Technological foundations of software development

Manage your source code

ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

Objectives of the session

Ensure you are familiar with source code management methodologies and tools, in particular the git software, the gitlab platform used at school, and the github platform.

Technological foundations of software development

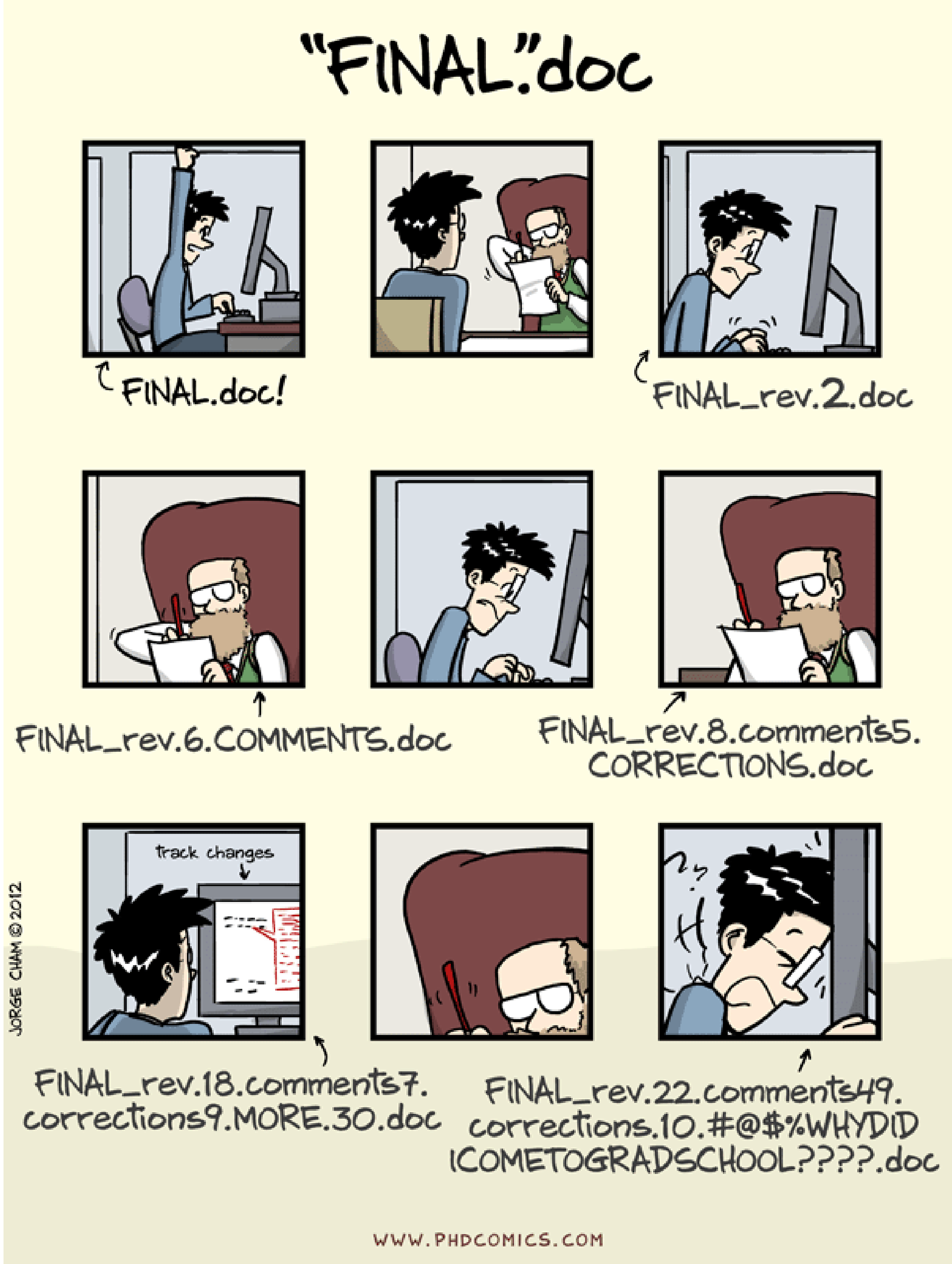
Manage your source code

Part 1: generalities

ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

Why track versions ?

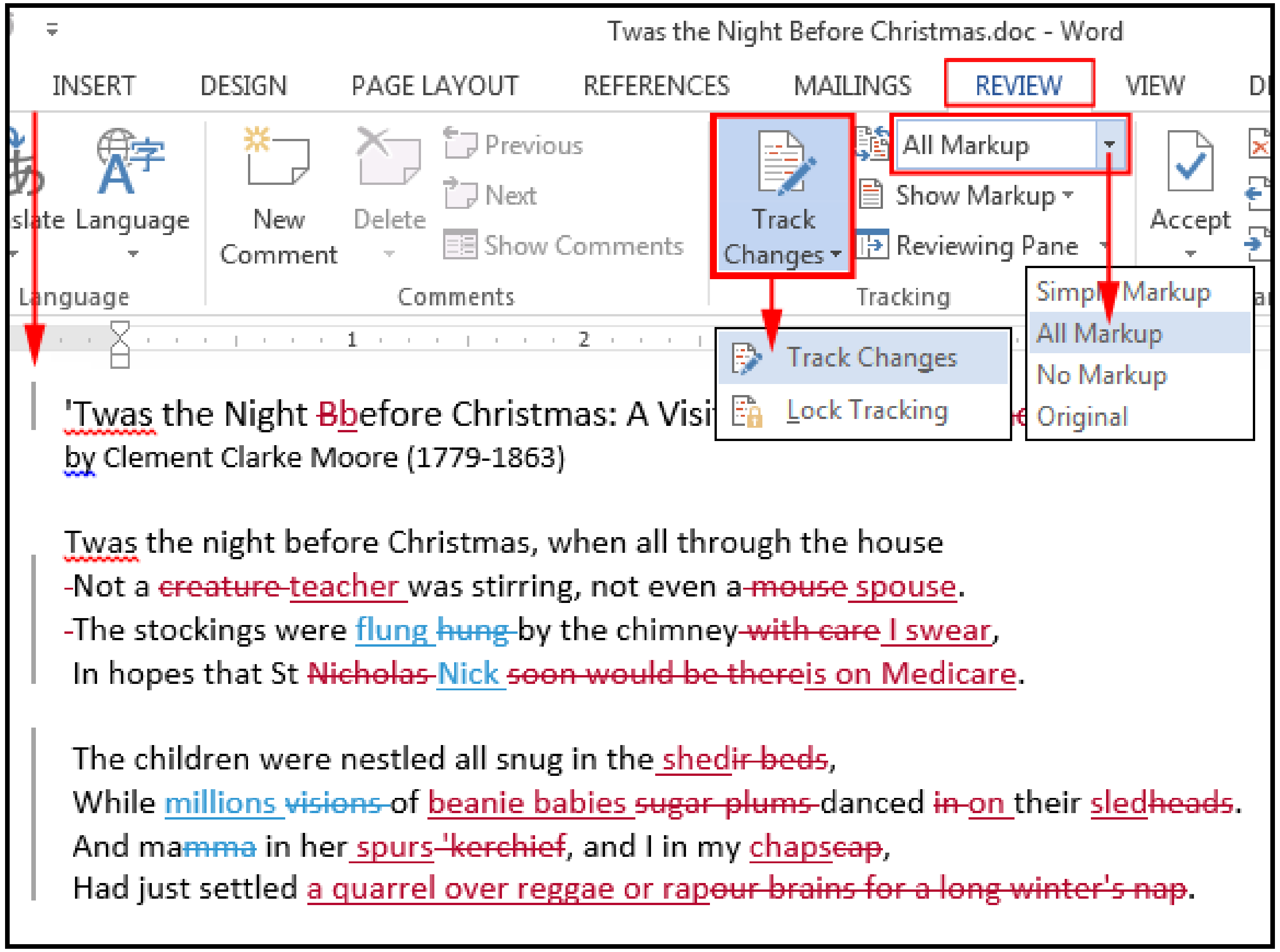
Example: many versions of the same file


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5. CORRECTIONS.doc
track changes
FINAL_rev.18.comments7. corrections9.MORE.30.doc
FINAL_rev.22.comments49. corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

http://phdcomics.com/comics/archive.php?comicid=1531

Track changes in a file

Example with Word



'Twas the Night ~~B~~before Christmas: A Visi

by Clement Clarke Moore (1779-1863)

Twas the night before Christmas, when all through the house
Not a ~~creature~~ teacher was stirring, not even a ~~mouse~~ spouse.
The stockings were flung ~~hung~~ by the chimney ~~with care~~ I swear,
In hopes that St ~~Nicholas~~ Nick ~~soon would be there~~is on Medicare.

The children were nestled all snug in the shed~~ir beds~~,
While millions ~~visions~~ of beanie babies ~~sugar plums~~ danced ~~in~~ on their sled~~heads~~.
And ma~~mma~~ in her spurs ~~'kerchief~~, and I in my chaps~~cap~~,
Had just settled a quarrel over reggae or rap~~our brains for a long winter's nap~~.

Why save versions ?

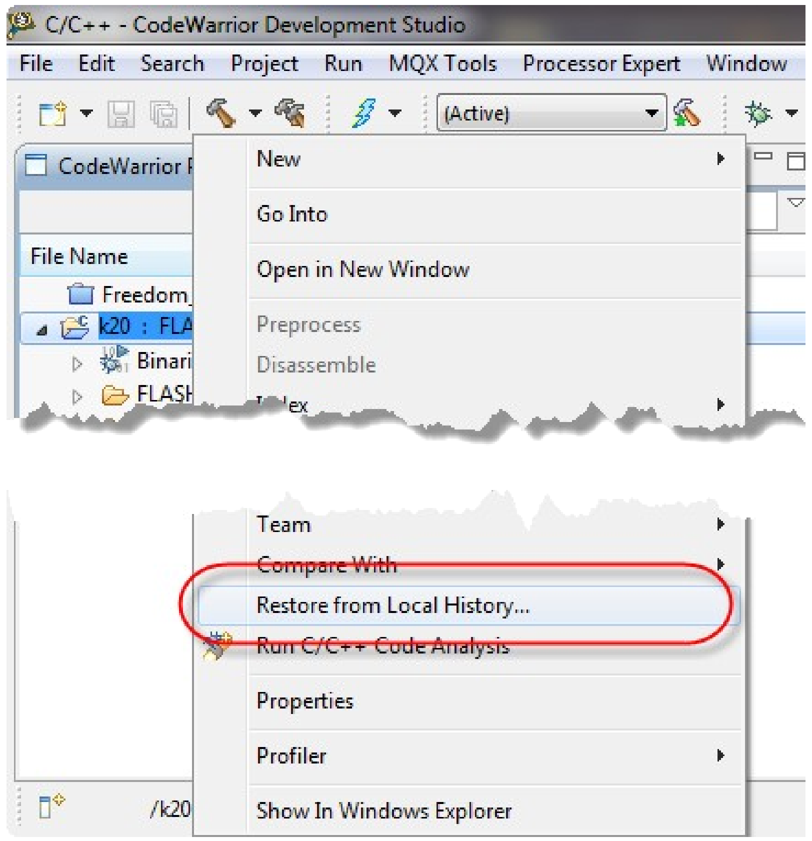
Example: to restore to a previous state

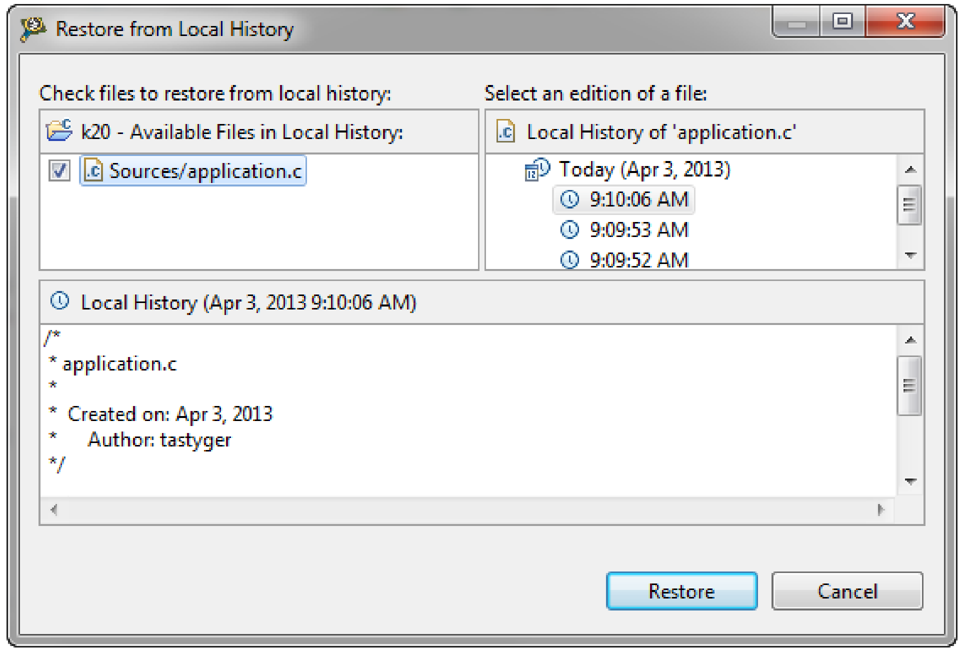
Return to Zero
WHAT'S THIS CUP OF USB DRIVES?
AH, THAT'S HOW I VERSION-CONTROL MY FILES...
TYPE
TYPE
TYPE
STAN'S DESK!
IT'S BRILLANT REALLY. JUST MAKE A NEW COPY ON A NEW ONE EVERYDAY.
WHAT HAPPENS WHEN YOUR CUP FILLS UP?
EASY! I DUMP THEM IN HERE!
USB
EEWeb.com

http://library.ucmerced.edu/node/66631

For code ?

Some IDEs have embedded solutions
For example with Eclipse



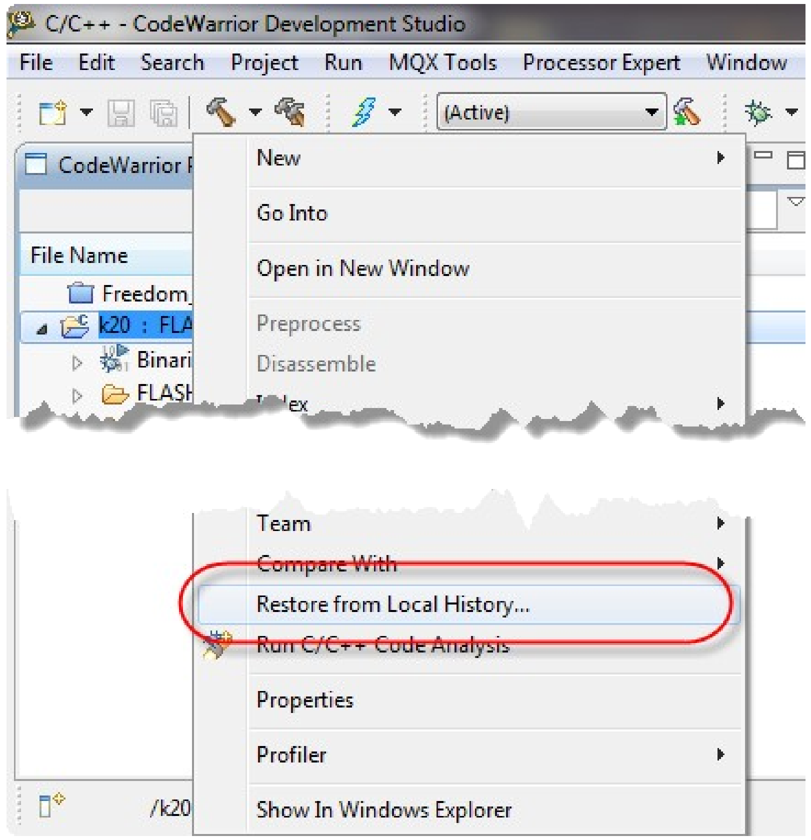

Restore from Local History
Check files to restore from local history:
Select an edition of a file:
k20 - Available Files in Local History:
Local History of 'application.c'
Sources/application.c
Today (Apr 3, 2013)
9:10:06 AM
9:09:53 AM
9:09:52 AM
Local History (Apr 3, 2013 9:10:06 AM)
Restore
Cancel

```
/*
 * application.c
 *
 *  Created on: Apr 3, 2013
 *      Author: tastyger
 */
```

https://mcuoneclipse.com/2013/04/03/restore-deleted-files-in-eclipse-with-local-history/

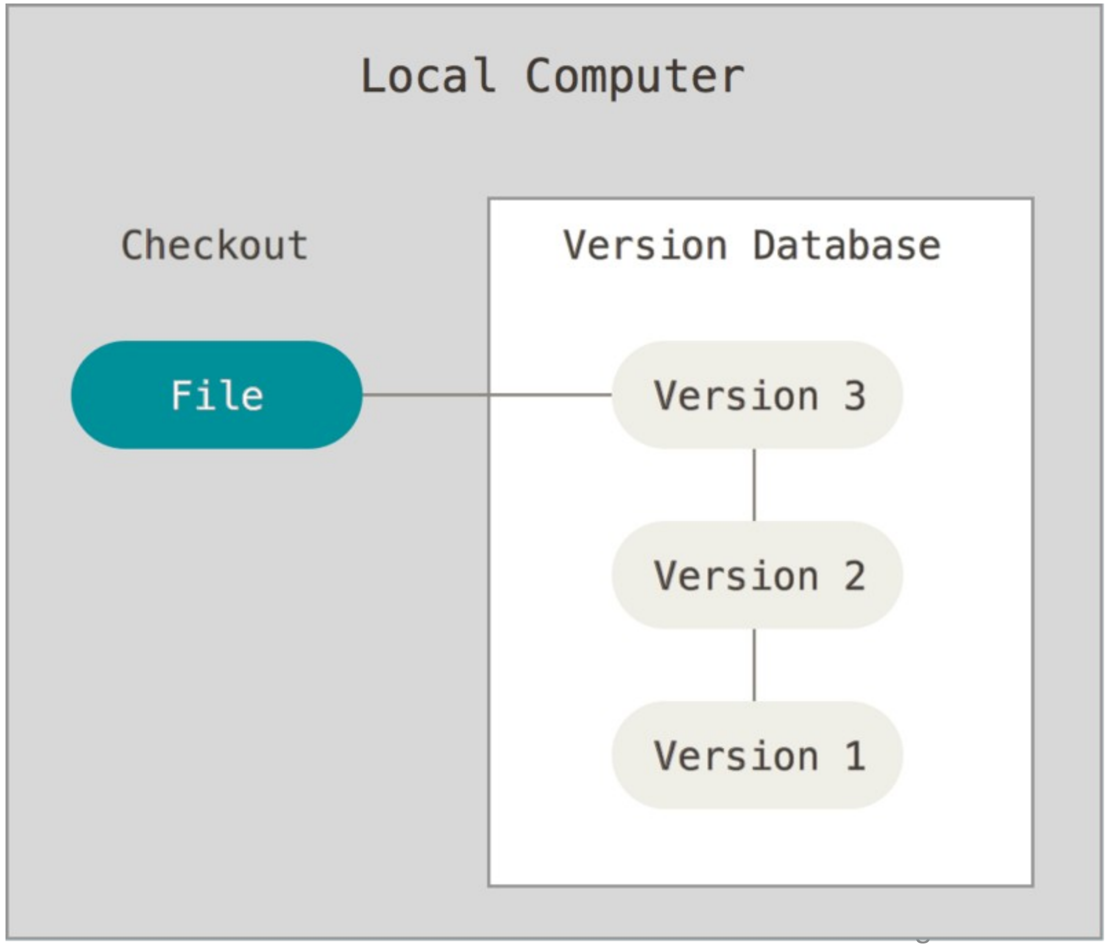
For code ?

Some IDEs have embedded solutions
For example with Eclipse



https://mcuoneclipse.com/2013/04/03/restore-deleted-files-in-eclipse-with-local-history/

Local version management



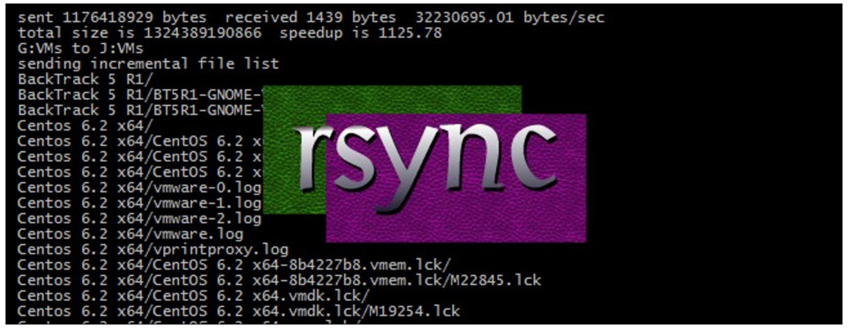
https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

File synchronization software

Many solutions exist

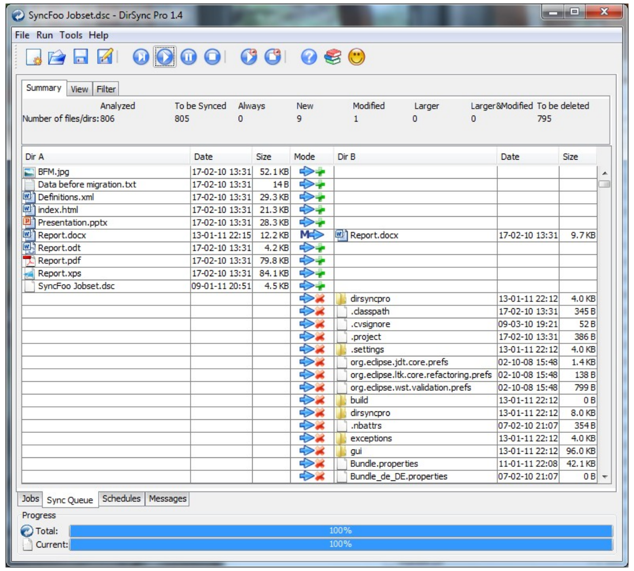
https://en.wikipedia.org/wiki/Comparison_of_file_synchronization_software

- commercial or open-source
- local, or with server, or with cloud
- personal or collaborative folders



```
sent 1176418929 bytes  received 1439 bytes  32230695.01 bytes/sec
total size is 1324389190866  speedup is 1125.78
G:VMs to J:VMs
sending incremental file list
BackTrack 5 R1/
BackTrack 5 R1/BT5R1-GNOME-
BackTrack 5 R1/BT5R1-GNOME-
Centos 6.2 x64/
Centos 6.2 x64/CentOS 6.2 x
Centos 6.2 x64/CentOS 6.2 x
Centos 6.2 x64/CentOS 6.2 x
Centos 6.2 x64/vmware-0.log
Centos 6.2 x64/vmware-1.log
Centos 6.2 x64/vmware-2.log
Centos 6.2 x64/vmware.log
Centos 6.2 x64/vprintproxy.log
Centos 6.2 x64/CentOS 6.2 x64-8b4227b8.vmem.lck/
Centos 6.2 x64/CentOS 6.2 x64-8b4227b8.vmem.lck/M22845.lck
Centos 6.2 x64/CentOS 6.2 x64.vmdk.lck/
Centos 6.2 x64/CentOS 6.2 x64.vmdk.lck/M19254.lck
```

rsync, 1996

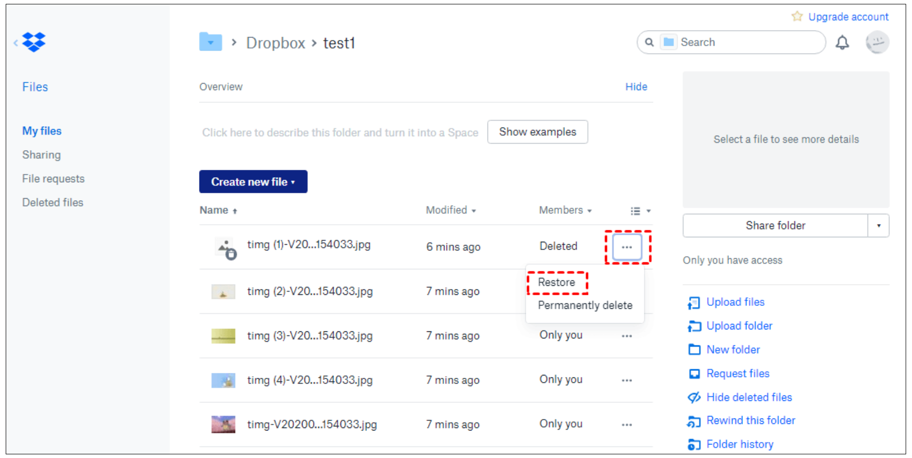


	Analyzed	To be Synced	Always	New	Modified	Larger	Larger&Modified	To be deleted
Number of files/dirs:	806	805	0	9	1	0	0	795

Dir A	Date	Size	Mode	Dir B	Date	Size
BFM.jpg	17-02-10 13:31	52.1 KB	→+			
Data before migration.txt	17-02-10 13:31	14 B	→+			
Definitions.xml	17-02-10 13:31	29.3 KB	→+			
index.html	17-02-10 13:31	21.3 KB	→+			
Presentation.pptx	17-02-10 13:31	28.3 KB	→+			
Report.docx	13-01-11 22:15	12.2 KB	M→	Report.docx	17-02-10 13:31	9.7 KB
Report.odt	17-02-10 13:31	4.2 KB	→+			
Report.pdf	17-02-10 13:31	79.8 KB	→+			
Report.xps	17-02-10 13:31	84.1 KB	→+			
SyncFoo Jobset.dsc	09-01-11 20:51	4.5 KB	→+			
			→✗	dirsyncpro	13-01-11 22:12	4.0 KB
			→✗	.classpath	17-02-10 13:31	345 B
			→✗	.cvsignore	09-03-10 19:21	52 B
			→✗	.project	17-02-10 13:31	386 B
			→✗	.settings	13-01-11 22:12	4.0 KB
			→✗	org.eclipse.jdt.core.prefs	02-10-08 15:48	1.4 KB
			→✗	org.eclipse.ltk.core.refactoring.prefs	02-10-08 15:48	138 B
			→✗	org.eclipse.wst.validation.prefs	02-10-08 15:48	799 B
			→✗	build	13-01-11 22:12	0 B
			→✗	dirsyncpro	13-01-11 22:12	8.0 KB
			→✗	.nbattrs	07-02-10 21:07	354 B
			→✗	exceptions	13-01-11 22:12	4.0 KB
			→✗	gui	13-01-11 22:12	96.0 KB
			→✗	Bundle.properties	11-01-11 22:08	42.1 KB
			→✗	Bundle_de_DE.properties	07-02-10 21:07	0 B

Total: 100%
Current: 100%

dirsyncpro, 2004



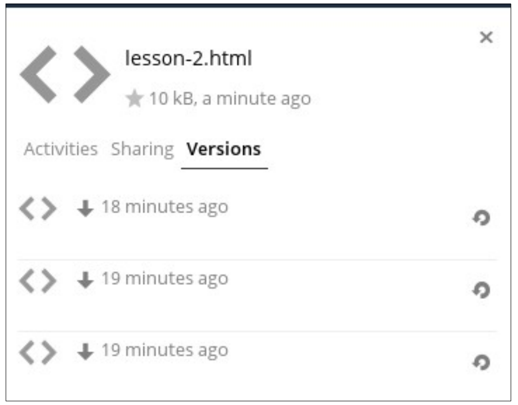
Click here to describe this folder and turn it into a Space

Select a file to see more details

Only you have access

Name	Modified	Members
timg (1)-V20...154033.jpg	6 mins ago	Deleted
timg (2)-V20...154033.jpg	7 mins ago	
timg (3)-V20...154033.jpg	7 mins ago	Only you
timg (4)-V20...154033.jpg	7 mins ago	Only you
timg-V20200...154033.jpg	7 mins ago	Only you

dropbox, 2007



lesson-2.html

10 kB, a minute ago

18 minutes ago

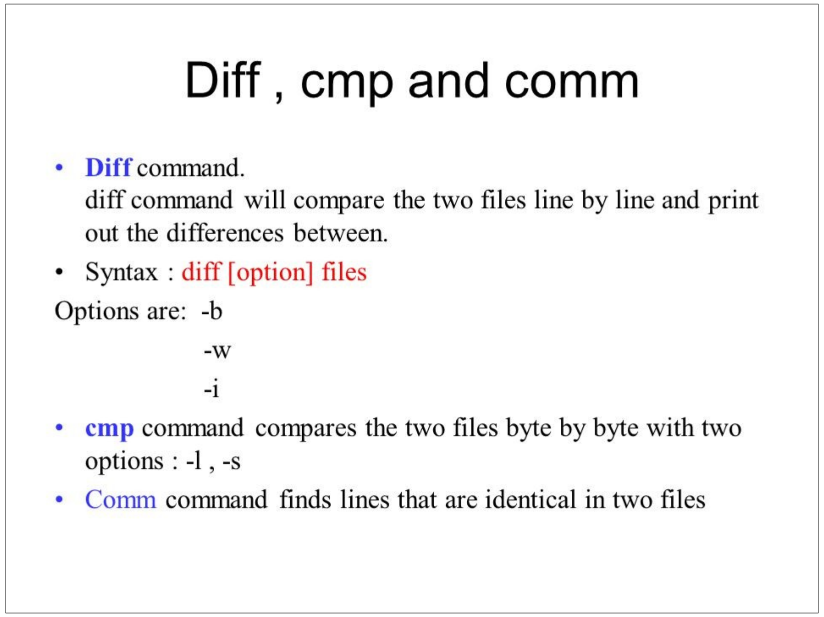
19 minutes ago

19 minutes ago

ownCloud, 2010

Diff utilities

https://en.wikipedia.org/wiki/Diff



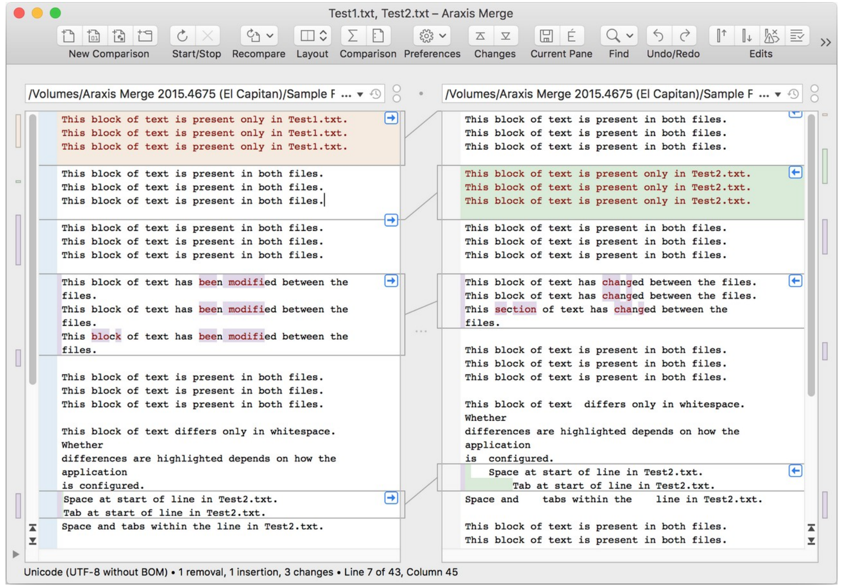
Diff , cmp and comm

- **Diff** command.
  diff command will compare the two files line by line and print out the differences between.
- Syntax : diff [option] files

Options are: -b
-w
-i

- **cmp** command compares the two files byte by byte with two options : -l , -s
- Comm command finds lines that are identical in two files

linux diff, comp, comm, cli tools



/Volumes/Araxis Merge 2015.4675 (El Capitan)/Sample F ...

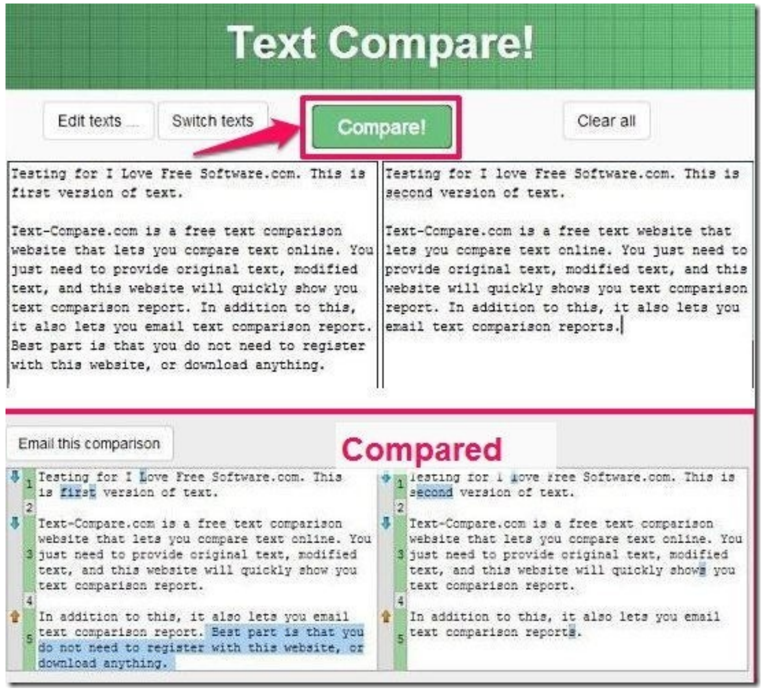
```
This block of text is present only in Test1.txt.
This block of text is present only in Test1.txt.
This block of text is present only in Test1.txt.

This block of text is present in both files.
This block of text is present in both files.
This block of text is present in both files.

This block of text is present in both files.
This block of text is present in both files.
This block of text is present in both files.

This block of text has been modified between the
files.
This block of text has been modified between the
files.
This block of text has been modified between the
files.

This block of text is present in both files.
This block of text is present in both files.
This block of text is present in both files.

This block of text differs only in whitespace.
Whether
differences are highlighted depends on how the
application
is configured.
Space at start of line in Test2.txt.
Tab at start of line in Test2.txt.
Space and tabs within the line in Test2.txt.
```

/Volumes/Araxis Merge 2015.4675 (El Capitan)/Sample F ...

```
This block of text is present in both files.
This block of text is present in both files.
This block of text is present in both files.

This block of text is present only in Test2.txt.
This block of text is present only in Test2.txt.
This block of text is present only in Test2.txt.

This block of text is present in both files.
This block of text is present in both files.
This block of text is present in both files.

This block of text has changed between the files.
This block of text has changed between the files.
This section of text has changed between the
files.

This block of text is present in both files.
This block of text is present in both files.
This block of text is present in both files.

This block of text  differs only in whitespace.
Whether
differences are highlighted depends on how the
application
is  configured.
  Space at start of line in Test2.txt.
        Tab at start of line in Test2.txt.
Space and    tabs within the    line in Test2.txt.

This block of text is present in both files.
This block of text is present in both files.
```

Unicode (UTF-8 without BOM) • 1 removal, 1 insertion, 3 changes • Line 7 of 43, Column 45

Araxis merge, software



```
Testing for I Love Free Software.com. This is
first version of text.

Text-Compare.com is a free text comparison
website that lets you compare text online. You
just need to provide original text, modified
text, and this website will quickly show you
text comparison report. In addition to this,
it also lets you email text comparison report.
Best part is that you do not need to register
with this website, or download anything.
```

```
Testing for I love Free Software.com. This is
second version of text.

Text-Compare.com is a free text website that
lets you compare text online. You just need to
provide original text, modified text, and this
website will quickly shows you text comparison
report. In addition to this, it also lets you
email text comparison reports.
```

```
Testing for I Love Free Software.com. This
is first version of text.

Text-Compare.com is a free text comparison
website that lets you compare text online. You
just need to provide original text, modified
text, and this website will quickly show you
text comparison report.

In addition to this, it also lets you email
text comparison report. Best part is that you
do not need to register with this website, or
download anything.
```

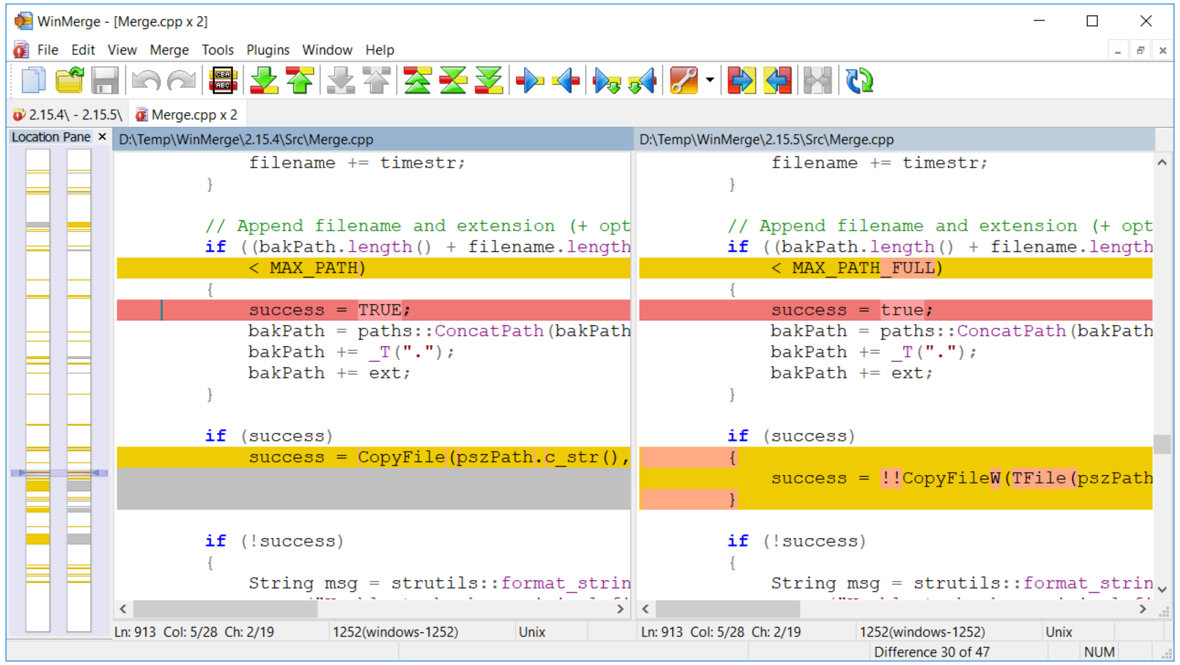
```
Testing for I love Free Software.com. This is
second version of text.

Text-Compare.com is a free text comparison
website that lets you compare text online. You
just need to provide original text, modified
text, and this website will quickly shows you
text comparison report.

In addition to this, it also lets you email
text comparison reports.
```

https://text-compare.com/ , online

Diff utilities

https://en.wikipedia.org/wiki/Diff

- syntax - semantics



WinMerge - [Merge.cpp x 2]
File Edit View Merge Tools Plugins Window Help
2.15.4\ - 2.15.5\
Merge.cpp x 2
Location Pane

D:\Temp\WinMerge\2.15.4\Src\Merge.cpp

```
        filename += timestr;
    }

    // Append filename and extension (+ opt
    if ((bakPath.length() + filename.length
        < MAX_PATH)
    {
        success = TRUE;
        bakPath = paths::ConcatPath(bakPath
        bakPath += _T(".");
        bakPath += ext;
    }

    if (success)
        success = CopyFile(pszPath.c_str(),


    if (!success)
    {
        String msg = strutils::format_strin
```

Ln: 913 Col: 5/28 Ch: 2/19 1252(windows-1252) Unix

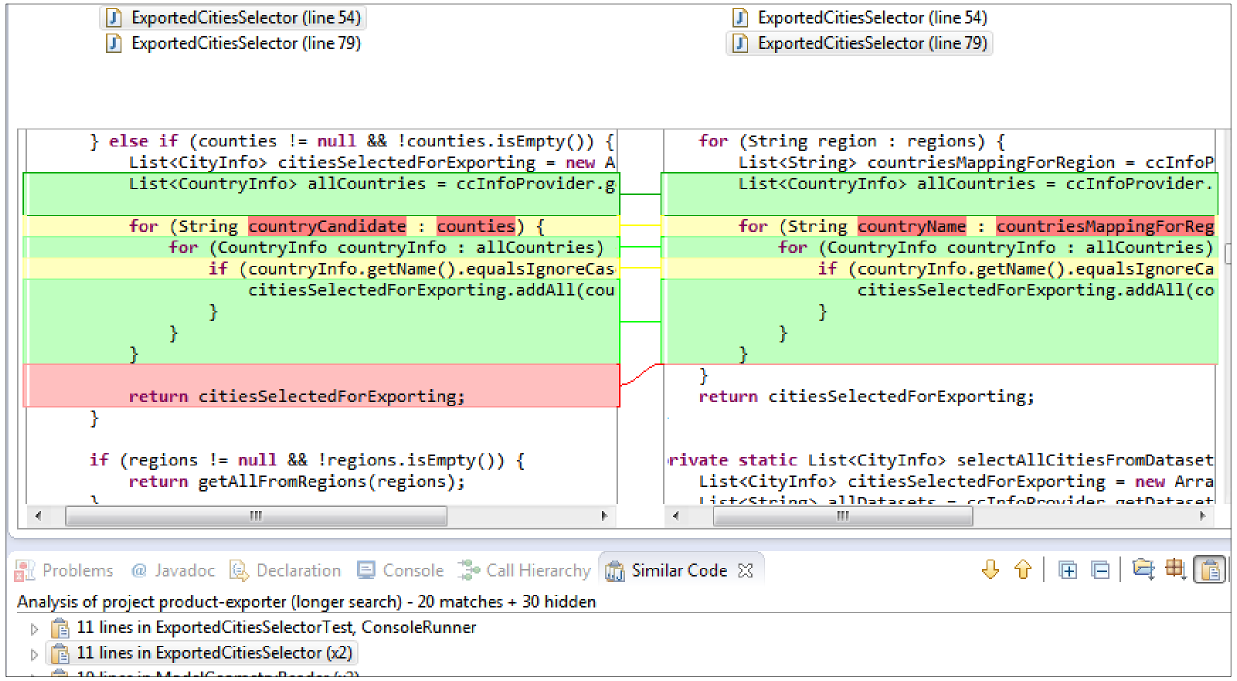
D:\Temp\WinMerge\2.15.5\Src\Merge.cpp

```
        filename += timestr;
    }

    // Append filename and extension (+ opt
    if ((bakPath.length() + filename.length
        < MAX_PATH_FULL)
    {
        success = true;
        bakPath = paths::ConcatPath(bakPath
        bakPath += _T(".");
        bakPath += ext;
    }

    if (success)
    {
        success = !!CopyFileW(TFile(pszPath
    }

    if (!success)
    {
        String msg = strutils::format_strin
```

Ln: 913 Col: 5/28 Ch: 2/19 1252(windows-1252) Unix

Difference 30 of 47 NUM

WinMerge



ExportedCitiesSelector (line 54)
ExportedCitiesSelector (line 79)

```
} else if (counties != null && !counties.isEmpty()) {
    List<CityInfo> citiesSelectedForExporting = new A
    List<CountryInfo> allCountries = ccInfoProvider.g

    for (String countryCandidate : counties) {
        for (CountryInfo countryInfo : allCountries)
            if (countryInfo.getName().equalsIgnoreCas
                citiesSelectedForExporting.addAll(cou
            }
        }
    }

    return citiesSelectedForExporting;
}

if (regions != null && !regions.isEmpty()) {
    return getAllFromRegions(regions);
```

ExportedCitiesSelector (line 54)
ExportedCitiesSelector (line 79)

```
for (String region : regions) {
    List<String> countriesMappingForRegion = ccInfoP
    List<CountryInfo> allCountries = ccInfoProvider.

    for (String countryName : countriesMappingForReg
        for (CountryInfo countryInfo : allCountries)
            if (countryInfo.getName().equalsIgnoreCa
                citiesSelectedForExporting.addAll(co
            }
        }
    }
}
return citiesSelectedForExporting;


rivate static List<CityInfo> selectAllCitiesFromDataset
    List<CityInfo> citiesSelectedForExporting = new Arra
```

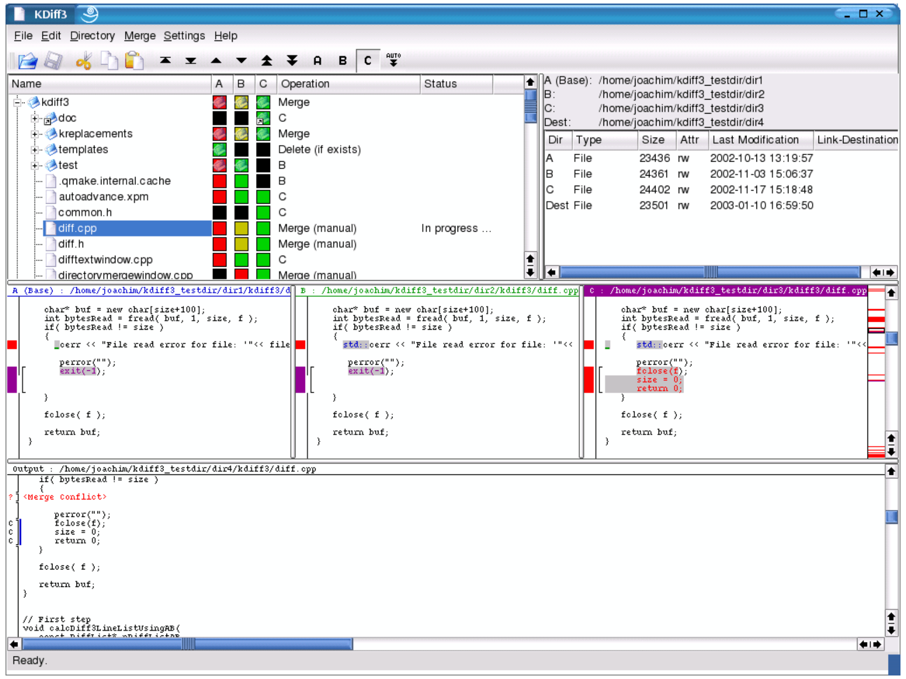
Analysis of project product-exporter (longer search) - 20 matches + 30 hidden
11 lines in ExportedCitiesSelectorTest, ConsoleRunner
11 lines in ExportedCitiesSelector (x2)

In Eclipse

Content comparison utilities

https://en.wikipedia.org/wiki/Content_similarity_detection

Example: plagiarism detection



Name	A	B	C	Operation	Status
kdiff3				Merge	
doc				C	
kreplacements				Merge	
templates				Delete (if exists)	
test				B	
.qmake.internal.cache	🟥	🟩	■	B	
autoadvance.xpm	🟥	🟩	🟩	C	
common.h	■	■	🟩	C	
diff.cpp	🟥	🟨	🟩	Merge (manual)	In progress ...
diff.h	🟥	🟨	🟩	Merge (manual)	
difftextwindow.cpp	🟥	🟩	🟩	C	
directorymergewindow.cpp	■	🟥	🟩	Merge (manual)	

```
A (Base):  /home/joachim/kdiff3_testdir/dir1
B:         /home/joachim/kdiff3_testdir/dir2
C:         /home/joachim/kdiff3_testdir/dir3
Dest:      /home/joachim/kdiff3_testdir/dir4
```

Dir	Type	Size	Attr	Last Modification	Link-Destination
A	File	23436	rw	2002-10-13 13:19:57	
B	File	24361	rw	2002-11-03 15:06:37	
C	File	24402	rw	2002-11-17 15:18:48	
Dest	File	23501	rw	2003-01-10 16:59:50	

```
A (Base) : /home/joachim/kdiff3_testdir/dir1/kdiff3/d
    char* buf = new char[size+100];
    int bytesRead = fread( buf, 1, size, f );
    if( bytesRead != size )
    {
      cerr << "File read error for file: '"<< file

       perror("");
       exit(-1);

    }

    fclose( f );

    return buf;
 }
```

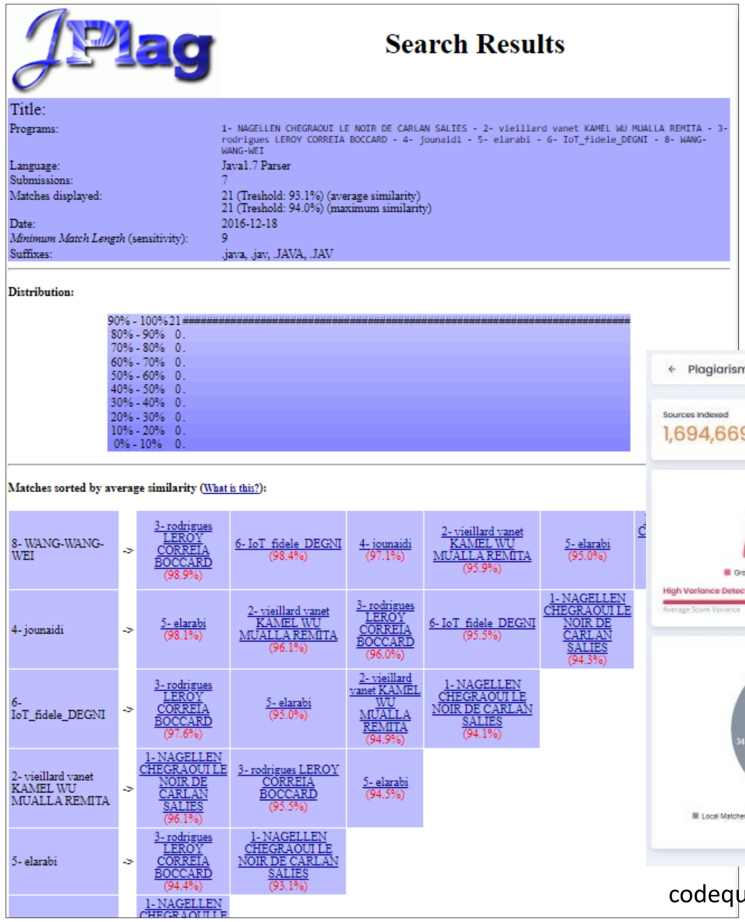
```
B : /home/joachim/kdiff3_testdir/dir2/kdiff3/diff.cpp
    char* buf = new char[size+100];
    int bytesRead = fread( buf, 1, size, f );
    if( bytesRead != size )
    {
      std::cerr << "File read error for file: '"<<

       perror("");
       exit(-1);

    }

    fclose( f );

    return buf;
 }
```

```
C : /home/joachim/kdiff3_testdir/dir3/kdiff3/diff.cpp
    char* buf = new char[size+100];
    int bytesRead = fread( buf, 1, size, f );
    if( bytesRead != size )
    {
      std::cerr << "File read error for file: '"<<

       perror("");
       fclose(f);
       size = 0;
       return 0;
    }

    fclose( f );

    return buf;
 }
```

```
Output : /home/joachim/kdiff3_testdir/dir4/kdiff3/diff.cpp
     if( bytesRead != size )
     {
? ] <Merge Conflict>

       perror("");
c      fclose(f);
c      size = 0;
c      return 0;
     }

     fclose( f );

     return buf;
  }


  // First step
  void calcDiff3LineListUsingAB(
```

Ready.

KDiff3, open-source



Search Results

Title:

Programs:	1- NAGELLEN CHEGRAOUI LE NOIR DE CARLAN SALIES - 2- vieillard vanet KAMEL WU MUALLA REMITA - 3- rodrigues LEROY CORREIA BOCCARD - 4- jounaidi - 5- elarabi - 6- IoT_fidele_DEGNI - 8- WANG-WANG-WEI
Language:	Java1.7 Parser
Submissions:	7
Matches displayed:	21 (Treshold: 93.1%) (average similarity) 21 (Treshold: 94.0%) (maximum similarity)
Date:	2016-12-18
Minimum Match Length (sensitivity):	9
Suffixes:	.java, .jav, .JAVA, .JAV

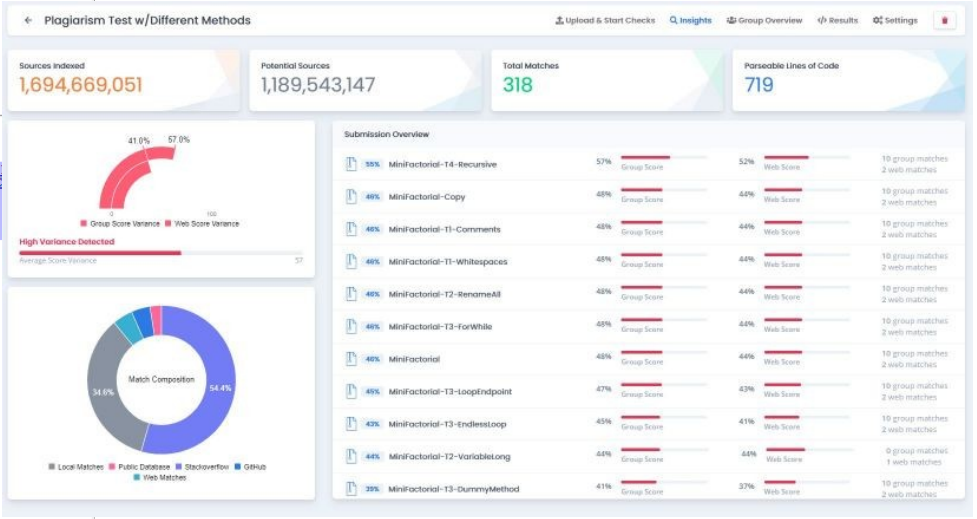
Distribution:

```
90% - 100%21 ##########################################################################
 80% - 90%   0.
 70% - 80%   0.
 60% - 70%   0.
 50% - 60%   0.
 40% - 50%   0.
 30% - 40%   0.
 20% - 30%   0.
 10% - 20%   0.
  0% - 10%   0.
```

Matches sorted by average similarity (What is this?):

8- WANG-WANG-WEI	->	3- rodrigues LEROY CORREIA BOCCARD (98.9%)	6- IoT_fidele_DEGNI (98.4%)	4- jounaidi (97.1%)	2- vieillard vanet KAMEL WU MUALLA REMITA (95.9%)	5- elarabi (95.0%)
4- jounaidi	->	5- elarabi (98.1%)	2- vieillard vanet KAMEL WU MUALLA REMITA (96.1%)	3- rodrigues LEROY CORREIA BOCCARD (96.0%)	6- IoT_fidele_DEGNI (95.5%)	1- NAGELLEN CHEGRAOUI LE NOIR DE CARLAN SALIES (94.3%)
6- IoT_fidele_DEGNI	->	3- rodrigues LEROY CORREIA BOCCARD (97.6%)	5- elarabi (95.0%)	2- vieillard vanet KAMEL WU MUALLA REMITA (94.9%)	1- NAGELLEN CHEGRAOUI LE NOIR DE CARLAN SALIES (94.1%)	
2- vieillard vanet KAMEL WU MUALLA REMITA	->	1- NAGELLEN CHEGRAOUI LE NOIR DE CARLAN SALIES (96.1%)	3- rodrigues LEROY CORREIA BOCCARD (95.5%)	5- elarabi (94.5%)		
5- elarabi	->	3- rodrigues LEROY CORREIA BOCCARD (94.4%)	1- NAGELLEN CHEGRAOUI LE NOIR DE CARLAN SALIES (93.1%)			
		1- NAGELLEN CHEGRAOUI LE				

Jplag, freeware



Sources Indexed	Potential Sources	Total Matches	Parseable Lines of Code
1,694,669,051	1,189,543,147	318	719

Submission Overview

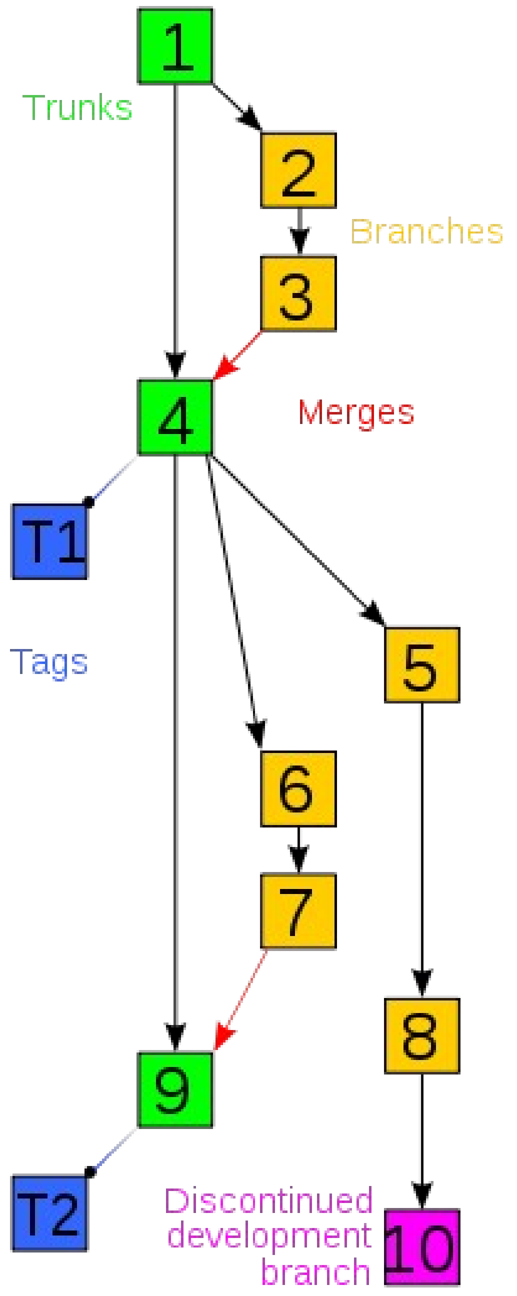
55%	MiniFactorial-T4-Recursive	57% Group Score	52% Web Score	10 group matches 2 web matches
46%	MiniFactorial-Copy	48% Group Score	44% Web Score	10 group matches 2 web matches
46%	MiniFactorial-T1-Comments	48% Group Score	44% Web Score	10 group matches 2 web matches
46%	MiniFactorial-T1-Whitespaces	48% Group Score	44% Web Score	10 group matches 2 web matches
46%	MiniFactorial-T2-RenameAll	48% Group Score	44% Web Score	10 group matches 2 web matches
46%	MiniFactorial-T3-ForWhile	48% Group Score	44% Web Score	10 group matches 2 web matches
46%	MiniFactorial	48% Group Score	44% Web Score	10 group matches 2 web matches
45%	MiniFactorial-T3-LoopEndpoint	47% Group Score	43% Web Score	10 group matches 2 web matches
43%	MiniFactorial-T3-EndlessLoop	45% Group Score	41% Web Score	10 group matches 2 web matches
44%	MiniFactorial-T2-VariableLong	44% Group Score	44% Web Score	0 group matches 1 web matches
39%	MiniFactorial-T3-DummyMethod	41% Group Score	37% Web Score	10 group matches 2 web matches

codequiry, shareware

VCS- Version Control System

Definition

A tool that helps developers/programmers solve certain day-to-day problems, such as: tracking code changes, helping with code maintenance, and allowing them to work on the same source code files without affecting each other's workflow.

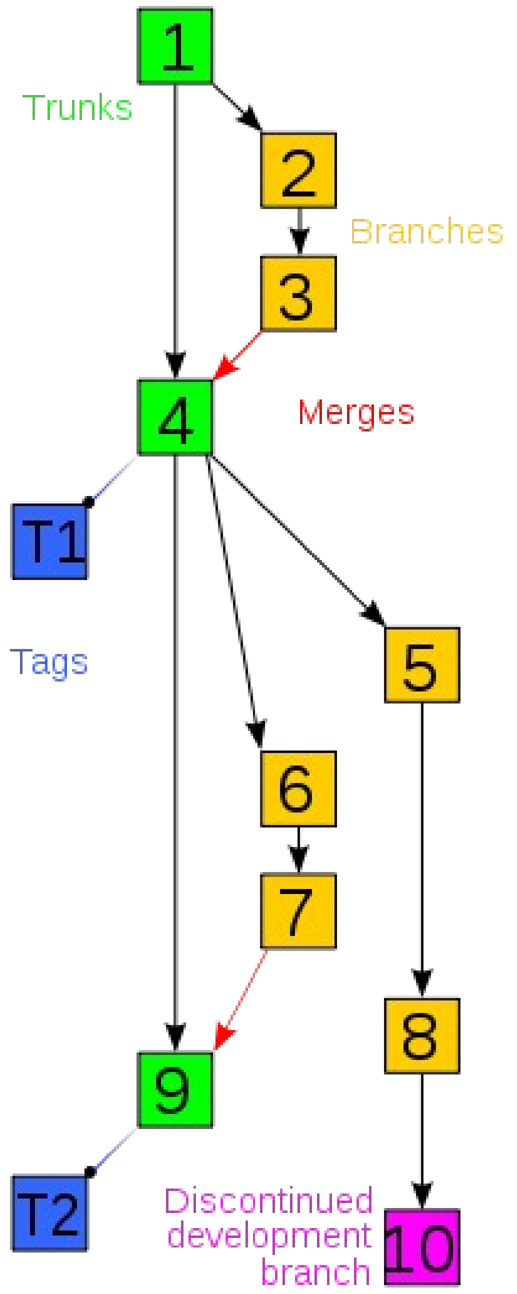


https://en.wikipedia.org/wiki/Version_control

VCS- Version Control System

Objectives

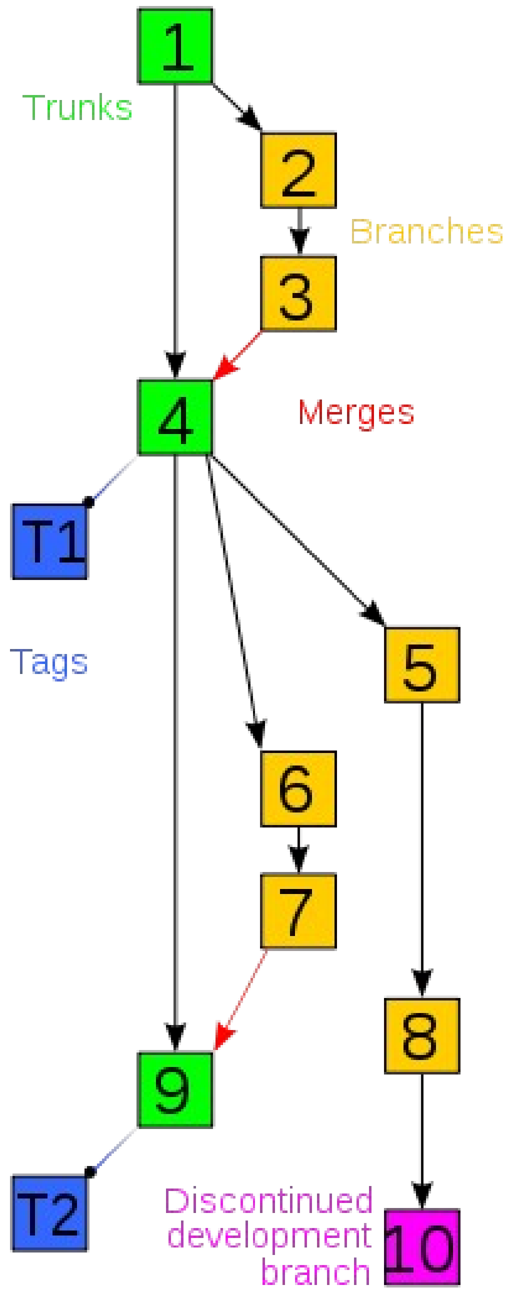
- Generate backups
- Test and experiment
- Keep history and track changes
- Collaborate and contribute remotely



https://en.wikipedia.org/wiki/Version_control

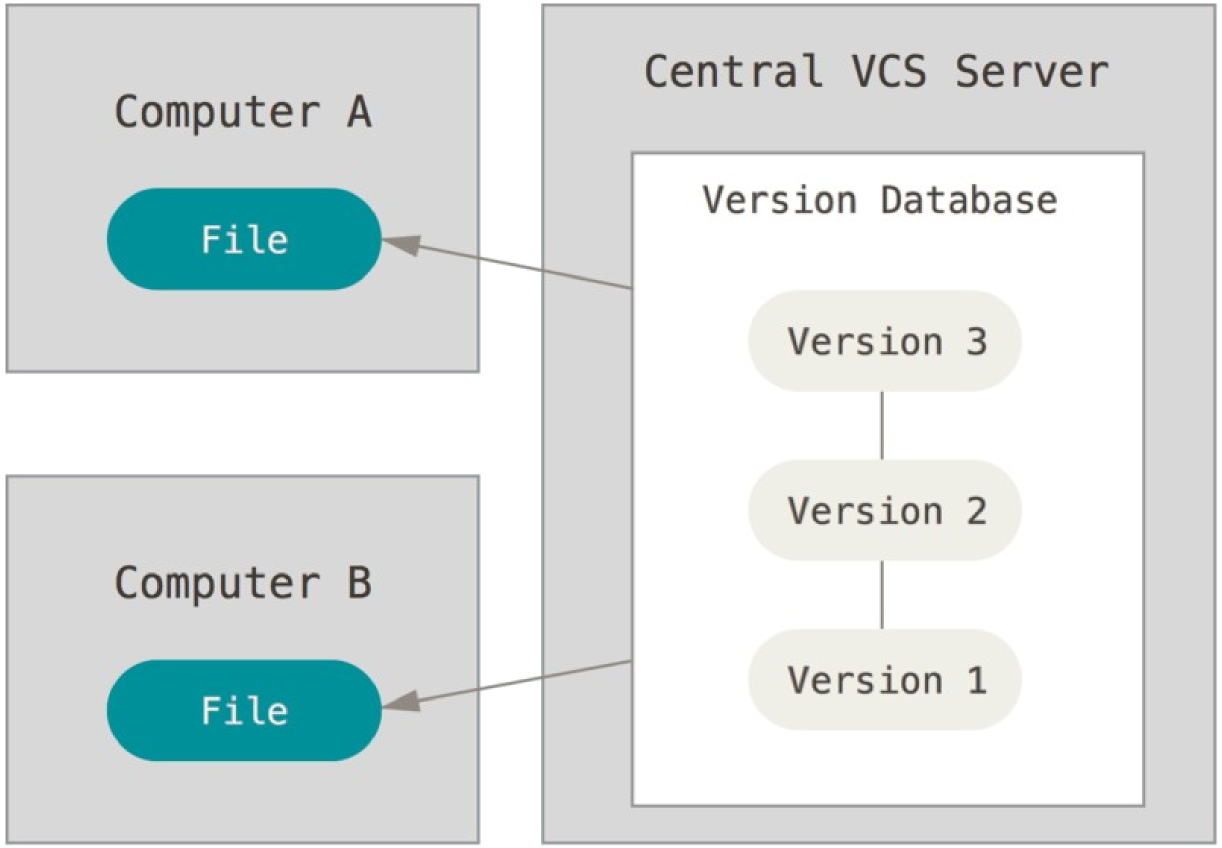
Concepts

- make a **local copy** of a **remote repository**
- make changes, **commit** changes ("submit")
- divergent **branches** containing version sequences
- **merge** branches, with resolution of possible conflicts
- **tag** versions
- propose revisions (PR, **pull request**)



https://en.wikipedia.org/wiki/Version_control

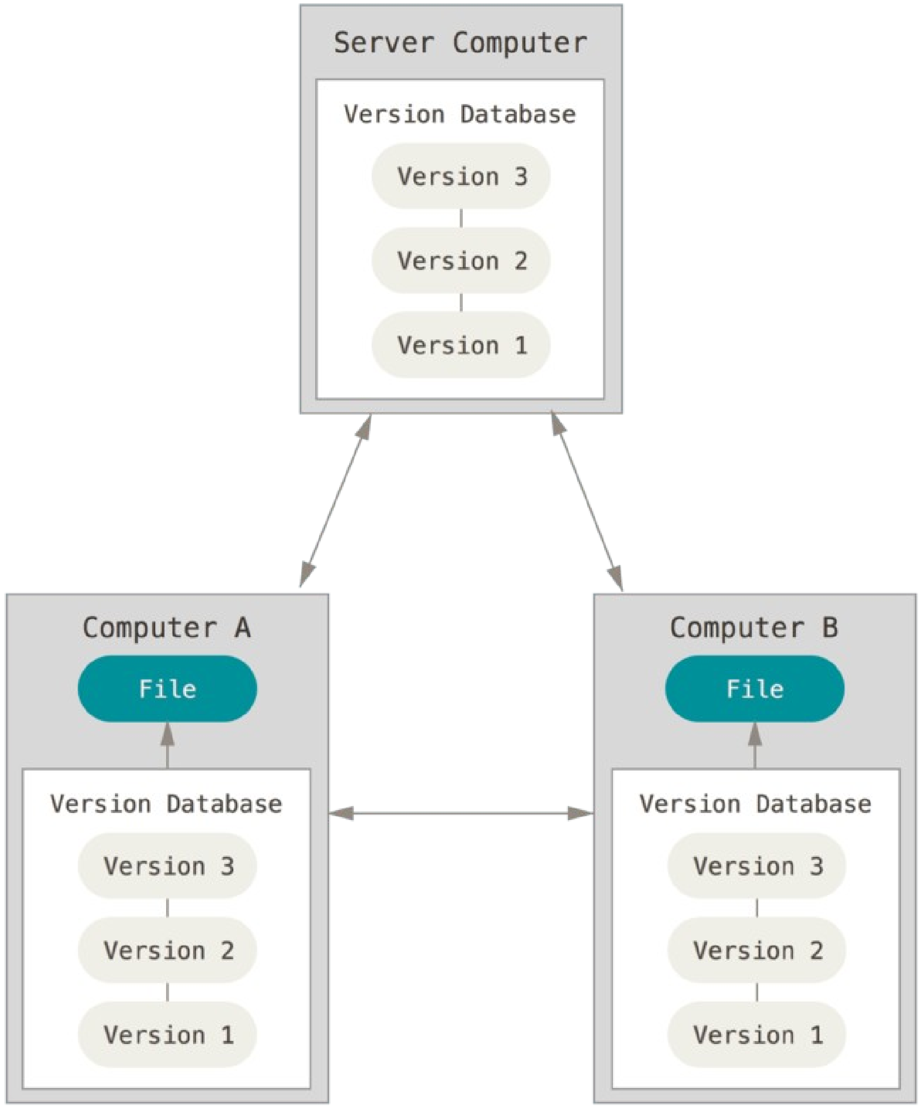
Centralized version management



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

Main limitation:
single point of failure

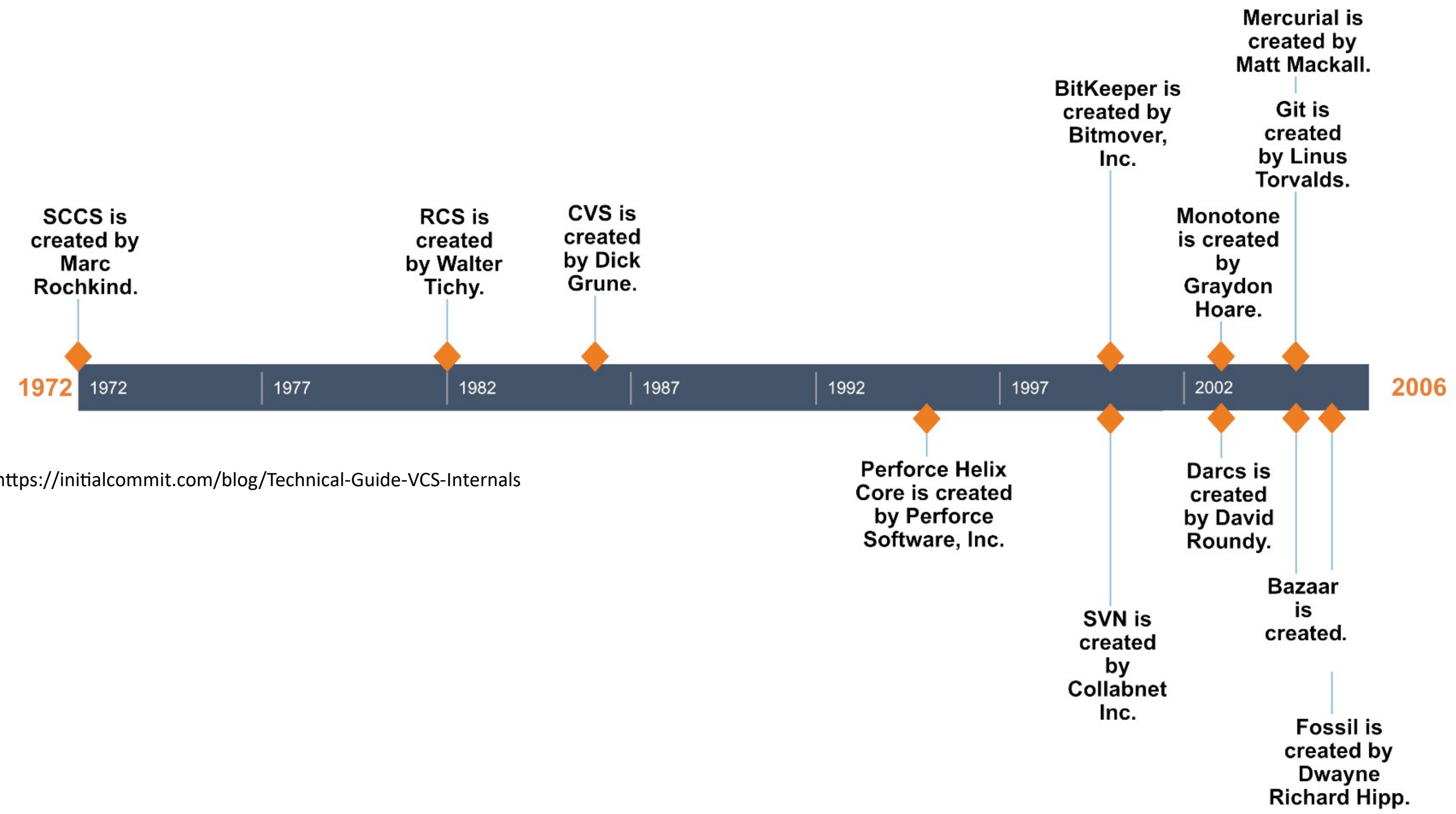
Distributed version management



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

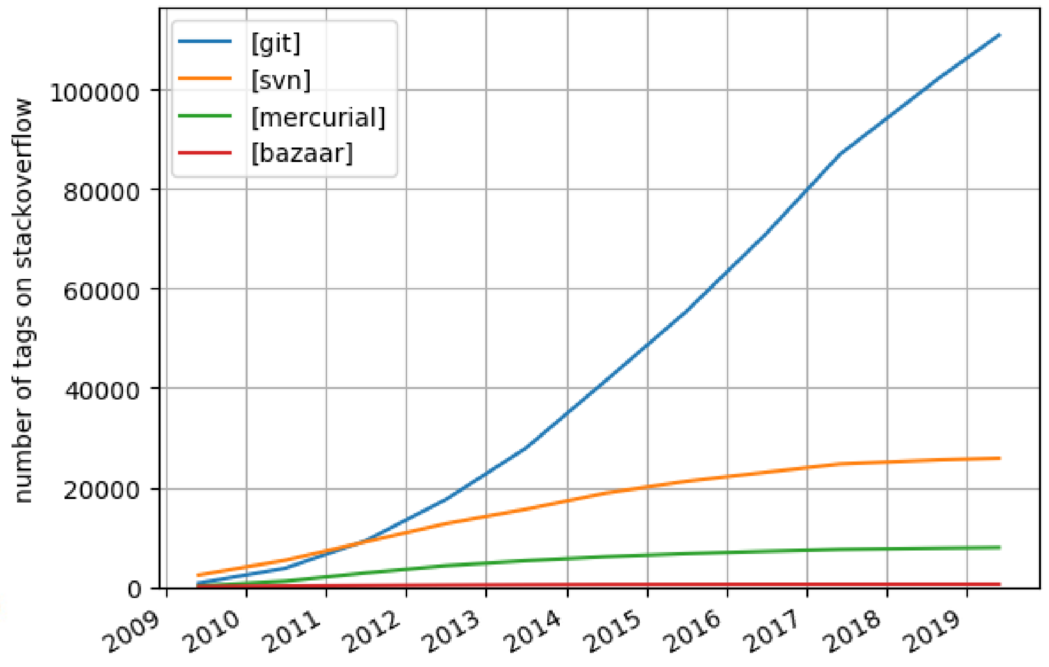
(Bazaar)

History



SCCS is created by Marc Rochkind.
RCS is created by Walter Tichy.
CVS is created by Dick Grune.
BitKeeper is created by Bitmover, Inc.
Mercurial is created by Matt Mackall.
Git is created by Linus Torvalds.
Monotone is created by Graydon Hoare.
1972
1972
1977
1982
1987
1992
1997
2002
2006
Perforce Helix Core is created by Perforce Software, Inc.
SVN is created by Collabnet Inc.
Darcs is created by David Roundy.
Bazaar is created.
Fossil is created by Dwayne Richard Hipp.

https://initialcommit.com/blog/Technical-Guide-VCS-Internals



https://askcodez.com/la-popularite-de-git-mercurial-bazar-vs-qui-a-recommander.html

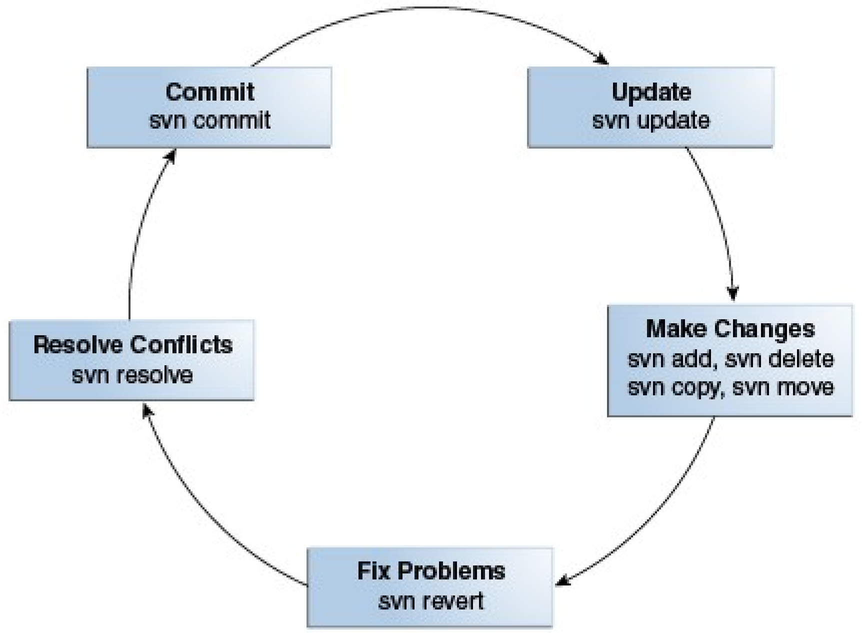
SVN – Apache Subversion

Homepage http://subversion.apache.org/

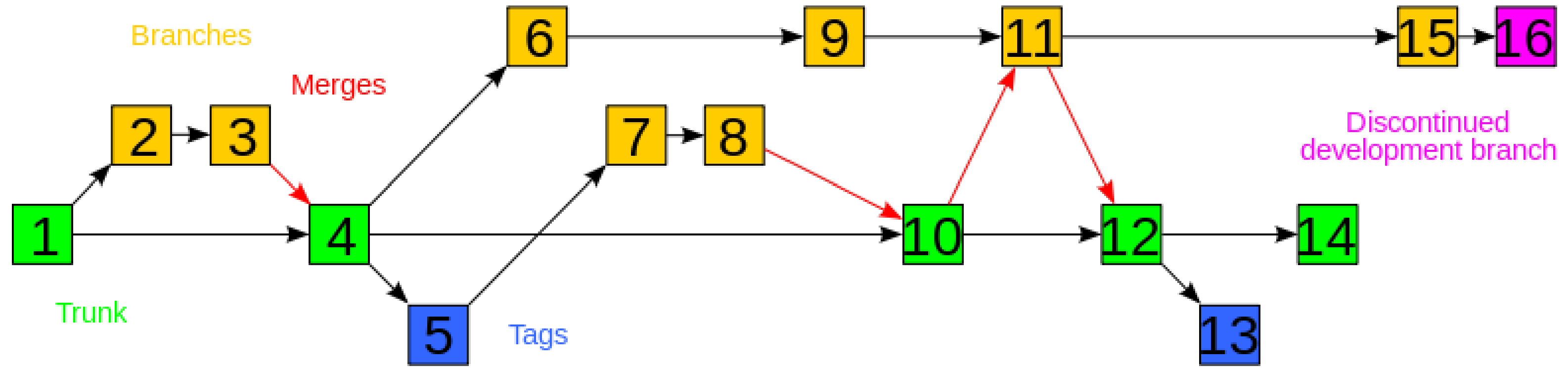
SVN Book http://svnbook.red-bean.com/

Limitations

- centralized system
- no time stamping
- no history management, global version numbering
- network almost always necessary
- poorly managed "move" operation (delete + add)
- no normalization of file names



https://docs.oracle.com/middleware/1221/core/MAVEN/config_svn.htm



https://en.wikipedia.org/wiki/Apache_Subversion

Git

(which means "unpleasant person" in British English slang). : "I'm an egotistical bastard, and I name all my projects after myself. First **'Linux'**, now **'git'**." The **man** page describes Git as "the stupid content tracker".

developers initialy Linus Torvalds. Mainly Junio Hamano +1620 to update: https://github.com/git/git/graphs/contributors

first version 2005

current version v2.41.0 to update: https://github.com/git/git/releases

license GPLv2 (free, open-source)

history

- Linux kernel contributions before 2002: patches transmitted and integrated by hand
- Linux kernel development 2002-2005: VCS distributed BitKeeper
- 2005: BitKeeper becomes payware, Linux Torvalds develops git with the following goals:
    - speed ;
    - simple design ;
    - support for non-linear development (thousands of parallel branches) ;
    - fully distributed ;
    - ability to efficiently manage large projects such as the Linux kernel (speed and data compactness)

https://en.wikipedia.org/wiki/Git
https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Une-rapide-histoire-de-Git

Technological foundations of software development

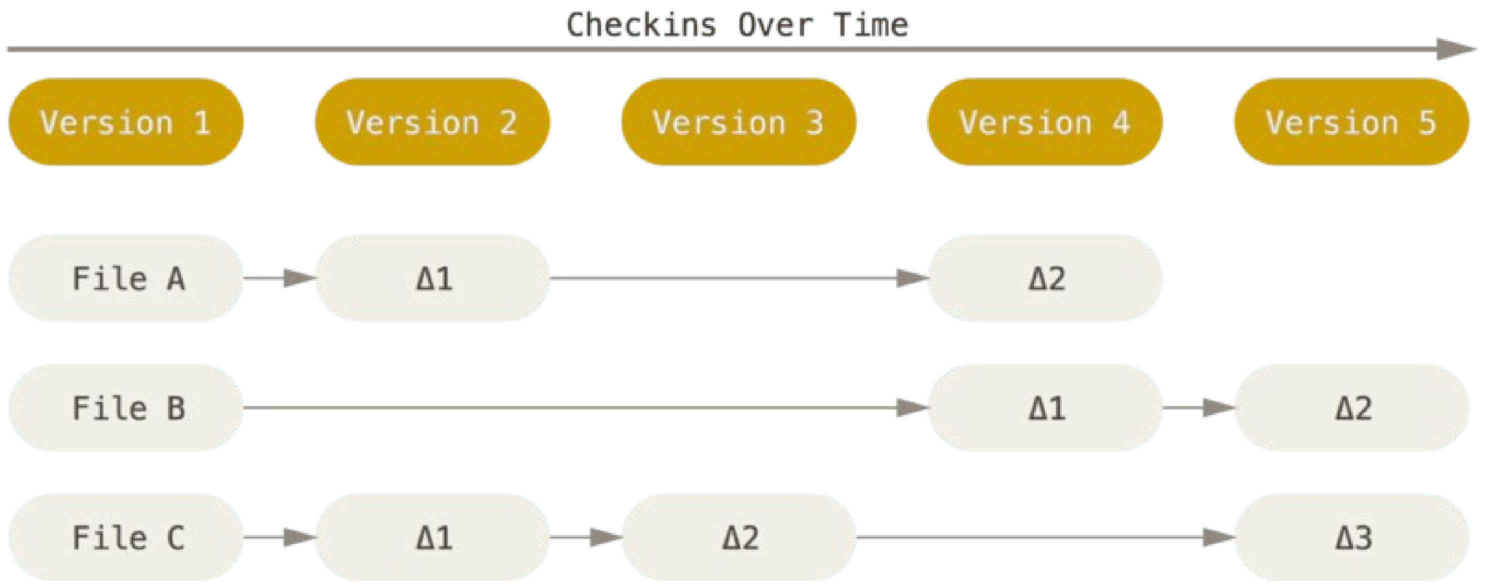
Manage your source code

Part 2 – Git basics

ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

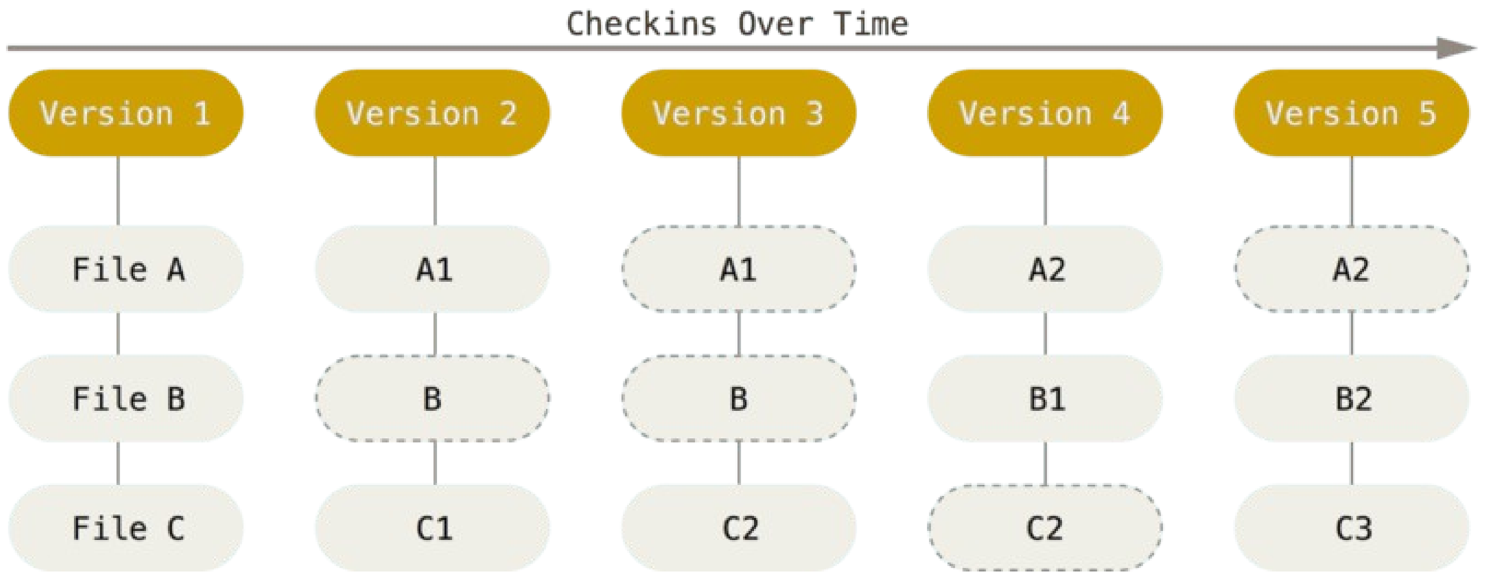
Philosophy

CVS, Subversion, Perforce, Bazaar, etc.
Save information as changes to files



How git works:
Stores data as snapshots of the project over time

(snapshot flow)



https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F

Philosophy

Almost all operations are local

- remote repository ≈ local directory
- no need to constantly access the central server

Git manages integrity

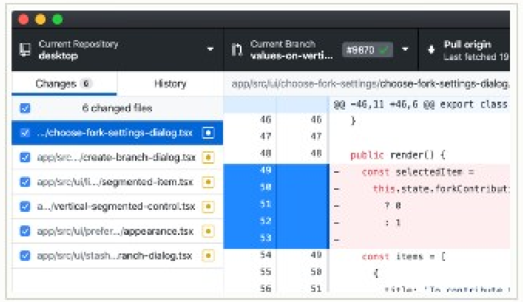
- id of a repo state = checksum(previous id, changes)
- SHA-1 (40 hex characters, example `24b9da6552252987aa493b52f8696cd6d3b00373`)
- git indexes the data by these checksum

Generally, Git only adds data

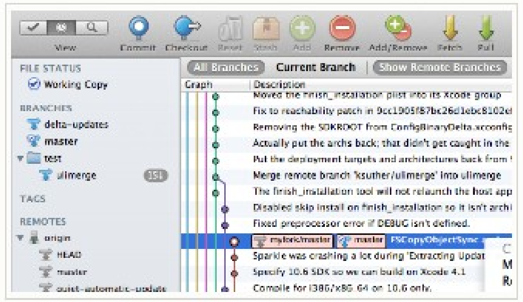
- almost impossible to lose permanently a repo state
- even undo actions are stored as a new change
- gives freedom to experiment safely

https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F

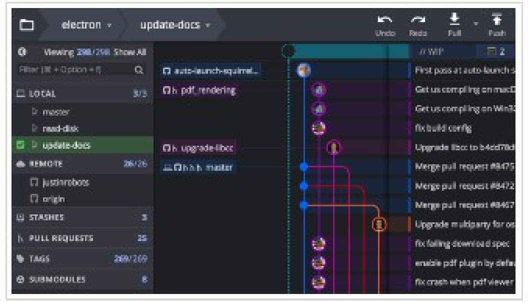
How to use git- softwares



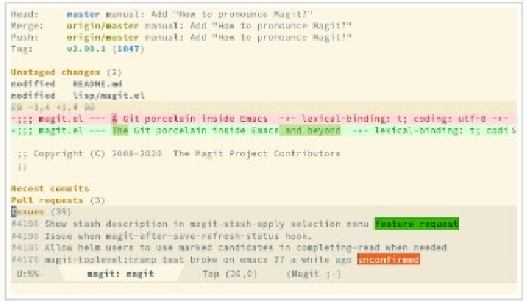
GitHub Desktop
Platforms: Mac, Windows
Price: Free
License: MIT



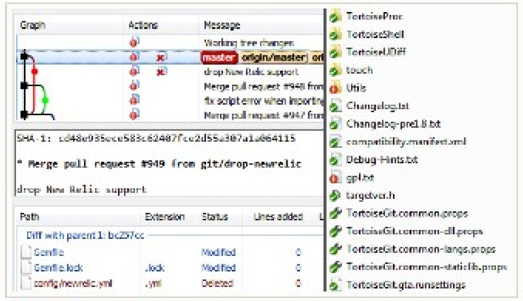
SourceTree
Platforms: Mac, Windows
Price: Free
License: Proprietary



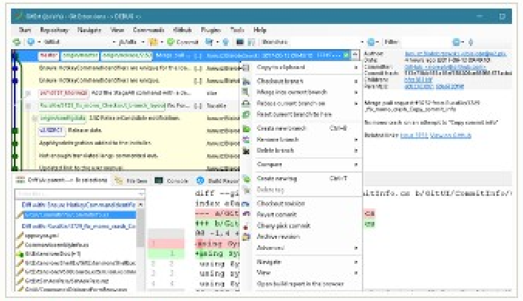
GitKraken
Platforms: Linux, Mac, Windows
Price: Free / $29 / $49
License: Proprietary



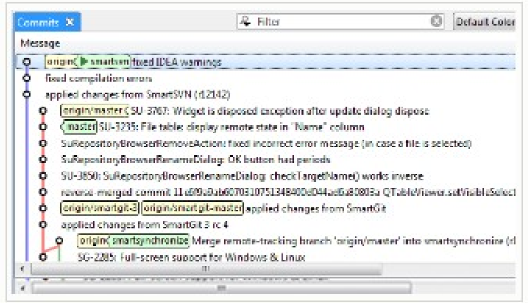
Magit
Platforms: Linux, Mac, Windows
Price: Free
License: GNU GPL



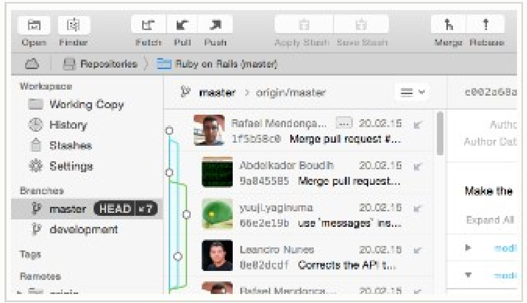
TortoiseGit
Platforms: Windows
Price: Free
License: GNU GPL



Git Extensions
Platforms: Linux, Mac, Windows
Price: Free
License: GNU GPL



SmartGit
Platforms: Linux, Mac, Windows
Price: $79/user / Free for non-commercial use
License: Proprietary

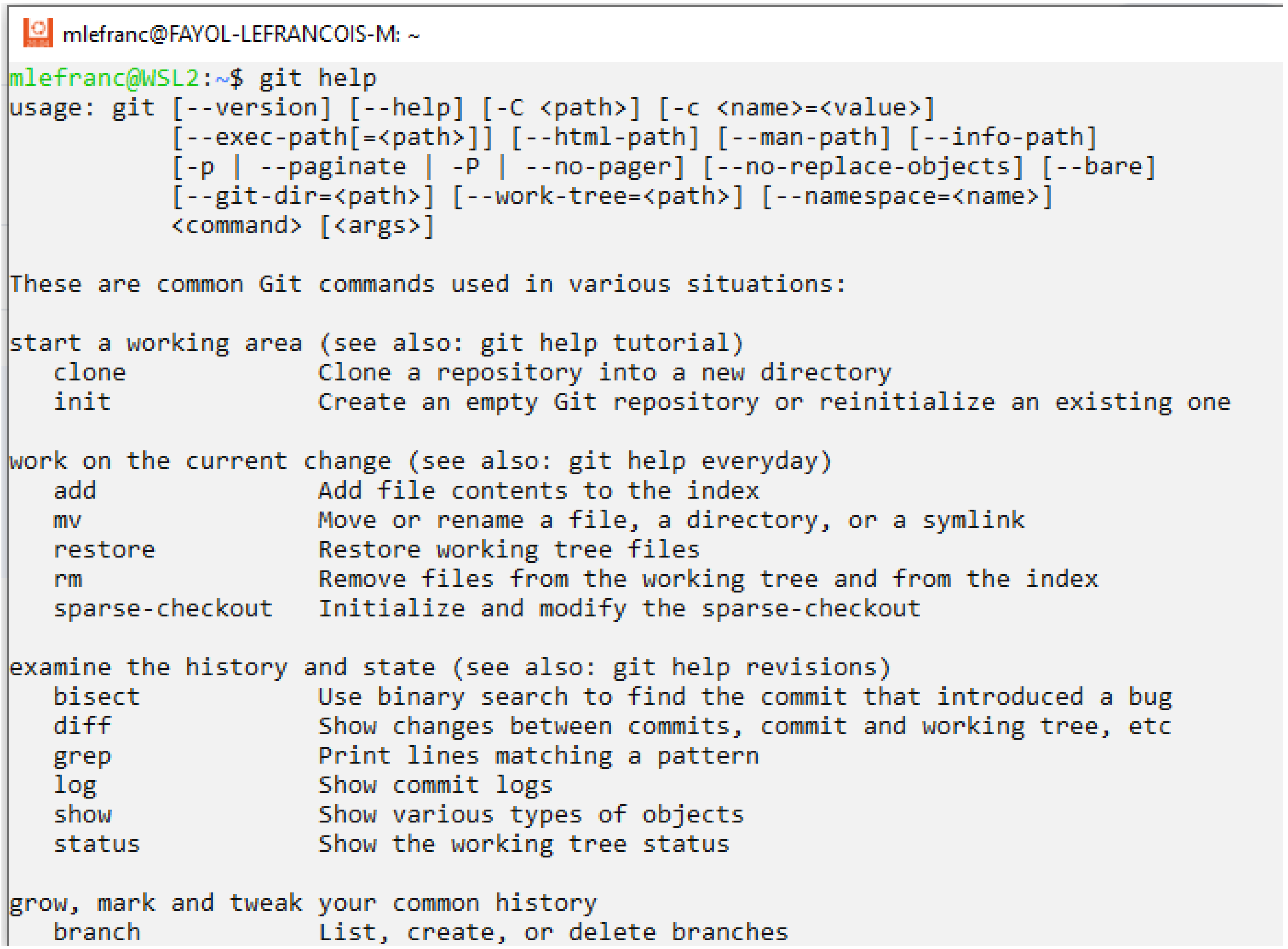


Tower
Platforms: Mac, Windows
Price: $69/user (Free 30 day trial)
License: Proprietary

…

https://git-scm.com/downloads/guis

How to use git- CLI



mlefranc@FAYOL-LEFRANCOIS-M: ~

```
mlefranc@WSL2:~$ git help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone             Clone a repository into a new directory
   init              Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add               Add file contents to the index
   mv                Move or rename a file, a directory, or a symlink
   restore           Restore working tree files
   rm                Remove files from the working tree and from the index
   sparse-checkout   Initialize and modify the sparse-checkout

examine the history and state (see also: git help revisions)
   bisect            Use binary search to find the commit that introduced a bug
   diff              Show changes between commits, commit and working tree, etc
   grep              Print lines matching a pattern
   log               Show commit logs
   show              Show various types of objects
   status            Show the working tree status

grow, mark and tweak your common history
   branch            List, create, or delete branches
```

How to use git- CLI

- Obtain help

```
$ git help <commande>
$ git <commande> --help
$ man git-<commande>
```

- Obtain a concise version of the help

```
$ git <commande> -h
```

https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Obtenir-de-l%E2%80%99aide

Option 1 to start a Git repository:
Initialize a Git repository in a directory

1
```
$ git init
Initialized empty Git repository in .
```

2
```
$ tree -a .
.
└── .git
    ├── HEAD
    ├── branches
    ├── config
    ├── description
    ├── hooks
    │   ├── applypatch-msg.sample
    │   ├── commit-msg.sample
    │   ├── fsmonitor-watchman.sample
    │   ├── post-update.sample
    │   ├── pre-applypatch.sample
    │   ├── pre-commit.sample
    │   ├── pre-merge-commit.sample
    │   ├── pre-push.sample
    │   ├── pre-rebase.sample
    │   ├── pre-receive.sample
    │   ├── prepare-commit-msg.sample
    │   └── update.sample
    ├── info
    │   └── exclude
    ├── objects
    │   ├── info
    │   └── pack
    └── refs
        ├── heads
        └── tag
```

3
```
$ git add *.html
$ git add README.md
$ git commit –m 'first version'
```

Option 2 to start a Git repository:

Clone an existing Git repository

1

```
$ git clone https://github.com/FFmpeg/FFmpeg
Cloning into 'FFmpeg'...
remote: Enumerating objects: 633677, done.
remote: Total 633677 (delta 0), reused 0 (delta 0), pack-reused 633677
Receiving objects: 100% (633677/633677), 263.28 MiB | 11.00 MiB/s, done.
Resolving deltas: 100% (498066/498066), done.
Updating files: 100% (7479/7479), done.
```

2

```
$ cd FFmpeg/
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

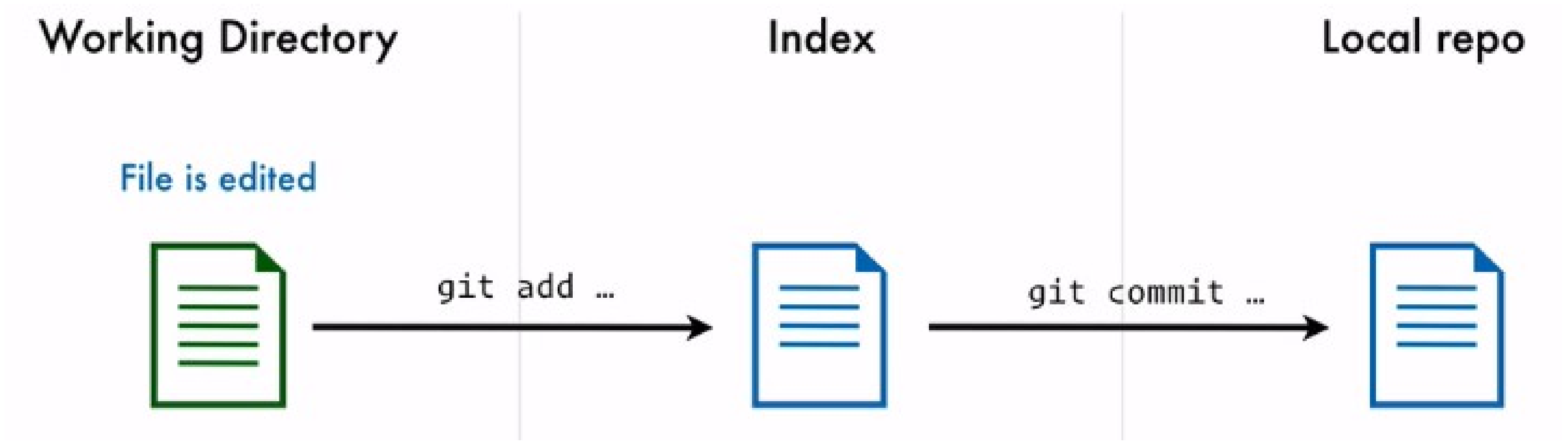
https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository

Three file states: **modified**, **indexed**, **validated**.

modified: file modified but not validated in the DB

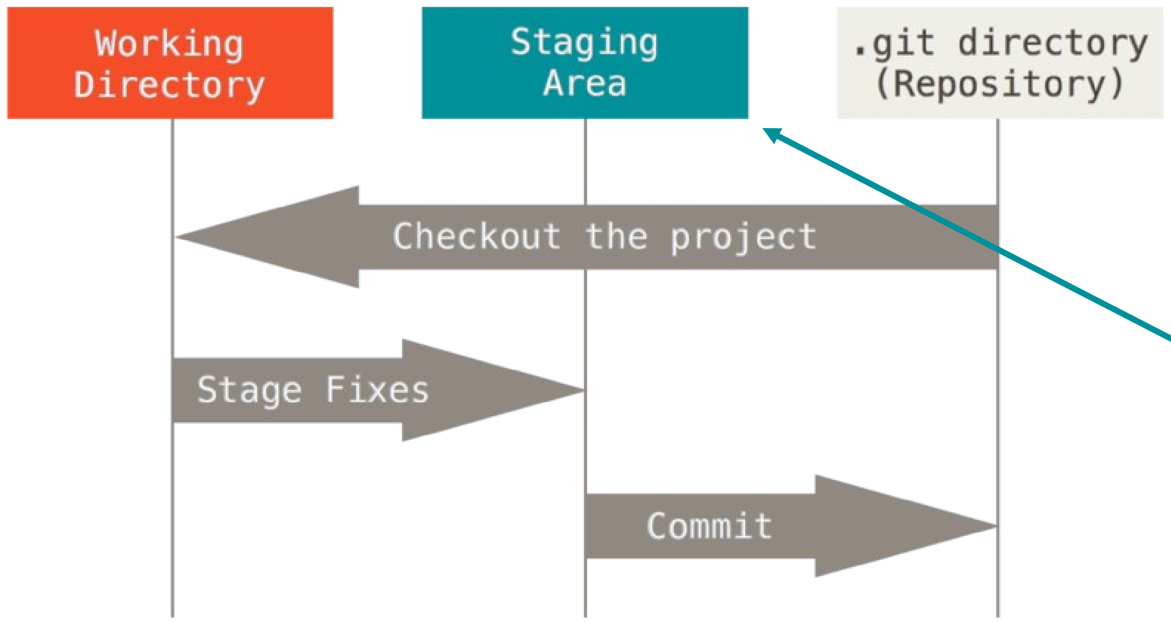
indexed: file marked to be part of the next snapshot

validated: data safely stored in the local database



Three repo areas:
working directory, index area, git directory

a unique project extraction



Where metadata, project database, compressed snapshots are stored

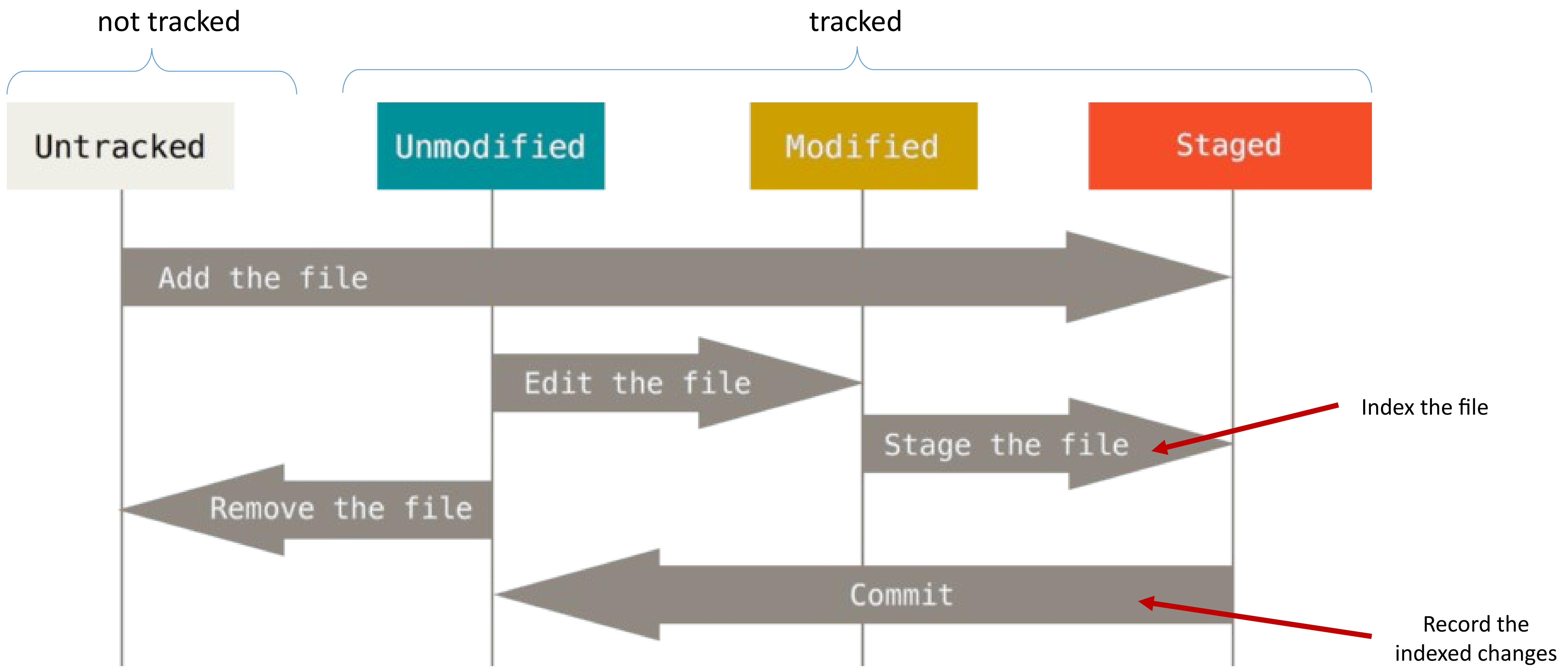
copied when cloning from a remote repository

simple file that stores what will be in the next snapshot

Working directory, index area and Git directory

Record changes to the repository

File states life cycle



https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Record changes to the repository

Check the files state

```
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

- no tracked files have been modified
- no untracked files
- master branch

```
$ echo "My project" > README.md
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        README.md

nothing added to commit but untracked files present (use "git
add" to track)
```

- new untracked file detected

Record changes to the repository

Index changed files

```
$ nano index.html
```

- editing an existing file

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
directory)
        modified:   index.html
```

- `README.md` file tracked and indexed
- `index.html` file modified

https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Record changes to the repository

Index changed files

```
$ git add index.html
```

- Index index.html

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README.md
        modified:   index.html
```

- README.md tracked and indexed
- index.html file tracked and indexed

```
$ nano index.html
```

- modified index.html

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README.md
        modified:   index.html
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
directory)
        modified:   index.html
```

?

https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Record changes to the repository

Index changed files

```
$ git add index.html
```

- Index `index.html`

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README.md
        modified:   index.html
```

- `README.md` tracked and indexed
- `index.html` file tracked and indexed

```
$ nano index.html
```

- modified `index.html`

```
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   README.md
        modified:   index.html
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working
directory)
        modified:   index.html
```

- `index.html` file indexed when running `git add`
- then modified

Record changes to the repository

Commit changes

```
$ git commit –h
usage: git commit [<options>] [--] <pathspec>...

Commit message options
    -F, --file <file>     read message from file
    --author <author>     override author for commit
    --date <date>         override date for commit
    -m, --message <message>
                          commit message
    --status              include status in commit message template
    ...

Commit contents options
    -a, --all             commit all changed files
    -i, --include         add specified files to index for commit
    --dry-run             show what would be committed
    --short               show status concisely
    --branch              show branch information
    --amend               amend previous commit
    ...
```

- `git commit -a`
  skip the indexing step

https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Technological foundations of software development

Manage your source code

Part 3 – Branching and merging with Git

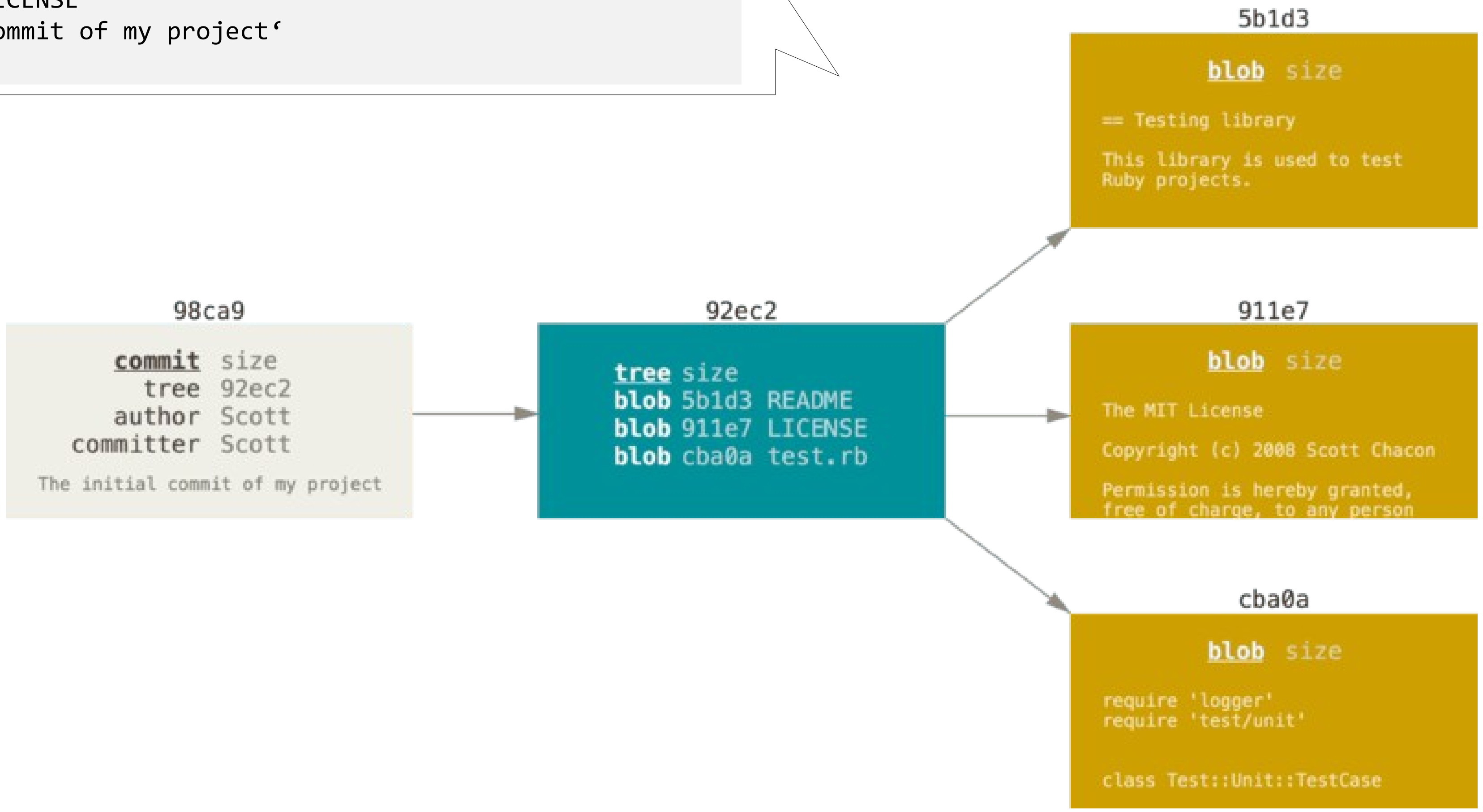
ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

Branches in a nutshell

A commit and its tree

First commit of a repository with three files indexed, then validated
Calculated checksum: SHA-1

```
$ git init
$ git add README test.rb LICENSE
$ git commit -m 'initial commit of my project‘
$ git tree -a
```

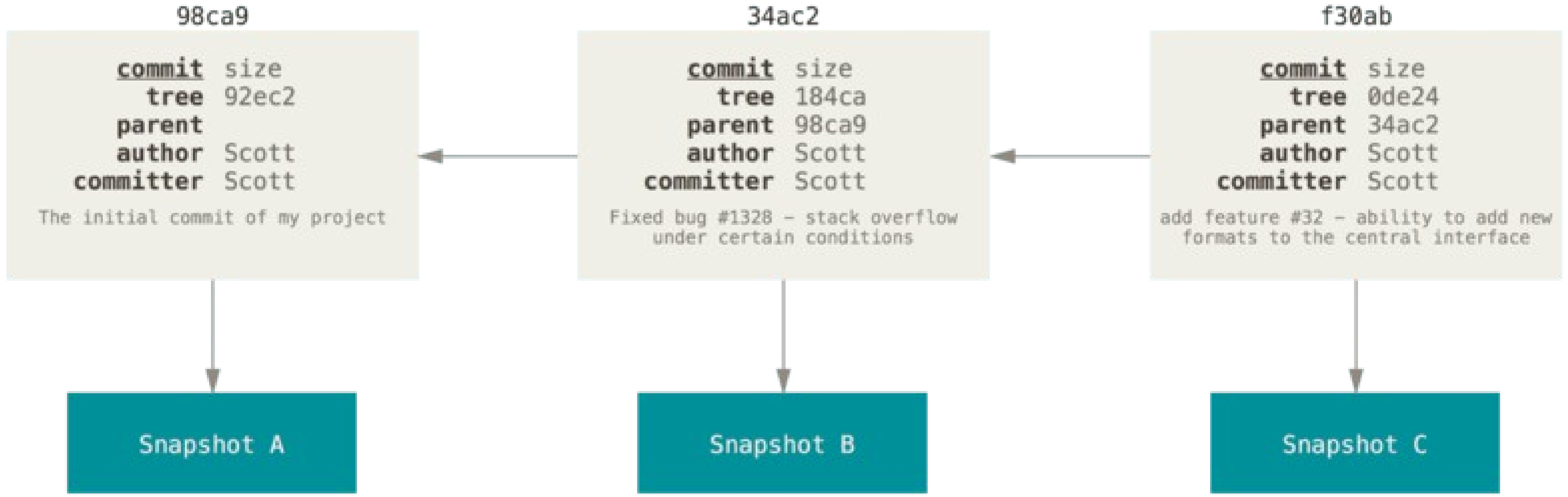


https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

Branches in a nutshell

Commits and their parents

New changes: each commit stores a pointer to the previous commit(s)



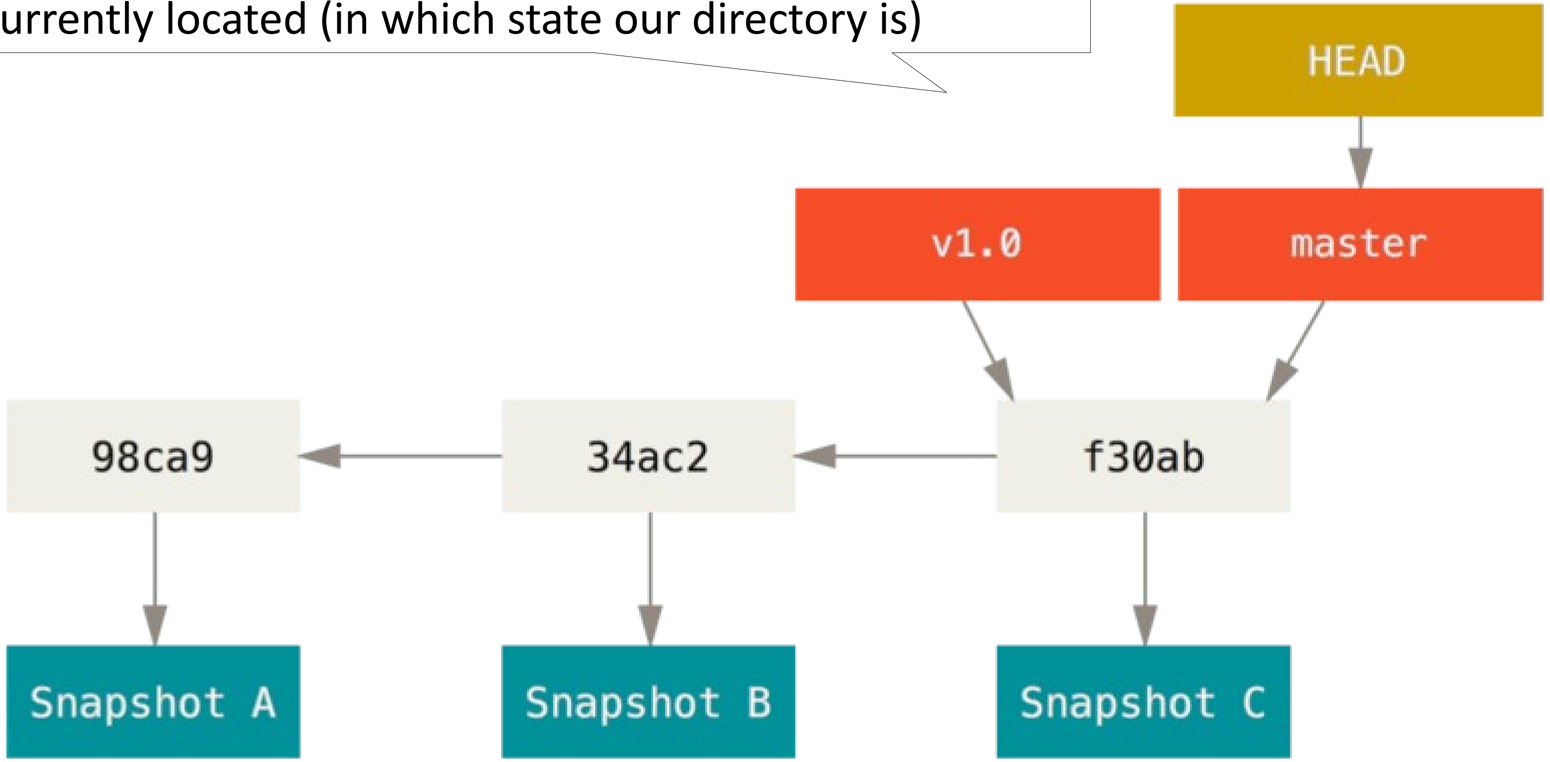
Branch = pointer to the last commit of a sequence.

Automatically advances as new commits are made.

https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

Branches in a nutshell
A branch and its commits history

A sequence of commits,
- A label pointer
- A branch pointer
- The HEAD pointer indicates on which object we are currently located (in which state our directory is)

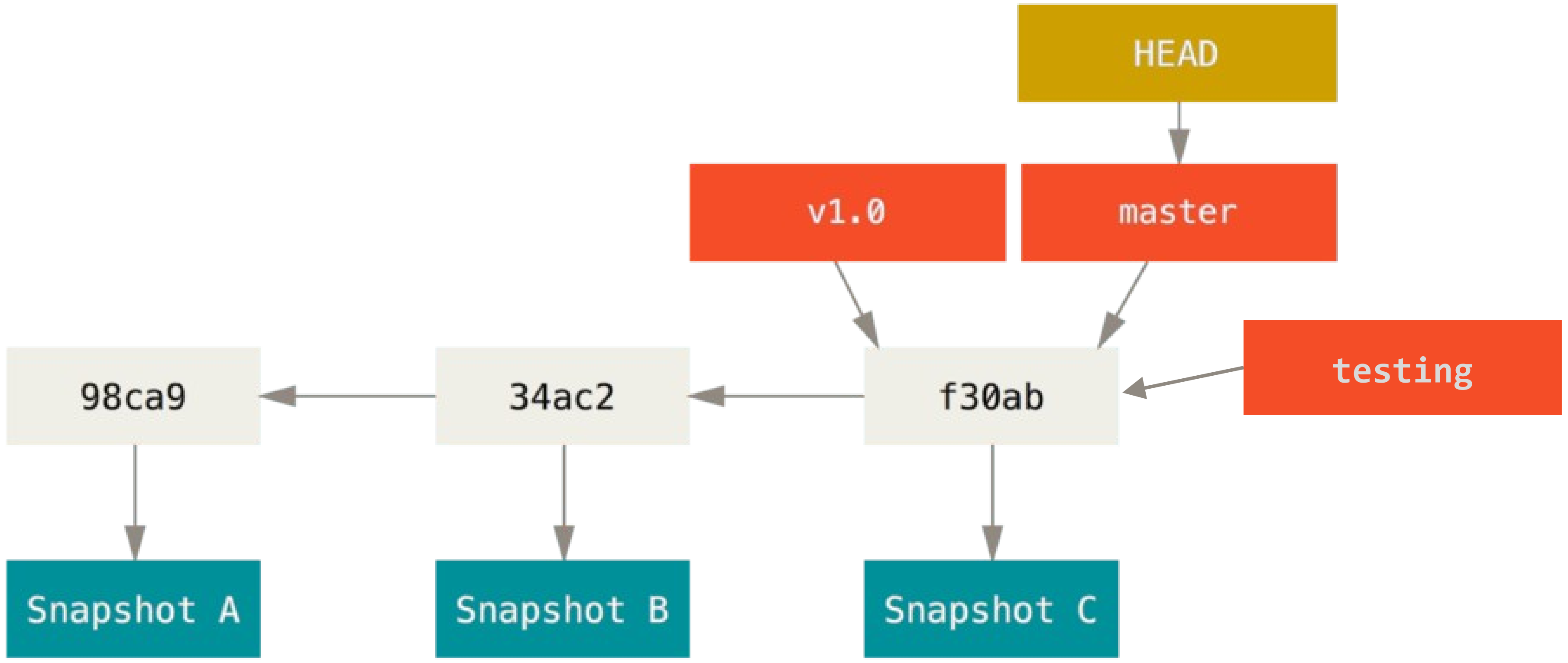


https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

Branches in a nutshell
Create a new branch

```
$ git branch testing
```

Two branches now point to the same set of *commits*

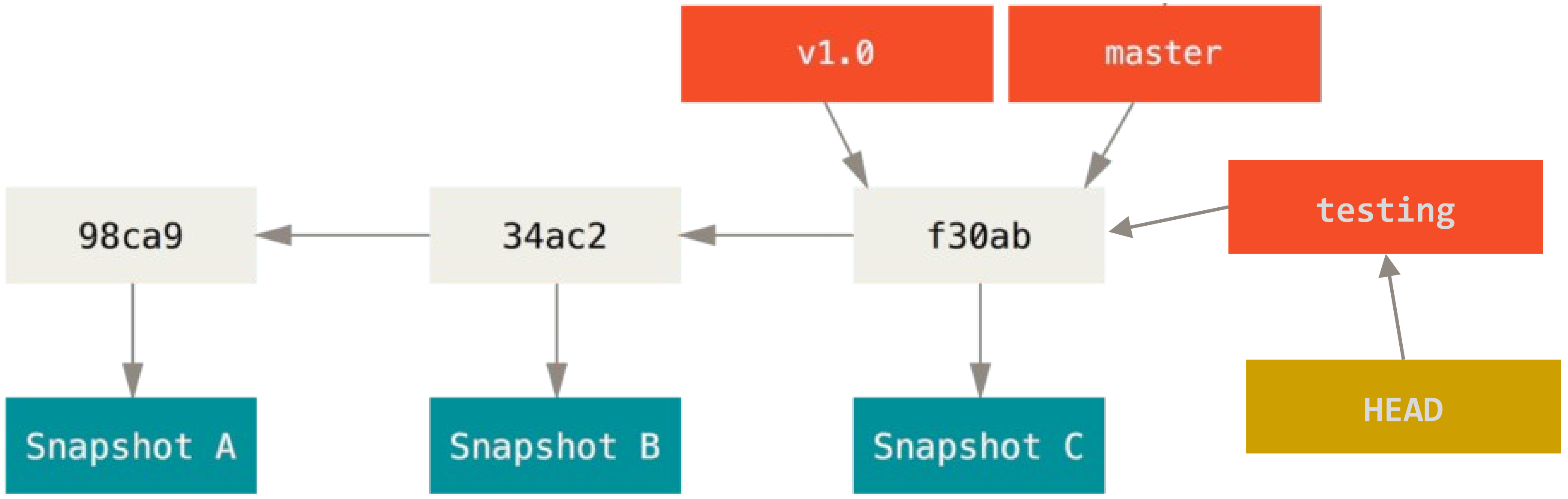


Branches in a nutshell

Switching between branches

```
$ git checkout testing
```

HEAD now points to the `testing` branch



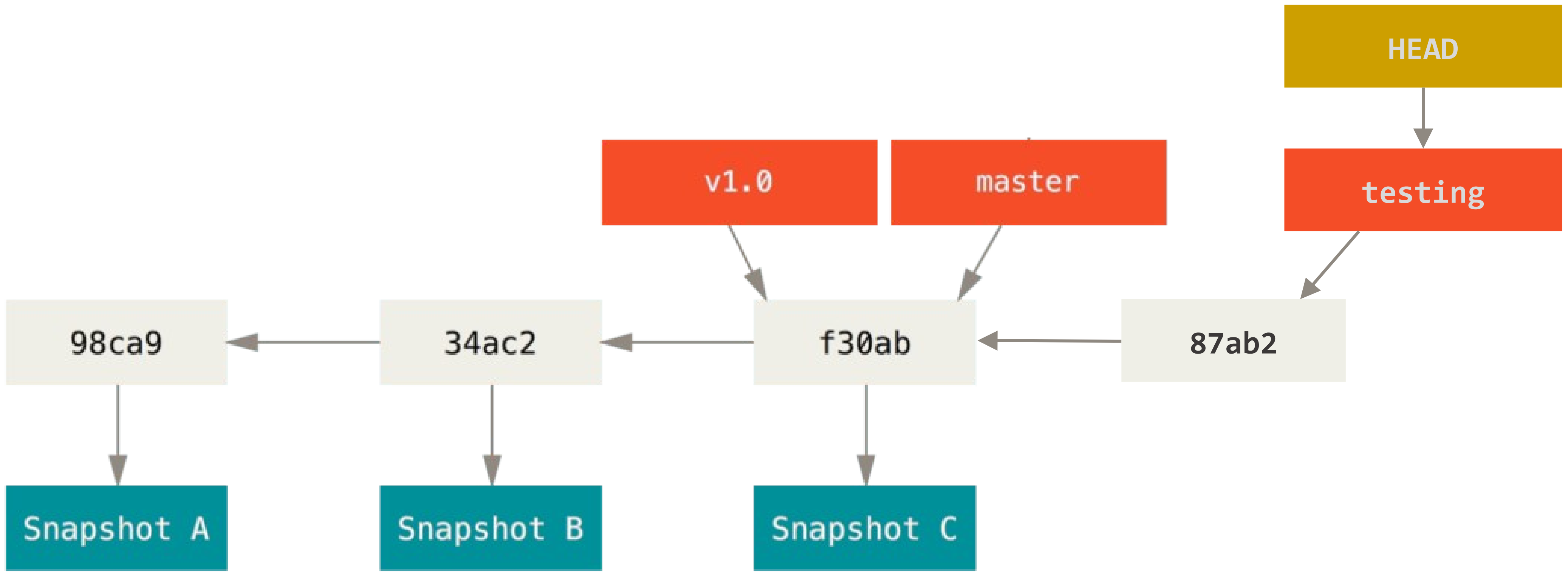
https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

Branches in a nutshell

Moving the HEAD

```
$ vim test.rb
$ git commit -a -m 'made a change'
```

The testing pointer advances with each commit

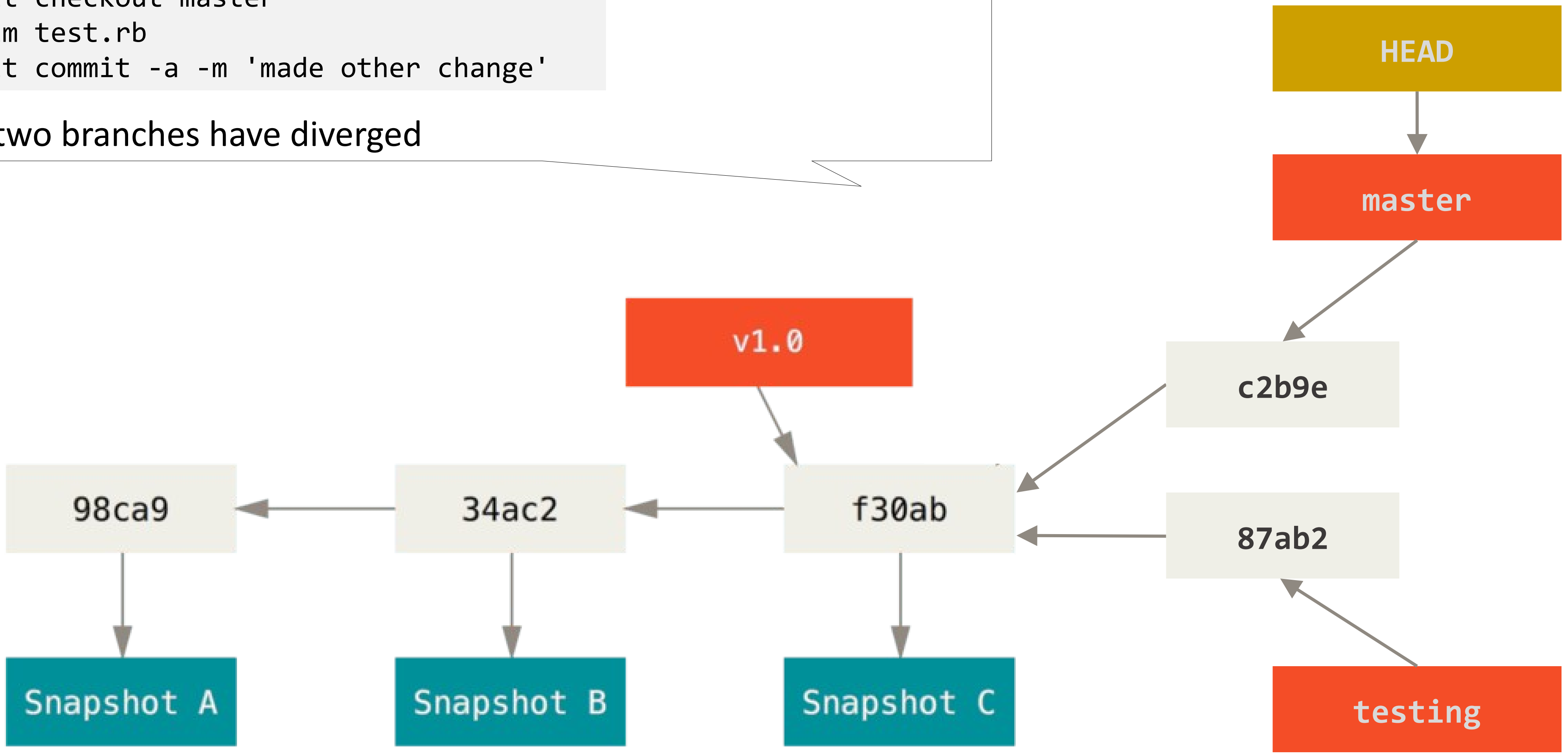


Branches in a nutshell

Divergent history

```
$ git checkout master
$ vim test.rb
$ git commit -a -m 'made other change'
```

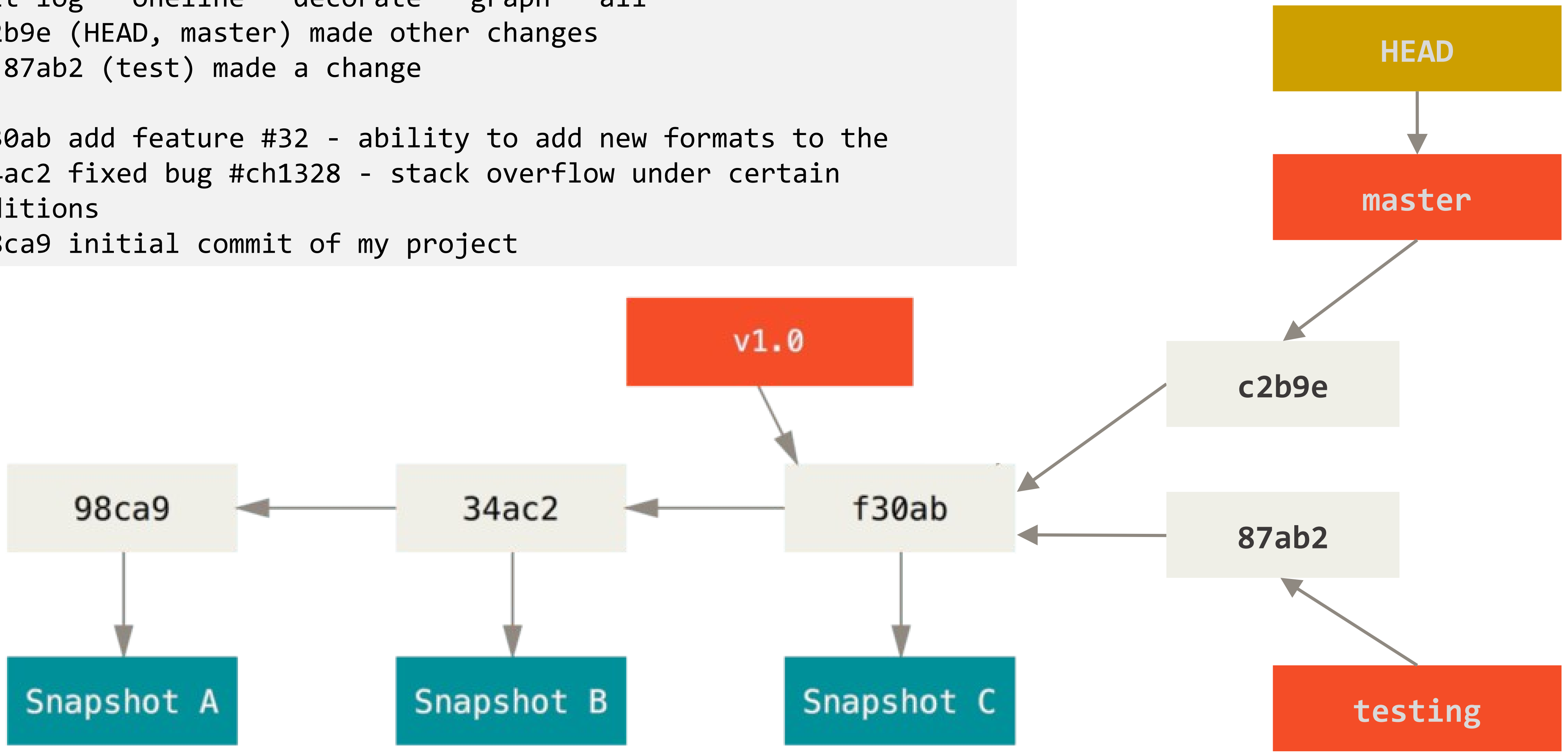
The two branches have diverged



Branches in a nutshell

Divergent history

```
$ git log --oneline --decorate --graph --all
* c2b9e (HEAD, master) made other changes
| * 87ab2 (test) made a change
|/
* f30ab add feature #32 - ability to add new formats to the
* 34ac2 fixed bug #ch1328 - stack overflow under certain
conditions
* 98ca9 initial commit of my project
```



Technological foundations of software development

Manage your source code

Part 4 – Source code management platforms

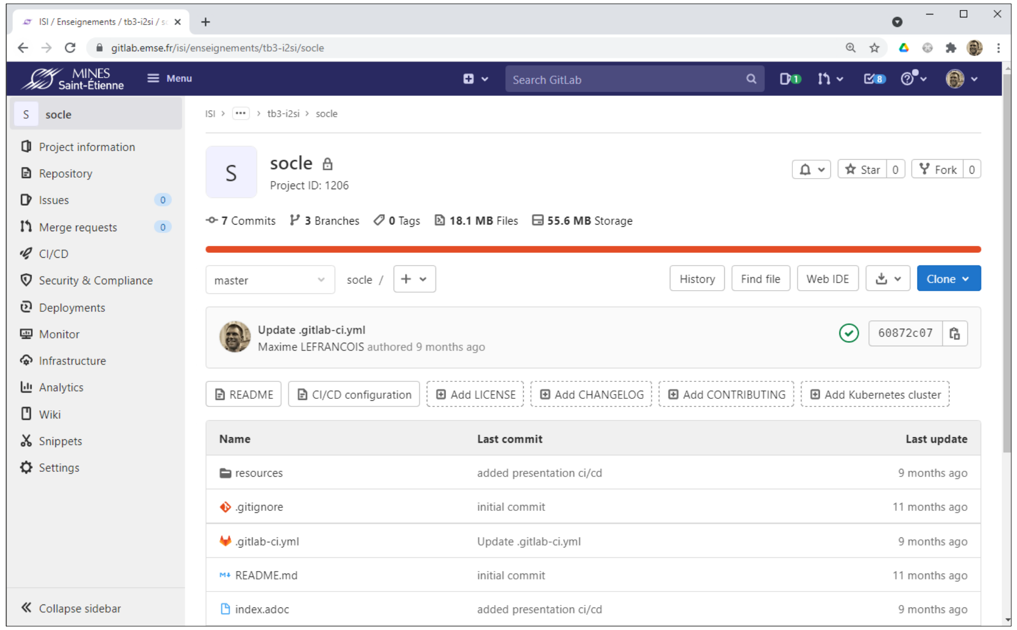
https://gitlab.emse.fr/
https://gitlab.com/

https://github.com

ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

In addition to Git…

- ✓ Rights management
- ✓ Ticket management, Kanban
- ✓ Merge Requests / Pull Requests
- ✓ Integration and continuous deployment (e.g. github pages)
- ✓ Wiki
- ✓ Analytics
- ✓ Integration with other applications,
- ✓ Social network for developers, and opensource resume

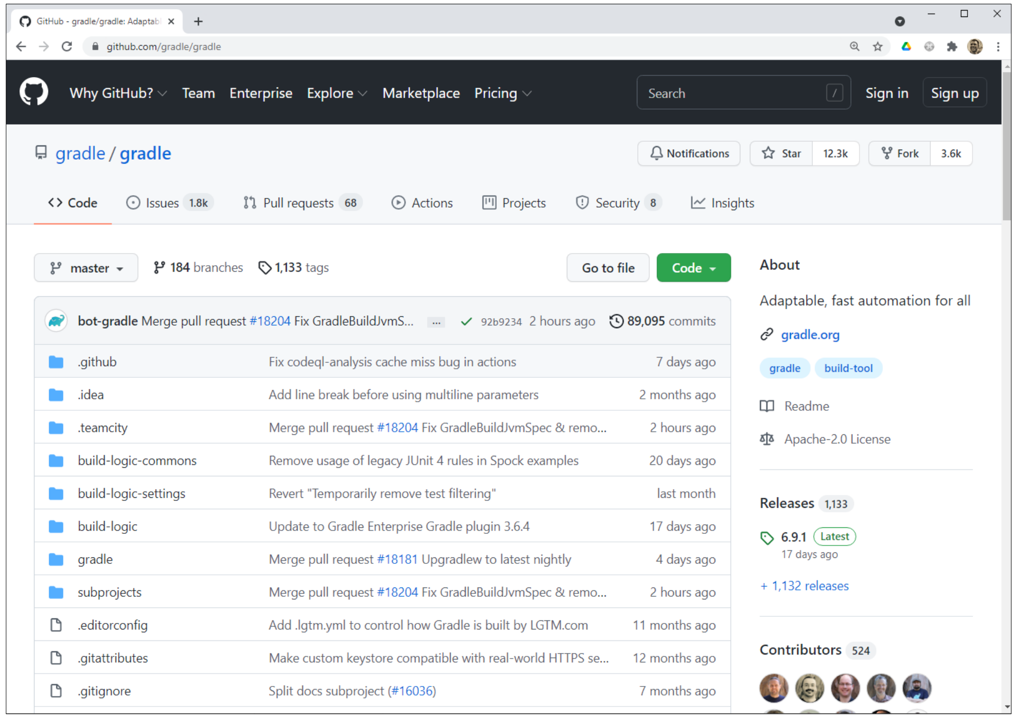

ISI / Enseignements / tb3-i2si / s
gitlab.emse.fr/isi/enseignements/tb3-i2si/socle

socle
Project ID: 1206

7 Commits · 3 Branches · 0 Tags · 18.1 MB Files · 55.6 MB Storage

Update .gitlab-ci.yml
Maxime LEFRANCOIS authored 9 months ago · 60872c07

Name	Last commit	Last update
resources	added presentation ci/cd	9 months ago
.gitignore	initial commit	11 months ago
.gitlab-ci.yml	Update .gitlab-ci.yml	9 months ago
README.md	initial commit	11 months ago
index.adoc	added presentation ci/cd	9 months ago


GitHub - gradle/gradle: Adaptab
github.com/gradle/gradle

gradle / gradle

master · 184 branches · 1,133 tags

bot-gradle Merge pull request #18204 Fix GradleBuildJvmS… ✓ 92b9234 2 hours ago · 89,095 commits

.github	Fix codeql-analysis cache miss bug in actions	7 days ago
.idea	Add line break before using multiline parameters	2 months ago
.teamcity	Merge pull request #18204 Fix GradleBuildJvmSpec & remo…	2 hours ago
build-logic-commons	Remove usage of legacy JUnit 4 rules in Spock examples	20 days ago
build-logic-settings	Revert "Temporarily remove test filtering"	last month
build-logic	Update to Gradle Enterprise Gradle plugin 3.6.4	17 days ago
gradle	Merge pull request #18181 Upgradlew to latest nightly	4 days ago
subprojects	Merge pull request #18204 Fix GradleBuildJvmSpec & remo…	2 hours ago
.editorconfig	Add .lgtm.yml to control how Gradle is built by LGTM.com	11 months ago
.gitattributes	Make custom keystore compatible with real-world HTTPS se…	12 months ago
.gitignore	Split docs subproject (#16036)	7 months ago

About
Adaptable, fast automation for all
gradle.org
gradle · build-tool
Readme
Apache-2.0 License
Releases 1,133
6.9.1 Latest
17 days ago
+ 1,132 releases
Contributors 524

https://git-scm.com/book/fr/v2/Les-branches-avec-Git-Branches-et-fusions%C2%A0%3A-les-bases

… References to deepen this course

Pro Git (2e edition), Scott Chacon and Ben Straub, Apress, 2014, 978-1-4842-0076-6
https://git-scm.com/book/en/v2

Git reference documentation https://git-scm.com/docs

Interactive Git cheat sheet http://ndpsoftware.com/git-cheatsheet.html

Gitlab basics: https://docs.gitlab.com/ee/gitlab-basics/
Gitlab docs: https://docs.gitlab.com/ sections *Agile with GitLab* et *Collaboration*

Github guides: https://guides.github.com/introduction/flow/
https://guides.github.com/activities/forking/
https://guides.github.com/activities/socialize/

… your turn

Complete the TODO section:

https://ci.mines-stetienne.fr/cps2/course/tfsd/course-2.html#_todos

Technological foundations of software development

Manage your source code

Part 2 – Git basics – complementary slides

ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

Record changes to the repository
Ignore files

- Ignore log files, automatically generated files, ...

```
$ cat .gitignore
*.log
*~
target/*
```

- standard shell file patterns (`*, [abc], ?, [0-9], **`)
- applied recursively in the working tree ;
- starts with '/': not recursive ;
- ends with a slash ('/'): directory ;
- starts with '!': include file despite other rules.

```
# pas de fichier .a
*.a

# mais suivre lib.a malgré la règle précédente
!lib.a

# ignorer uniquement le fichier TODO à la racine du projet
/TODO

# ignorer tous les fichiers dans le répertoire build
build/

# ignorer doc/notes.txt, mais pas doc/server/arch.txt
doc/*.txt

# ignorer tous les fichiers .txt sous le répertoire doc/
doc/**/*.txt
```

Record changes to the repository

Inspect indexed and non-indexed changes

```
$ git diff
diff --git a/CONTRIBUTING.md b/CONTRIBUTING.md
index 8ebb991..643e24f 100644
--- a/CONTRIBUTING.md
+++ b/CONTRIBUTING.md
@@ -65,7 +65,8 @@ branch directly, things can get messy.
 Please include a nice description of your changes when you submit your PR;
 if we have to read the whole diff to figure out why you're contributing
 in the first place, you're less likely to get feedback and have your change
-merged in.
+merged in. Also, split your changes into comprehensive chunks if you patch is
+longer than a dozen lines.

 If you are starting to work on a particular area, feel free to submit a PR
 that highlights your work in progress (and note in the PR title that it's
```

- git diff
- git diff --cached
- git difftool --tool-help

Record changes to the repository

Delete files

```
$ rm index.html
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

deletes the file in the directory only
need `git add index.html`
(or `git rm index.html`)

```
$ git rm index.html
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    index.html
```

deletes the file in the directory and in the index

```
$ git rm --cached big_file_should_be_in_gitignore.log
rm 'big_file_should_be_in_gitignore.log'
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        big_file_should_be_in_gitignore.log

nothing added to commit but untracked files present (use "git add" to track)
```

deletes a file in the index only

View the history of validations

```
$ git log
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    first commit
```

https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History

View the history of validations

```
$ git log
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    first commit
```

```
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 changed the version
number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit

$ git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 6 years ago : changed the version number
085bb3b - Scott Chacon, 6 years ago : removed unnecessary test
a11bef0 - Scott Chacon, 6 years ago : first commit
```

```
$ git log --pretty=format:"%h %s" --graph
* 2d3acf9 ignore errors from SIGCHLD on trap
*  5e3ee11 Merge branch 'master' of git://github.com/dustin/grit
|\
| * 420eac9 Added a method for getting the current branch.
* | 30e367c timeout code and tests
* | 5a09431 add timeout protection to grit
* | e1193f8 support for heads with slashes in them
|/
* d6016bc require time for xmlschema
*  11d191e Merge branch 'defunkt' into local
```

```
$ git log -p -2
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,7 +5,7 @@ require 'rake/gempackagetask'
 spec = Gem::Specification.new do |s|
     s.platform  =   Gem::Platform::RUBY
     s.name      =   "simplegit"
-    s.version   =   "0.1.0"
+    s.version   =   "0.1.1"
     s.author    =   "Scott Chacon"
     s.email     =   "schacon@gee-mail.com"
     s.summary   =   "A simple gem for using Git in Ruby
code."

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test

diff --git a/lib/simplegit.rb b/lib/simplegit.rb
index a0a60ae..47c6340 100644
--- a/lib/simplegit.rb
+++ b/lib/simplegit.rb
@@ -18,8 +18,3 @@ class SimpleGit
     end

 end
-
-if $0 == __FILE__
-  git = SimpleGit.new
-  puts git.show
-end
\ No newline at end of file
```

```
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

 Rakefile | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test

 lib/simplegit.rb | 5 -----
 1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    first commit

 README           |  6 ++++++
 Rakefile         | 23 +++++++++++++++++++++++
 lib/simplegit.rb | 25 +++++++++++++++++++++++++
 3 files changed, 54 insertions(+)
```

https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History

View the history of validations

Table 1. Useful options for git log --pretty=format

Option	**Description of formatting**
%H	Commit checksum
%h	Abbreviated commit checksum
%T	Tree checksum
%t	Tree abbreviated checksum
%P	Parent checksums
%p	Abbreviated parent checksums
%an	Author's name
%ae	Author's e-mail
%ad	Author's date (format of -date= <option>)
%ar	Author's relative date
%cn	Validator name
%ce	Validator's e-mail
%cd	Validator date
%cr	Validator relative date
%s	Subject

Table 2. Common git log options

Option	Description
-p	Displays the patch applied by each commit--stat Displays the statistics of each file for each commit
--shortstat	Displays only modified/inserted/deleted lines from the -stat option
--name-only	Displays the list of files modified after the commit information
--name-status	Displays list of affected files with add/change/delete information
--abbrev-commit	Displays only the first few characters of the SHA-1 checksum
--relative-date	Displays the date in relative format (e.g. "2 weeks ago") instead of the full date format
--graph	Displays in ASCII characters the graph of branches and merges opposite the history
--pretty	Displays commits in an alternative format. Formats include oneline, short, full, fuller, and format (where you can specify your own format)
--oneline	Convenience option corresponding to --pretty=oneline --abbrev-commit

Table 3. Options for limiting git log output

Option Description	
-(n)	Displays only the last n commits
--since, --after	Limit the display to commits made after the specified date
--until, --before	Limit the display to commits made before the specified date
--author	Show only commits whose author field matches the string passed as argument
--committer	Show only commits whose validator field matches the string passed as argument
--grep	Show only commits whose validation message contains the string
-S	Show only commits whose add or remove contains the string

https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History

Cancel actions

```
$ git commit --amend
```

allows to **replace** the last commit with a new one.

```
$ git commit -m 'validation initiale'
$ git add fichier_oublie
$ git commit --amend
```

```
$ git add index.html
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html
$ git restore --staged index.html
```

git explains how to **de-index an already indexed file**

```
$ nano index.html
$ git status | grep discard
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
$ git restore index.html
```

git explains how to **undo changes in the working directory**

https://git-scm.com/book/en/v2/Git-Basics-Undoing-Things

Working with remote repositories

```
$ git clone https://github.com/schacon/ticgit > /dev/null
$ cd ticgit && git remote
origin
$ git remote -v
origin     https://github.com/schacon/ticgit (fetch)
origin     https://github.com/schacon/ticgit (push)
```

when cloning a repository, it is named `origin` by default

we can then pull/push the contributions from this repository

https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes

Working with remote repositories

```
$ git clone https://github.com/schacon/ticgit > /dev/null
$ cd ticgit && git remote
origin
$ git remote –v
origin     https://github.com/schacon/ticgit (fetch)
origin     https://github.com/schacon/ticgit (push)
```

when cloning a repository, it is named `origin` by default

we can then pull/push the contributions from this repository

```
$ git remote -h
usage: git remote [-v | --verbose]
   or: git remote add [-t <branch>] [-m <master>] [-f] [--
tags | --no-tags] [--mirror=<fetch|push>] <name> <url>
   or: git remote rename <old> <new>
   or: git remote remove <name>
   or: git remote set-head <name> (-a | --auto | -d | --
delete | <branch>)
   or: git remote [-v | --verbose] show [-n] <name>
   or: git remote prune [-n | --dry-run] <name>
   or: git remote [-v | --verbose] update [-p | --prune]
[(<group> | <remote>)...]
   or: git remote set-branches [--add] <name> <branch>...
   or: git remote get-url [--push] [--all] <name>
   or: git remote set-url [--push] <name> <newurl> [<oldurl>]
   or: git remote set-url --add <name> <newurl>
   or: git remote set-url --delete <name> <url>
```

with `git remote` one can manage remote repositories:
- list remote repositories
- their names, their fetch/push urls,
- the tracked branches
- the default branch of the remote
- …

https://git-scm.com/book/en/v2/Git-Basics-Working-with-Remotes

Working with remote repositories

```
$ git remote add pb https://github.com/paulboone/ticgit
$ git fetch pb
remote: Counting objects: 43, done.
remote: Compressing objects: 100% (36/36), done.
remote: Total 43 (delta 10), reused 31 (delta 5)
Unpacking objects: 100% (43/43), fait.
From https://github.com/paulboone/ticgit
 * [new branch] master     -> pb/master
 * [new branch] ticgit     -> pb/ticgit
```

`git fetch`:

Retrieve all information from a remote repository

- the `master` branch from pb is now `pb/master`

✓ **No automatic merge, no local modification!**

```
$ git push pb ticgit
$ # man: git push <remote> <branch>
```

```
$ git push origin branch1:branch2
$ # man: git push <remote> <local-ref>:<remote-ref>
```

```
$ git push
```

`git push`:

Push to the remote repository

- ex 1: push the `ticgit` branch to pb
- ex 2: push `branch1` to the `branch2` branch of `origin`
- ex 3: with default values, equivalent to:
  `git push origin master`

⚠ **There may be conflicts during the merge !** ⚠

Tagging

- git allows you to tag states in the history

```
$ git tag
v0.1
v1.3
```

list tags

```
$ git tag –a v1.4 –m "Version 1.4"
$ git tag
v0.1
v1.3
v1.4
```

tag the current commit, with a tagging message

```
$ git tag -a v1.2 9fceb02
```

tag a specific commit

```
$ git push origin v1.2
```

push a tag
is not done by default during git push, except with the --tags option
`$ git push origin --tags`

```
$ git tag –d v1.4
$ git push origin –delete v1.4
```

deleting a tag
and deletion on the remote server

https://git-scm.com/book/en/v2/Git-Basics-Tagging

Tagging

```
$ git checkout v2.29.2
Note : basculement sur 'v2.29.2'.

You are in the "HEAD detached" state. You can visit,
make experimental modifications and and validate them.
You just need to make another switch to
abandon the commits you make in this state without
impacting the other branches

If you want to create a new branch to keep the commits
you create,
you just use the -c option of the switch command like
this:

  git switch -c <name-of-new-branch>

Or undo this operation with :

  git switch -

Disable this advice by setting the advice.detachedHead
configuration variable to false

HEAD is now on 898f80736c Git 2.29.2

$ git checkout v2.29.1
HEAD's previous position was on 898f80736c Git 2.29.2
HEAD is now on b927c80531 Git 2.29.1
```

in the "detached HEAD" state,
a new commit would not belong to any branch
it would be reachable only with its exact footprint
it is better to create a branch
for example:

```
$ git checkout -b v2.29.X
Basculement sur la nouvelle branche 'v2.29.X'
```

Technological foundations of software development

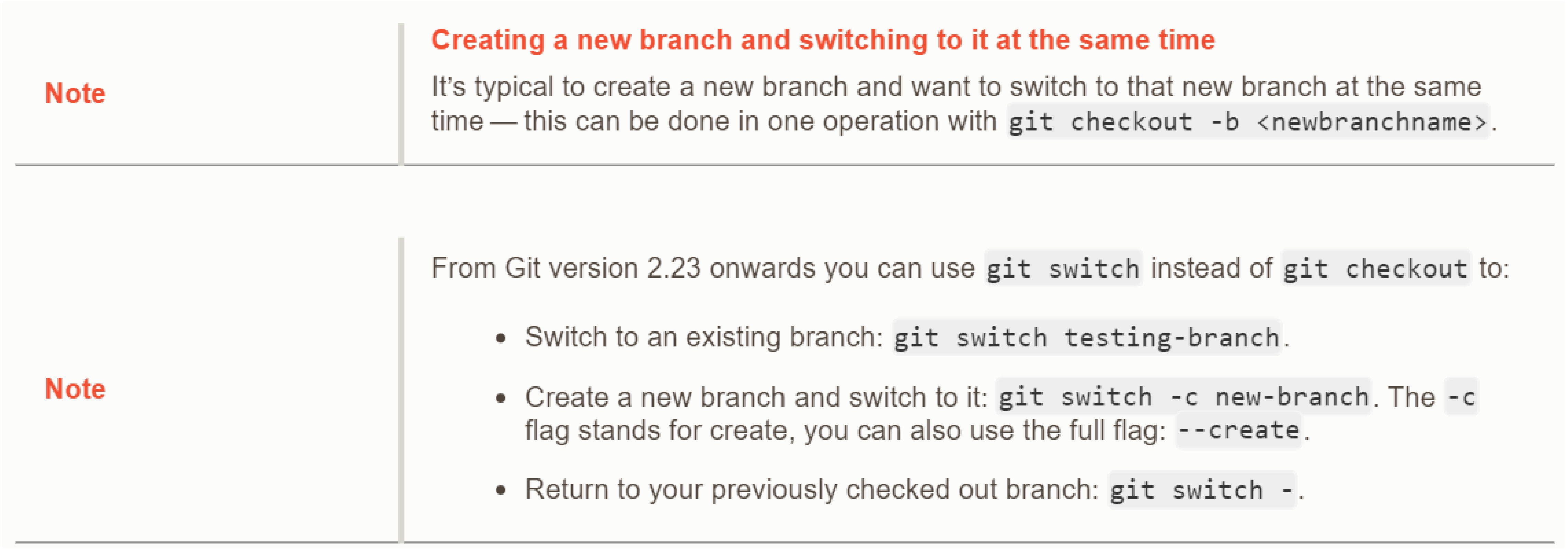
Manage your source code

Part 3 – Branching and merging with Git – complementary slides

ICM – Computer Science Major – Course unit on Technological foundations of computer science
M1 Cyber Physical and Social Systems – Course unit on CPS2 engineering and development, Part 2: Technological foundations of software development
Maxime Lefrançois https://maxime-lefrancois.info
online: https://ci.mines-stetienne.fr/cps2/course/tfsd/

Branches in a nutshell

Notes



Note

Creating a new branch and switching to it at the same time

It's typical to create a new branch and want to switch to that new branch at the same time — this can be done in one operation with `git checkout -b <newbranchname>`.

Note

From Git version 2.23 onwards you can use `git switch` instead of `git checkout` to:

- Switch to an existing branch: `git switch testing-branch`.
- Create a new branch and switch to it: `git switch -c new-branch`. The `-c` flag stands for create, you can also use the full flag: `--create`.
- Return to your previously checked out branch: `git switch -`.

https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

Branching and merging

Scenario

1. you are working on a web site;
2. you create a branch for a new article in progress;
3. you start working on this branch.

At this point, you get a call that a critical problem has been discovered and needs to be addressed as soon as possible.
So you do the following:

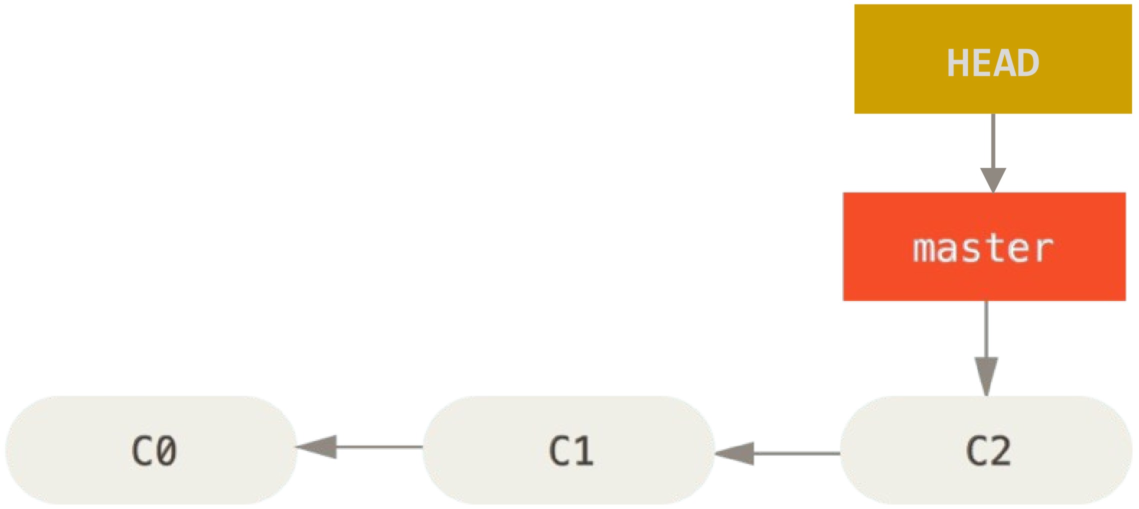
1. you switch to the production branch;
2. you create a branch to add the patch;
3. after testing it, you merge the patch branch and push the result to production;
4. you switch back to the initial branch and continue your work

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging

Scenario

1. you are working on a web site;

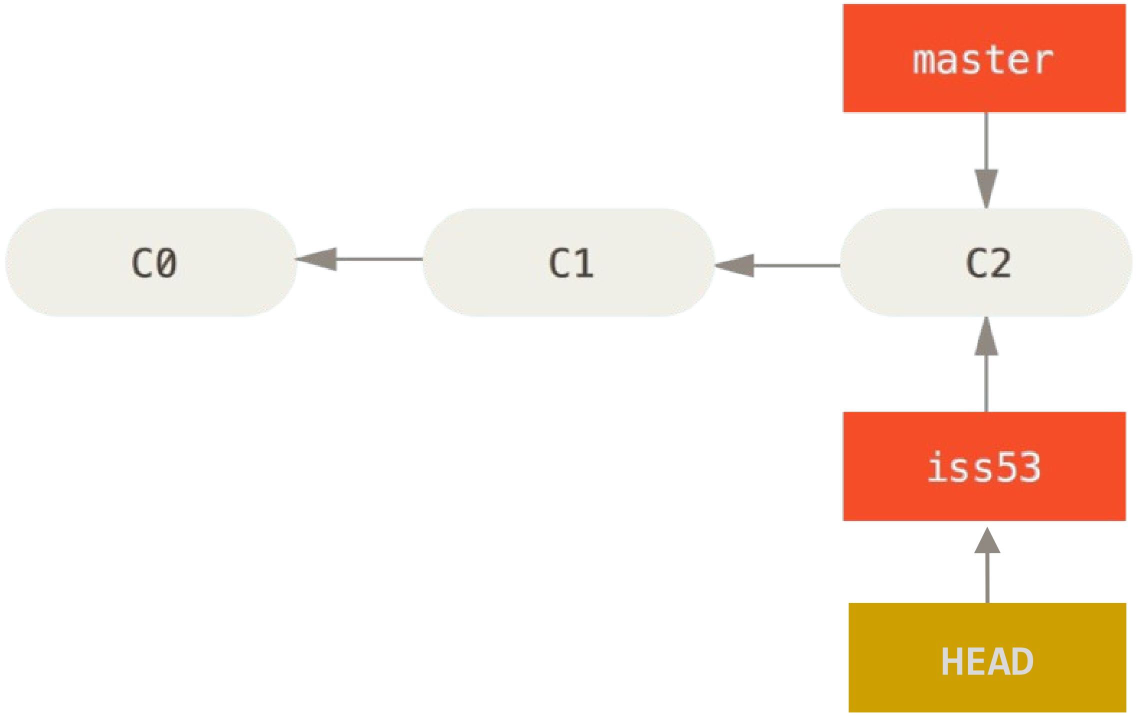


https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging

Scenario

1. you are working on a web site;
2. you create a branch for a new article in progress;



Creation of a new branch `iss53` to work on issue #53

```
$ git checkout -b iss53
Switched to a new branch "iss53"
```

```
$ git branch iss53
$ git checkout iss53
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

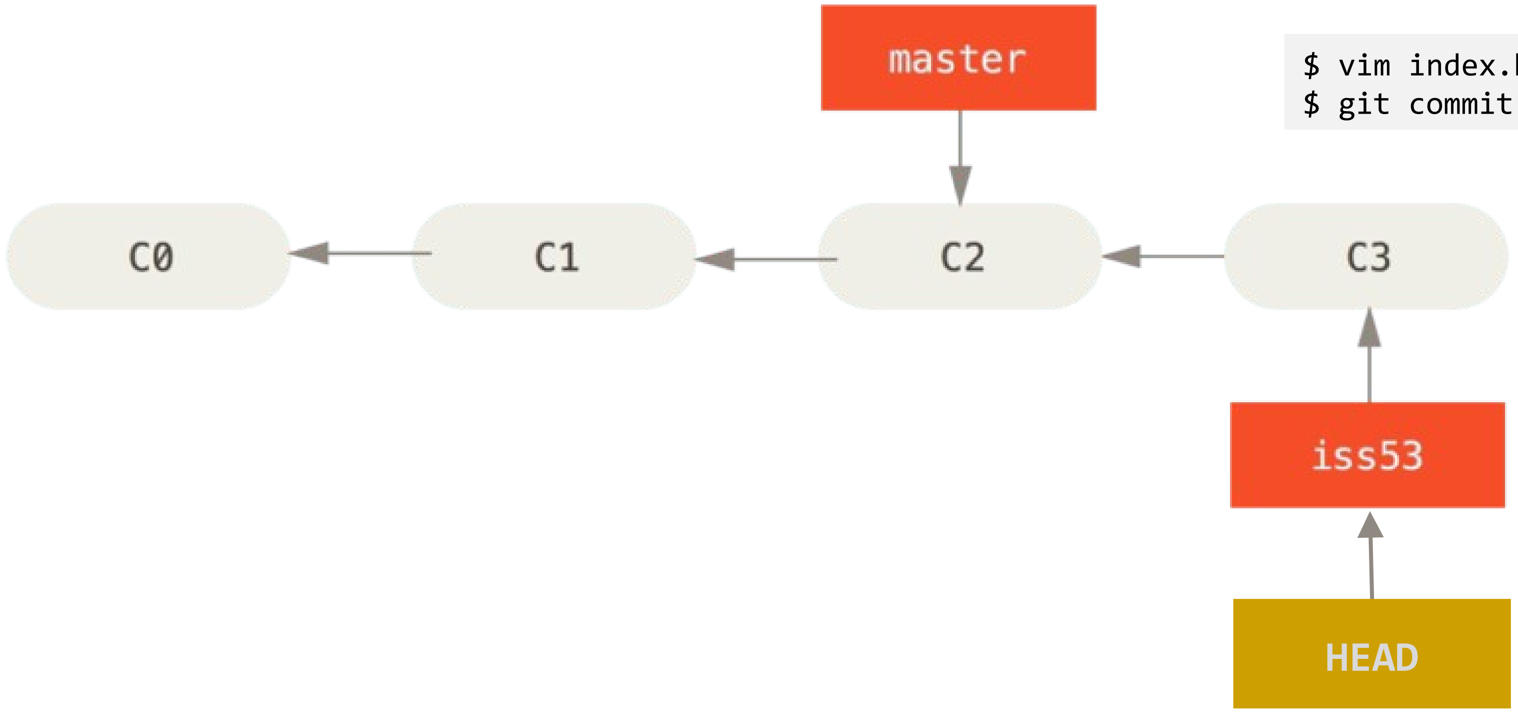
Branching and merging

Scenario

1. you are working on a web site;
2. you create a branch for a new article in progress;
3. you start working on this branch.

Example of a new commit on branch `iss53`

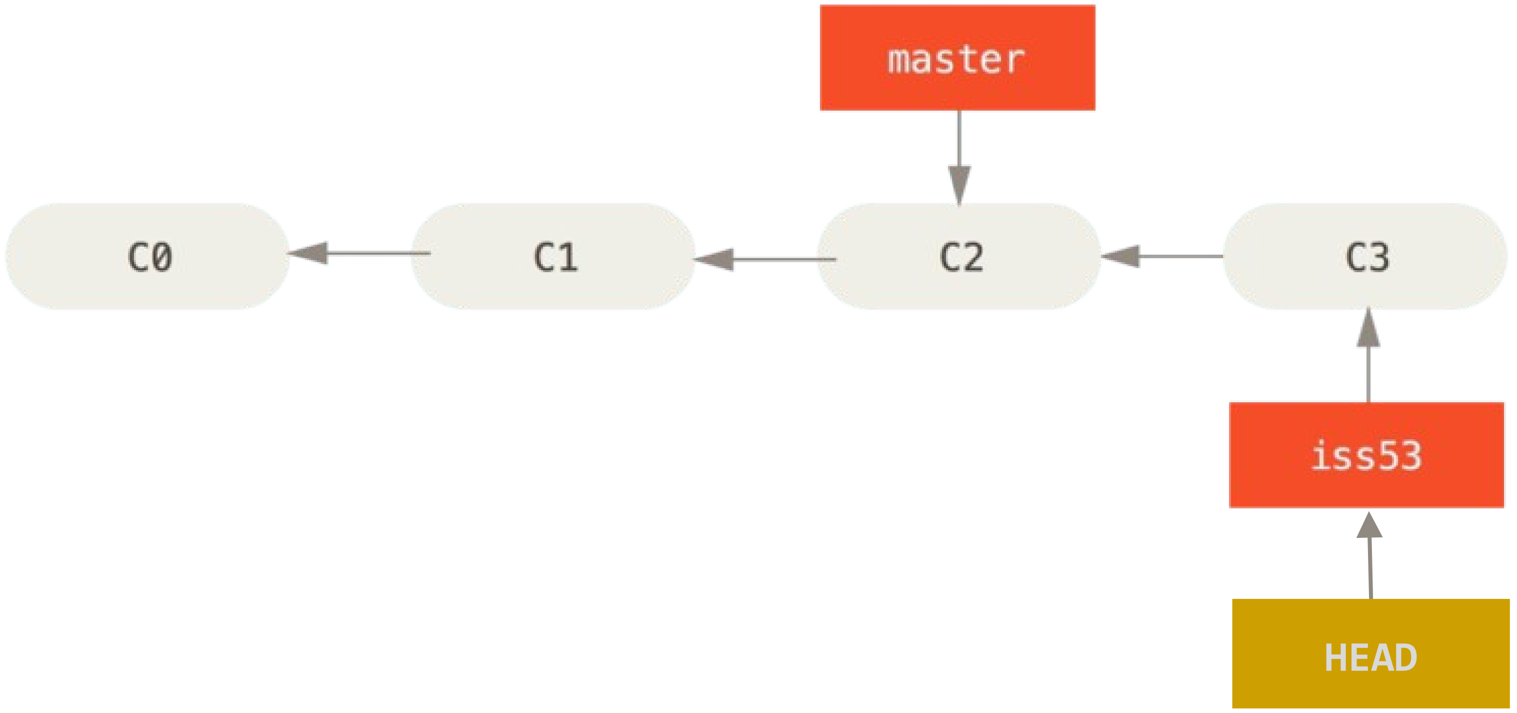
```
$ vim index.html
$ git commit -a -m "add footer [issue 53]"
```



Branching and merging

Scenario

At this point, you get a call that a critical problem has been discovered and needs to be addressed as soon as possible.

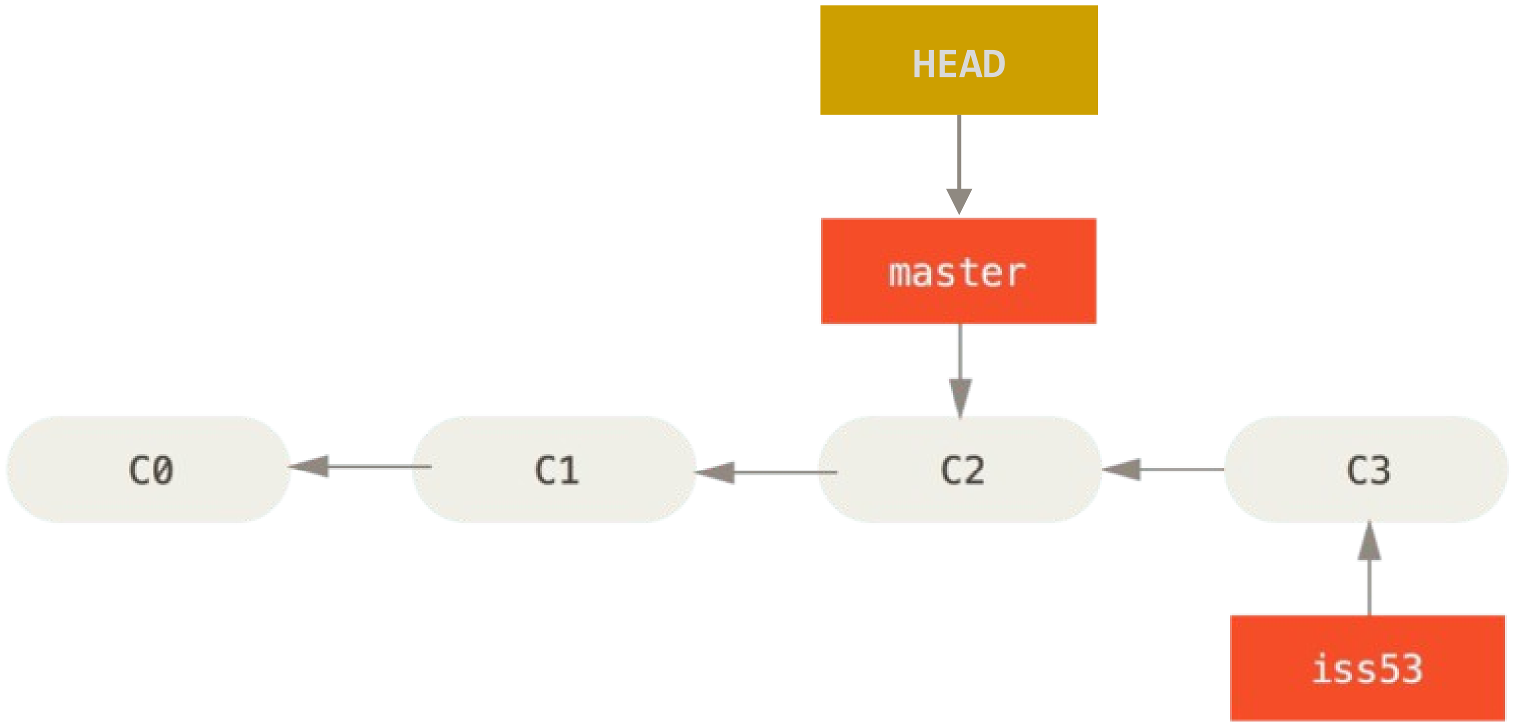


Branching and merging

Scenario

1. you switch to the production branch;

```
$ git checkout master
Switched to branch 'master'
```

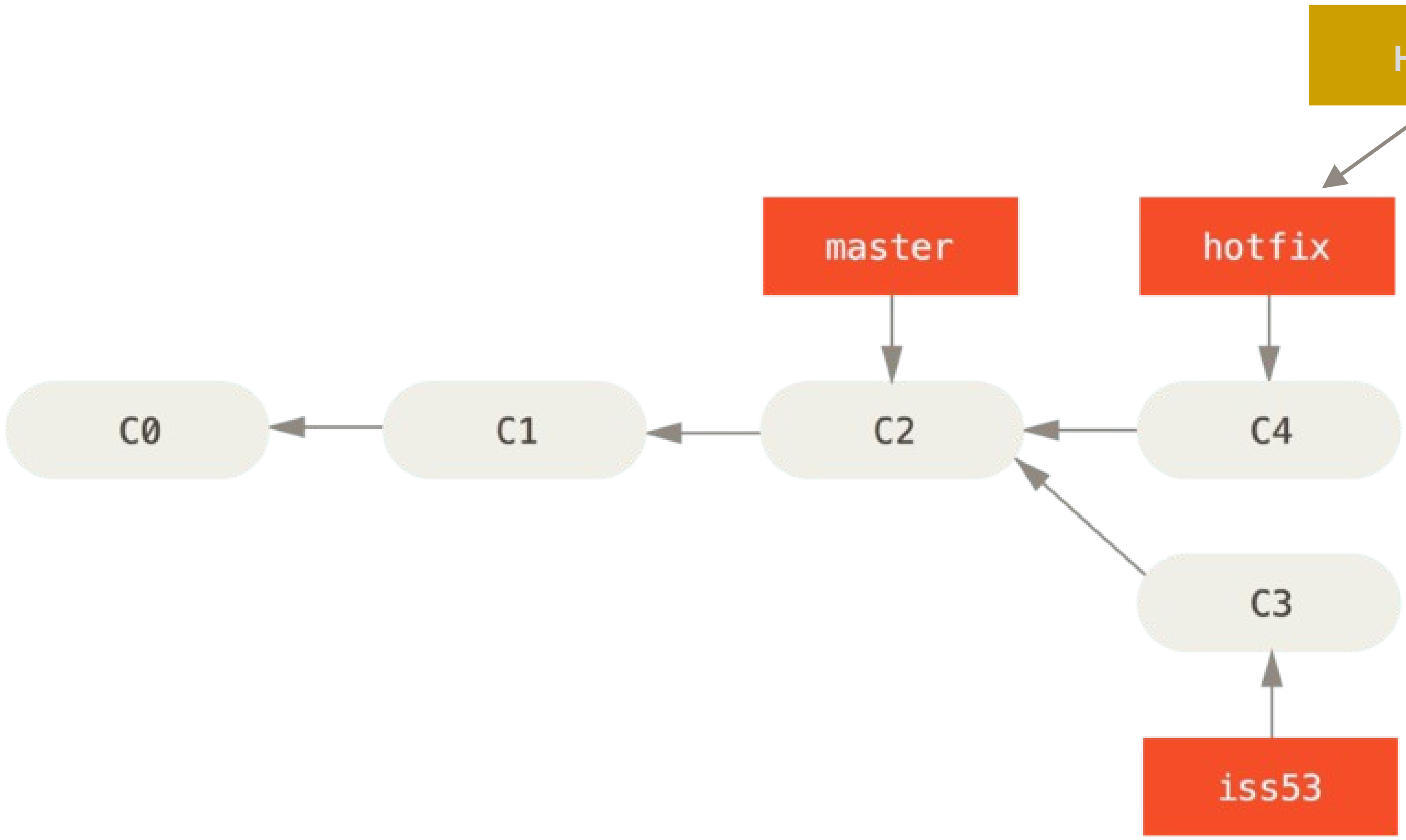


https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging

Scenario

2. you create a branch to add the patch;

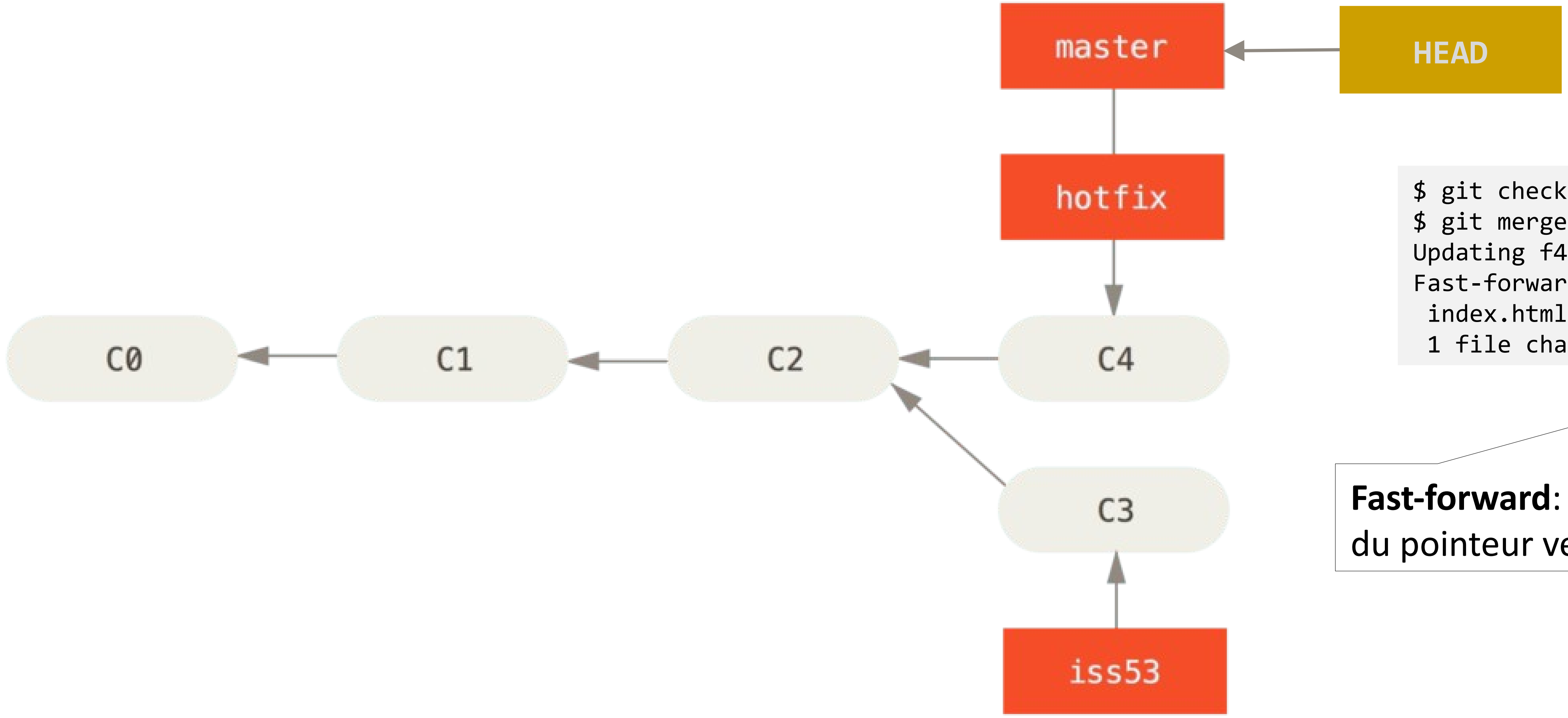


```
$ git checkout -b patch
Switched to a new branch patch'
$ vim index.html
$ git commit -a -m "incorrect email address"
[correctif 1fb7853] "incorrect email address"
 1 file changed, 2 insertions(+)
```

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging
Scenario

3. after testing it, you merge the patch branch and push the result to production;



```
$ git checkout master
$ git merge patch
Updating f42c576..3a0874c
Fast-forward
 index.html | 2 ++
 1 file changed, 2 insertions(+)
```

Fast-forward: simple déplacement du pointeur vers l’avant

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

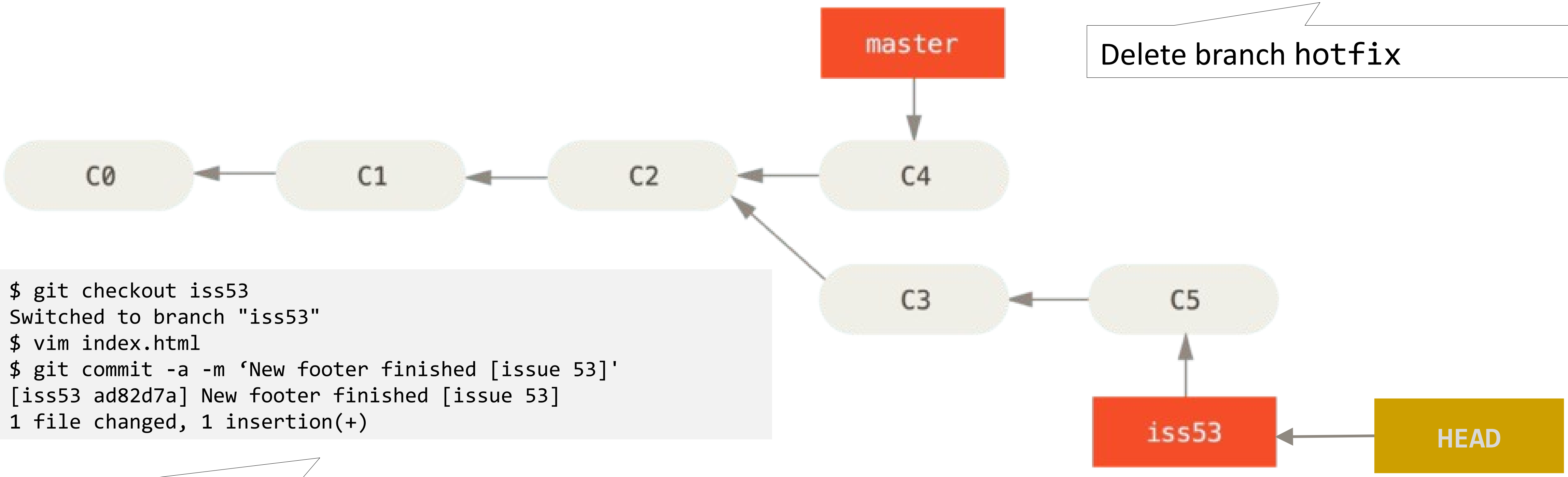
Branching and merging

Scenario

4. you switch back to the initial branch and continue your work

```
$ git branch -d patch
Deleted branch patch (3a0874c).
```

Delete branch hotfix

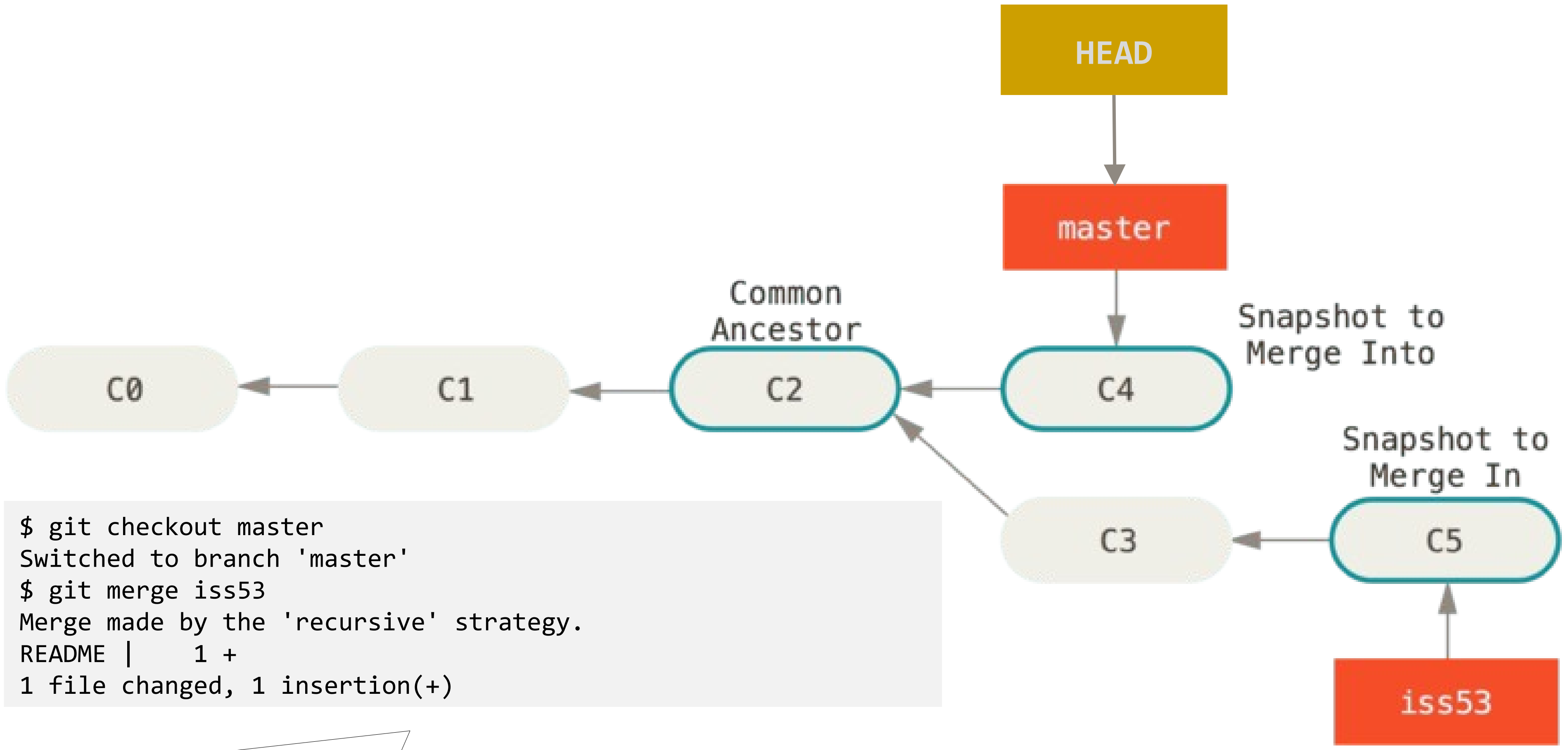


```
$ git checkout iss53
Switched to branch "iss53"
$ vim index.html
$ git commit -a -m 'New footer finished [issue 53]'
[iss53 ad82d7a] New footer finished [issue 53]
1 file changed, 1 insertion(+)
```

Switch to branch iss53 and create new commits

Branching and merging

Merging



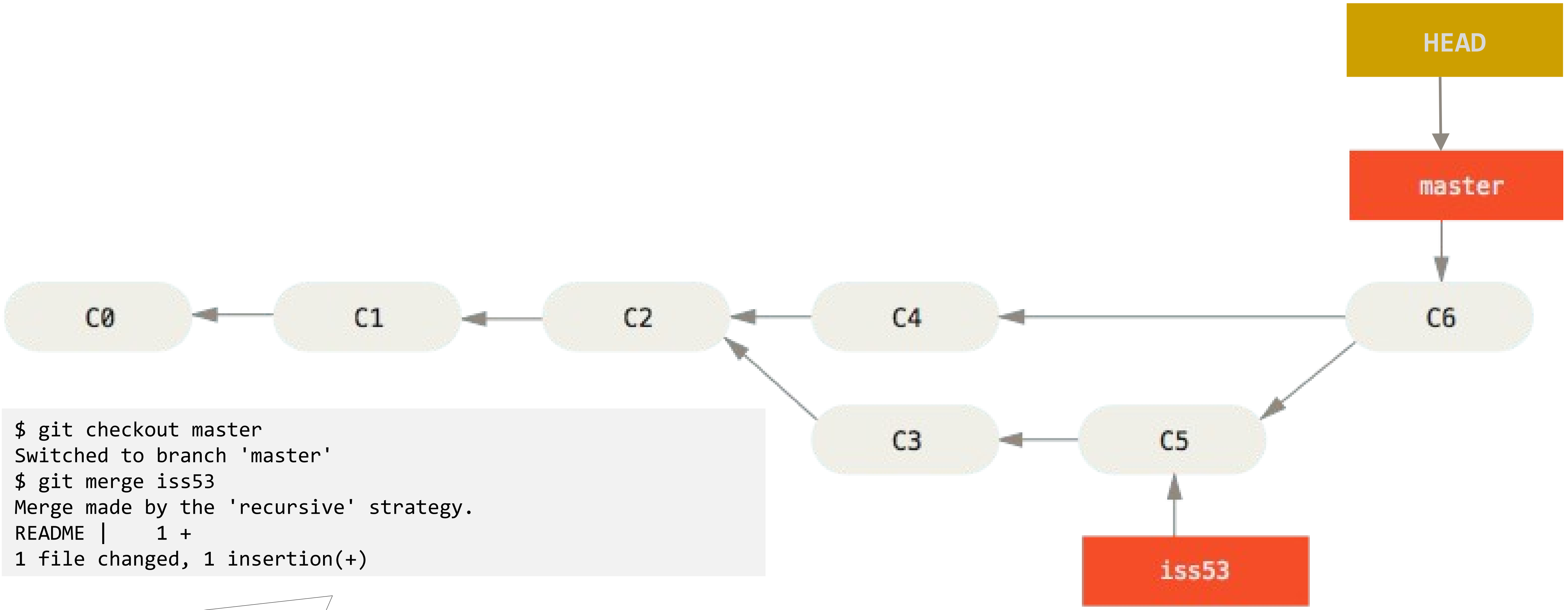
```
$ git checkout master
Switched to branch 'master'
$ git merge iss53
Merge made by the 'recursive' strategy.
README |    1 +
1 file changed, 1 insertion(+)
```

‘recusive’ strategy: Simple 3-sources fusion

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging

Merging



```
$ git checkout master
Switched to branch 'master'
$ git merge iss53
Merge made by the 'recursive' strategy.
README |    1 +
1 file changed, 1 insertion(+)
```

‘recusive’ strategy: Simple 3-sources fusion

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging
Merge conflicts

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")

Unmerged paths:
  (use "git add <file>..." to mark resolution)

    both modified:      index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

Example of a merger conflict to be resolved

```
<<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
======
<div id="footer">
 please contact us at support@github.com
</div>
>>>>>>> iss53:index.html
```

Contents of the `index.html` file with conflict markers

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Branching and merging

Merge conflicts

```
<<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
======
<div id="footer">
 please contact us at support@github.com
</div>
>>>>>>> iss53:index.html
```

```
<div id="footer">
please contact us at email.support@github.com
</div>
```

Example of manual resolution

```
$ git add index.html
```

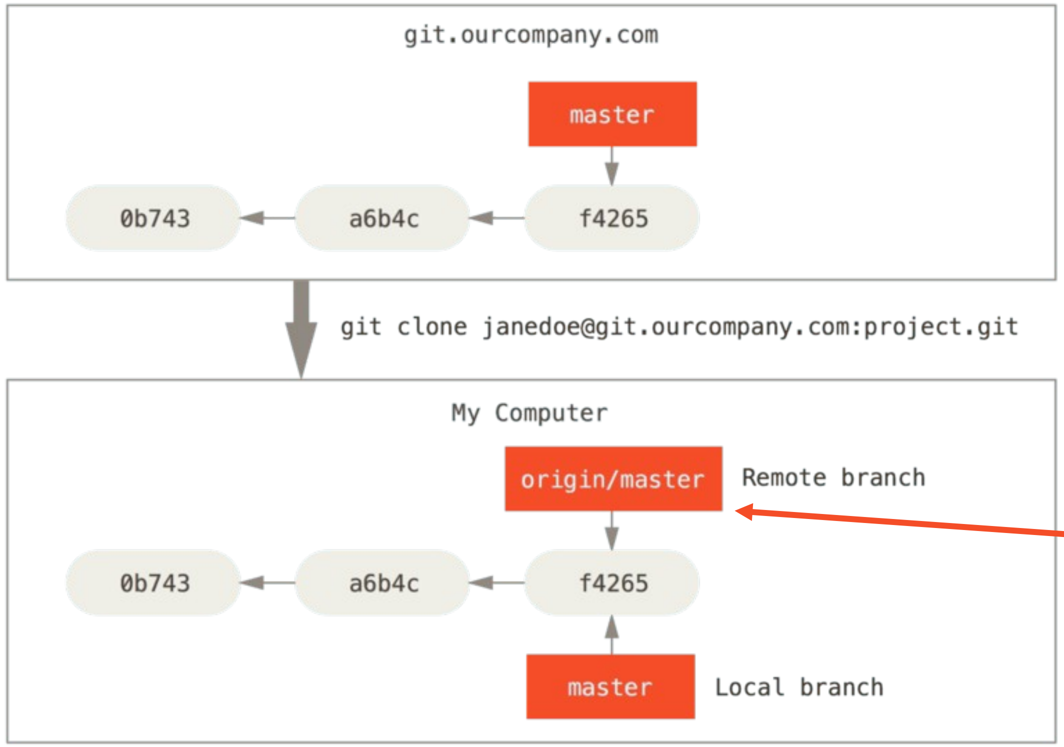
We mark this conflict "resolved".

```
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:

    modified:   index.html
```

```
$ git commit
```

We finalize the commit

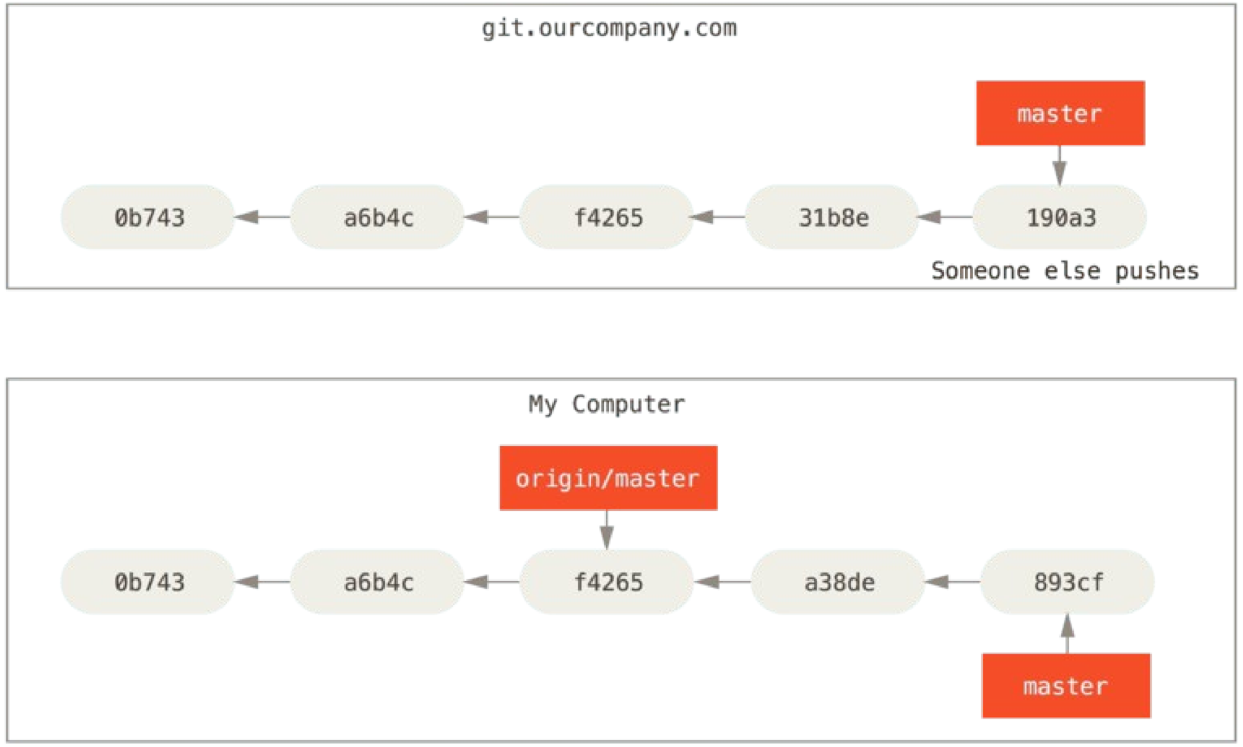
Remote branches



The <remote>/<branch> branch is unmodifiable. It is a bookmark to indicate the state of the remote branch

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Remote branches



Local and remote work may differ

Remote branches

```
$ git push origin issue-53
Counting objects: 24, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (24/24), 1.91 KiB | 0 bytes/s, done.
Total 24 (delta 2), reused 0 (delta 0)
To https://github.com/schacon/simplegit
 * [new branch]      issue-53 -> issue-53
```

I push my changes on the remote server

```
$ git fetch origin
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://github.com/schacon/simplegit
 * [new branch]      issue-53 -> origin/issue-53
```

My colleague retrieves the changes

```
$ git checkout -b issue-53 origin/issue-53
Branch issue-53 set up to track remote branch issue-53 from origin.
Switched to a new branch 'issue-53'
```

My colleague creates a modifiable local branch issue-53, based on the state of origin/issue-53

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Remote branches

```
$ git checkout -b issue-53 origin/issue-53
Branch issue-53 set up to track remote branch issue-53 from origin.
Switched to a new branch 'issue-53'
```

Create issue-53 that "follows" origin/issue-53

Allows you to push and pull from origin/issue-53 by default

```
$ git checkout --track origin/issue-53
Branch issue-53 set up to track remote branch issue-53 from origin.
Switched to a new branch 'issue-53'
```

Shortcut: automatic naming of the created branch

```
$ git checkout issue-53
Branch issue-53 set up to track remote branch issue-53 from origin.
Switched to a new branch 'issue-53'
```

Shortcut: if issue-53 does not exist, and exists on a single remote

https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Remote branches

```
$ git branch -vv
  iss53     7e424c3 [origin/iss53: ahead 2] forgot the brackets
  master    1ae2a45 [origin/master] deploying index fix
* correctionserveur f8674d9 [equipe1/correction-serveur-ok: ahead 3, behind 1] this should do it
  test   5ea463a trying something new
```

Visualize the branches and the branches they are configured to follow

```
$ git pull issue-53
```

Shortcut for `git fetch` then `git merge`