

git introduction

Andrey Sapronov, SANC, JINR

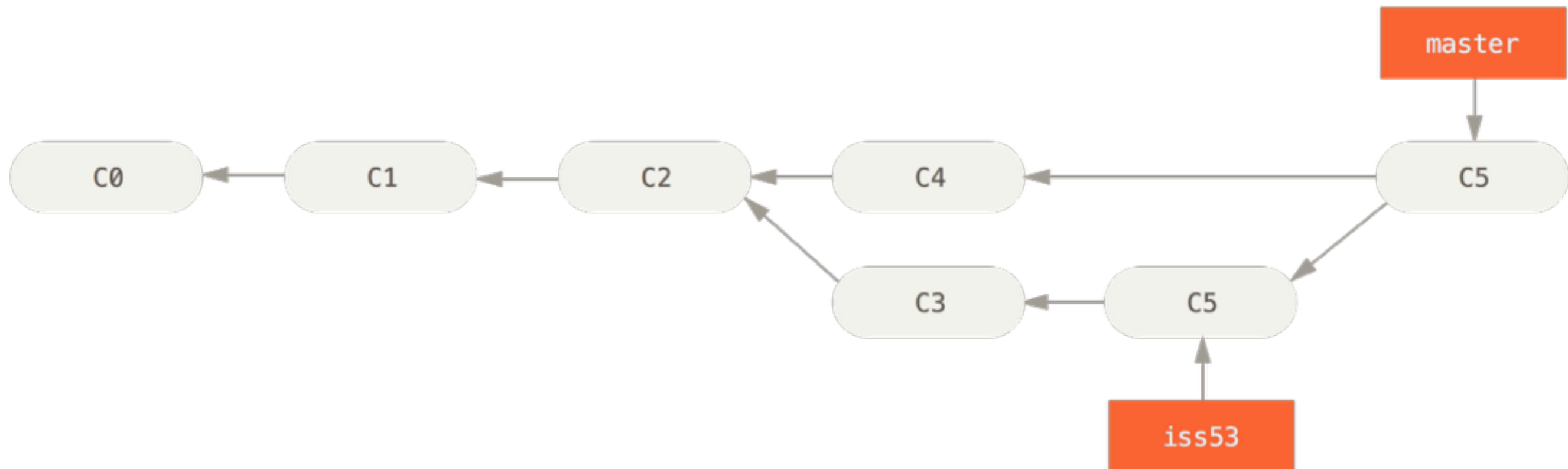
xFitter workshop in Dubna, 20 Feb 2016

git - what is it?

- Like svn, but better
- A versioning system which combines local and remote repositories.
- faster, mostly offline mode
- branching and merging is imperative
- atomic commits is one of the strategies
- has svn backport

what to know:

- in git you must branch
- any local work can be branched



- but branching can be global too (not today)

git - materials

- Tutorial, book:
<http://git-scm.com/docs/gittutorial>
<http://git-scm.com/book/>
- `$> git help <command>`

git CLI in brief

- Clone a project:

```
$> git clone git@git.jinr.ru:sanc/mcsanc.git
```

- This is your `_local_` version of repository.
- You can branch, merge, revert, commit here, etc.
- When you want others to see your efforts, push:

```
$> git push
```

git CLI in brief: basic edit

- Edit something. Look at the results:

```
$> git status
```

```
...
```

```
    modified:   share/input.cfg
```

```
Untracked files:
```

```
...
```

```
    log.
```

- Stage the changes.

```
$> git add share/input.cfg log
```

- When you are ready, commit them:

```
$> git commit -m 'I tried...'
```

git CLI in brief: basic branch

- If there are several things to do -> create a branch for each of them:

```
$> git checkout -b bad_idea
```

```
M      share/input.cfg
```

```
Switched to a new branch 'bad_idea'
```

- Oh, wait...

```
$> git checkout -b good_idea
```

```
...
```

```
$> git branch # list branches
```

```
  bad_idea
```

```
* good_idea
```

```
  master
```

- Now, if you edit, your edits will be in both ideas, but will stick to that where you “add and commit”:

```
$> git checkout bad_idea
```

```
$> git add
```

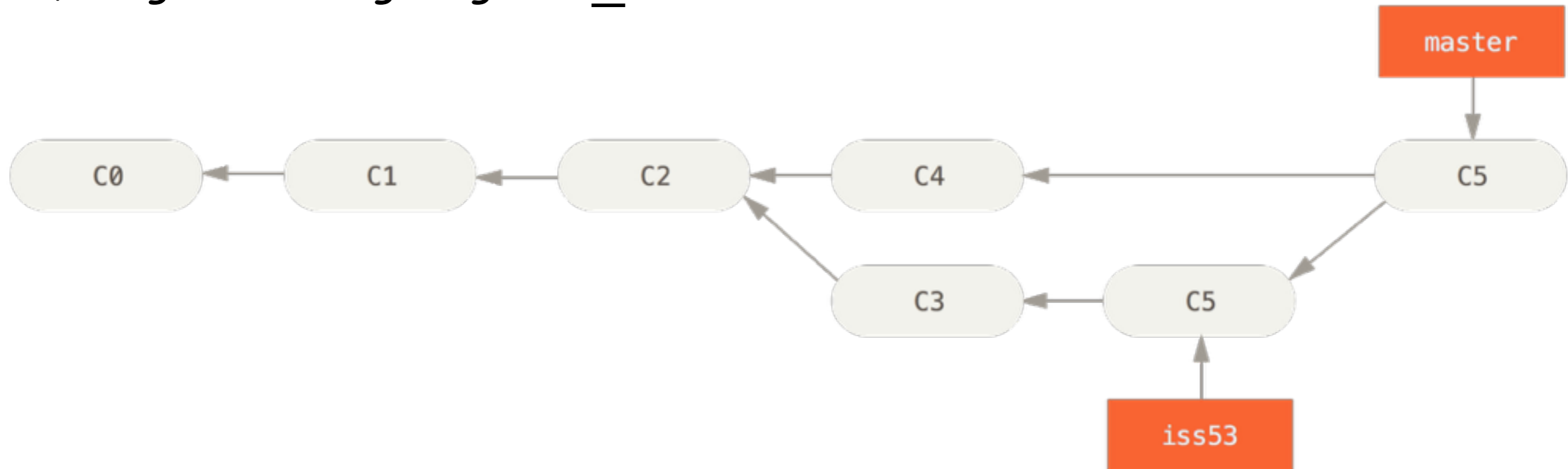
```
$> git commit ...
```

git CLI in brief: basic branch

- You test and polish your branch until you it blooms, and then you merge it to the **master** one:

```
$> git checkout master
```

```
$> git merge good_idea
```



- And continue with your other ideas

git CLI in brief: inspection

- See what are the changes in the current branch:

```
$> git diff
```

- See what happened before:

```
$> git log
```

```
74bc732 Adding .gitignore to ignore generated files  
4d1d55a Sergei's latest changes... I hope nothing breaks  
d3c37b5 fixed compiling without tinyxml by Volodya
```

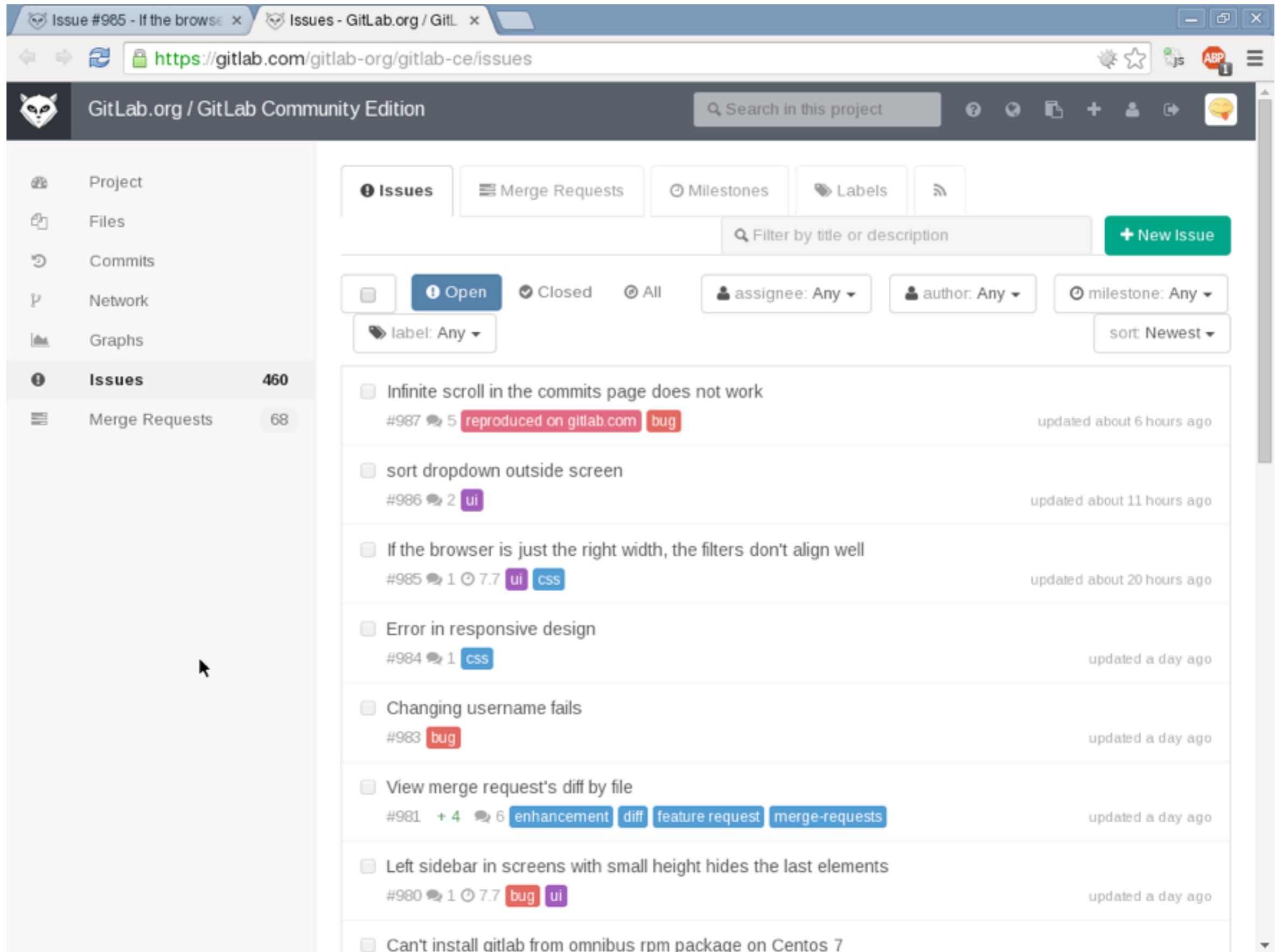
- See particular commit changes by commit tag:

```
$> git show 74bc732
```

...

- See particular ETC...
- Very flexible
- One may list the branches tree, selected commits,
- Please, refer to manuals and read a book once

GitLab is a web-based interface



The screenshot displays the GitLab web interface for the 'GitLab.org / GitLab Community Edition' project. The browser address bar shows the URL <https://gitlab.com/gitlab-org/gitlab-ce/issues>. The page features a navigation sidebar on the left with options: Project, Files, Commits, Network, Graphs, Issues (460), and Merge Requests (68). The main content area is titled 'Issues' and includes a search bar, a '+ New Issue' button, and filter controls for status (Open, Closed, All), assignee, author, milestone, and label. A list of issues is shown, including:

- Infinite scroll in the commits page does not work (#987, 5 comments, reproduced on gitlab.com, bug, updated about 6 hours ago)
- sort dropdown outside screen (#986, 2 comments, ui, updated about 11 hours ago)
- If the browser is just the right width, the filters don't align well (#985, 1 comment, 7.7 votes, ui, css, updated about 20 hours ago)
- Error in responsive design (#984, 1 comment, css, updated a day ago)
- Changing username fails (#983, bug, updated a day ago)
- View merge request's diff by file (#981, +4, 6 comments, enhancement, diff, feature request, merge-requests, updated a day ago)
- Left sidebar in screens with small height hides the last elements (#980, 1 comment, 7.7 votes, bug, ui, updated a day ago)
- Can't install gitlab from omnibus rpm package on Centos 7

GitLab is a web-based interface

Very convenient for:

- Code browsing
- Changes viewing
- Issue tracker
- General project management.
- Some learning curve

Choices to make:

- Platform: gitlab*, github, atlassian
- Site: CERN, DESY, JINR :), opensource (=limits)
- Tactics: switch at once, or gradually (not sure how)
- ...