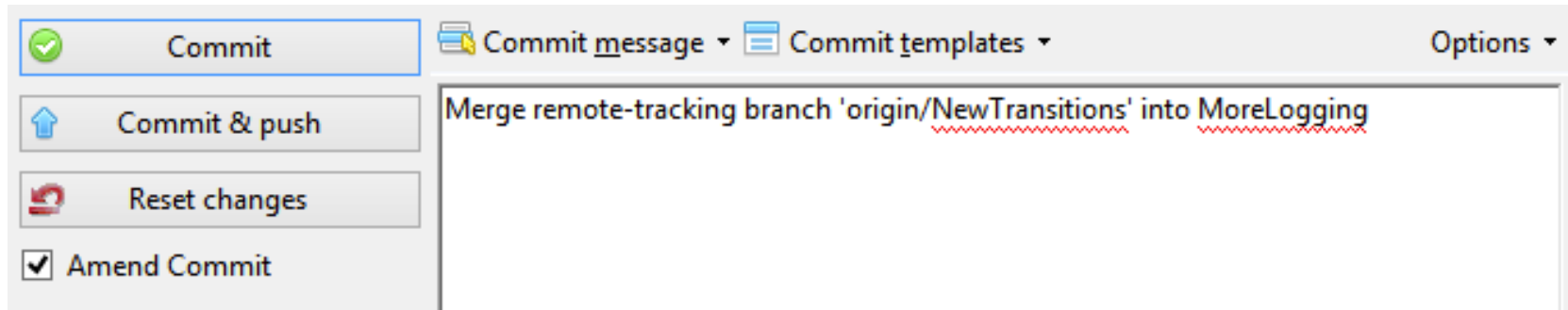


```
$ git reflog
af6290e HEAD@{0}: reset: moving to af6290ef74fdf4696d9da1319d7f33cdc3ab1b33
6c10c8e HEAD@{1}: commit: Added line to revert later.
af6290e HEAD@{2}: commit (initial): Initial commit of NaivFajl.txt
```

```
$ git reset 6c10c8e
```


Amend commit

Csak ha még nem push-oltunk!



git rebase -i HEAD~3

(Squashing)

```
pick 5b74f4a Uj nulladik sor es modositott masodik sor.
s 3c234c5 Negyedik sor
s 2c18aaf Otodik sor

# Rebase 5a1b6e1..2c18aaf onto 5a1b6e1
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```