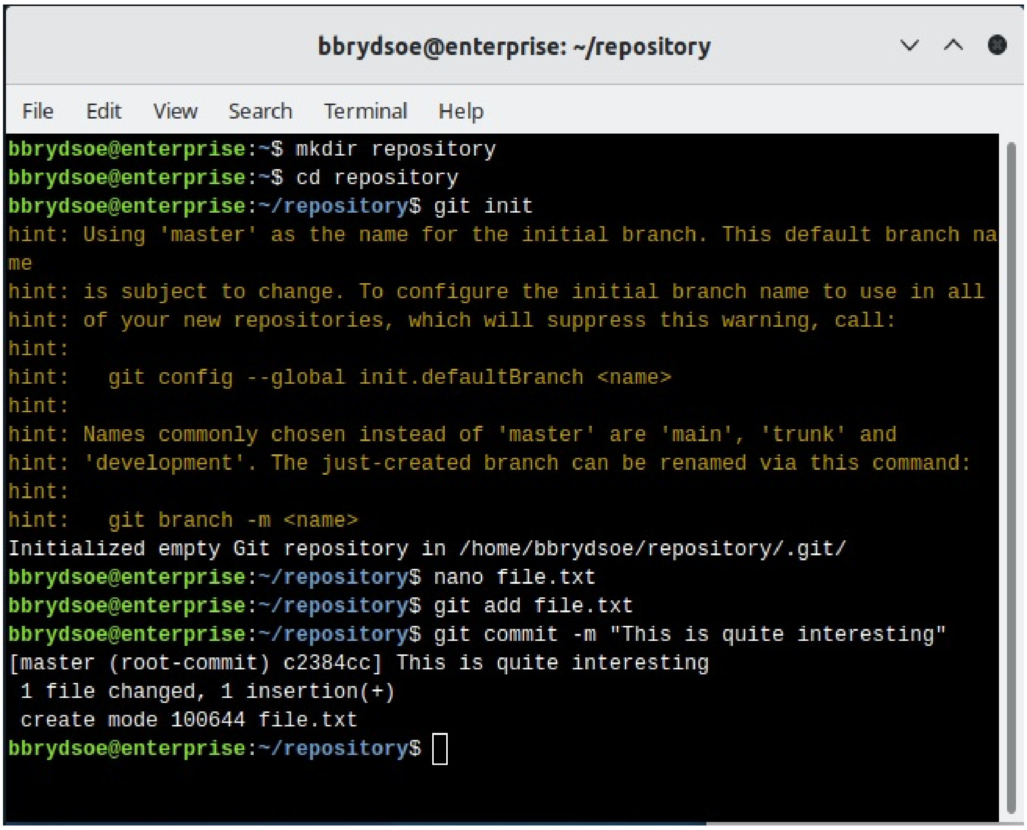
Answers to exercises for section 3. Basic concepts

1. Create a repository, add (stage) and commit a file

Steps:

```
mkdir repository
cd repository
git init
nano file.txt
git add file.txt
git commit -m "This is quite interesting"
```

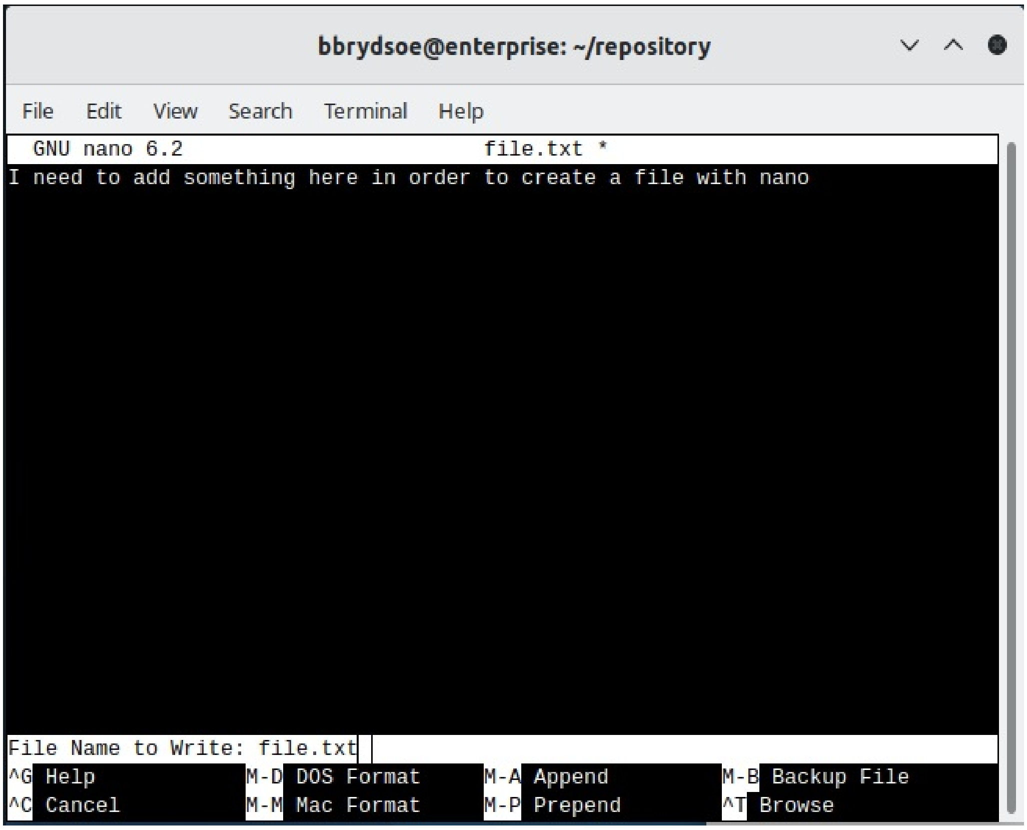
Answer:



bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
bbrydsoe@enterprise:~$ mkdir repository
bbrydsoe@enterprise:~$ cd repository
bbrydsoe@enterprise:~/repository$ git init
hint: Using 'master' as the name for the initial branch. This default branch na
me
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/bbrydsoe/repository/.git/
bbrydsoe@enterprise:~/repository$ nano file.txt
bbrydsoe@enterprise:~/repository$ git add file.txt
bbrydsoe@enterprise:~/repository$ git commit -m "This is quite interesting"
[master (root-commit) c2384cc] This is quite interesting
 1 file changed, 1 insertion(+)
 create mode 100644 file.txt
bbrydsoe@enterprise:~/repository$
```

Note that you need to write something in the nano editor in order for it to save the file:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
  GNU nano 6.2                        file.txt *
I need to add something here in order to create a file with nano









File Name to Write: file.txt
^G Help            M-D DOS Format     M-A Append         M-B Backup File
^C Cancel          M-M Mac Format     M-P Prepend        ^T Browse
```

1. Take a look at the stuff under the .git directory

Steps:

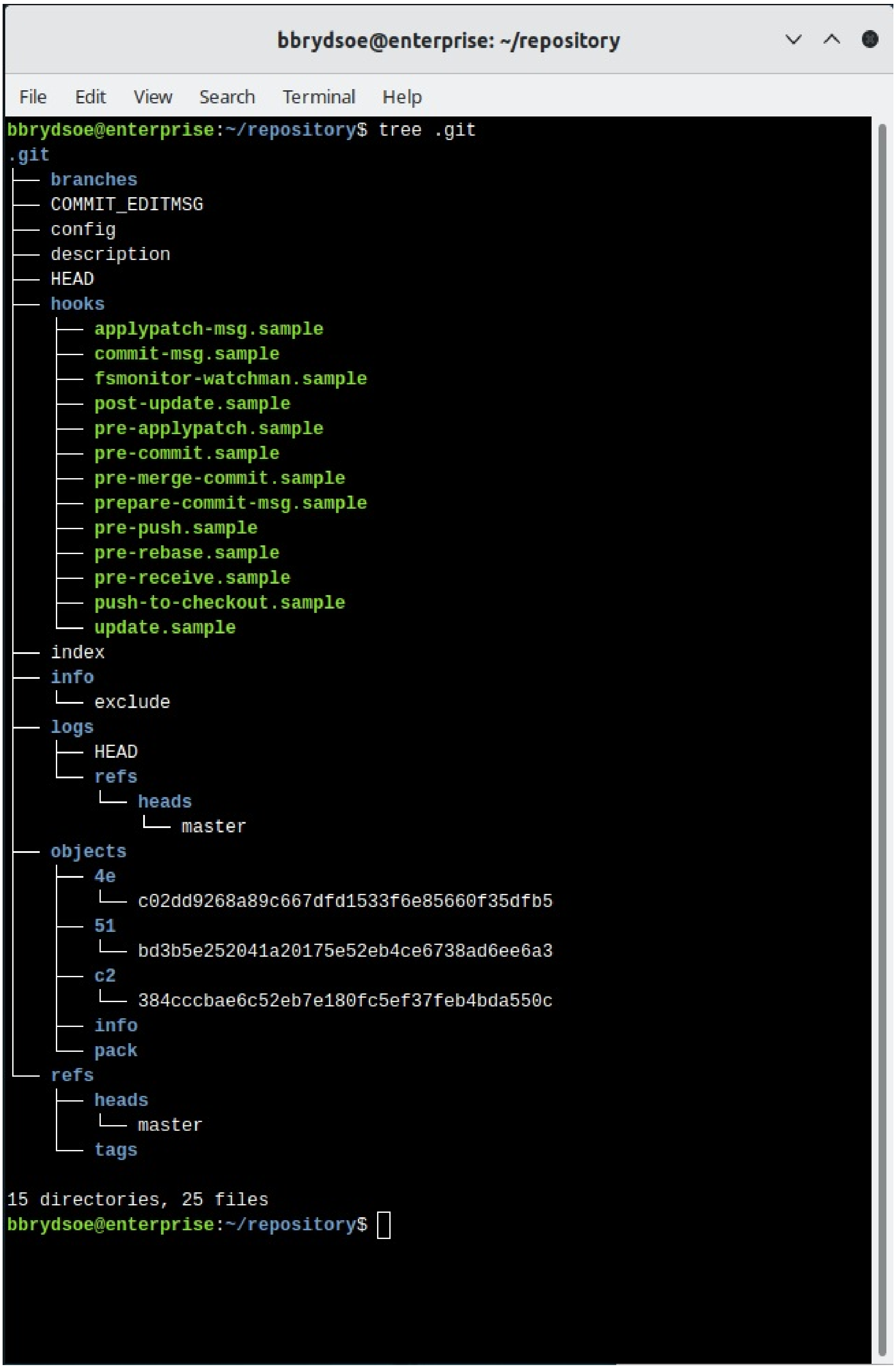
```
cd repository
tree .git
git cat-file -p <hash>
```

where is the various hash'es you can see with

```
tree .git
```

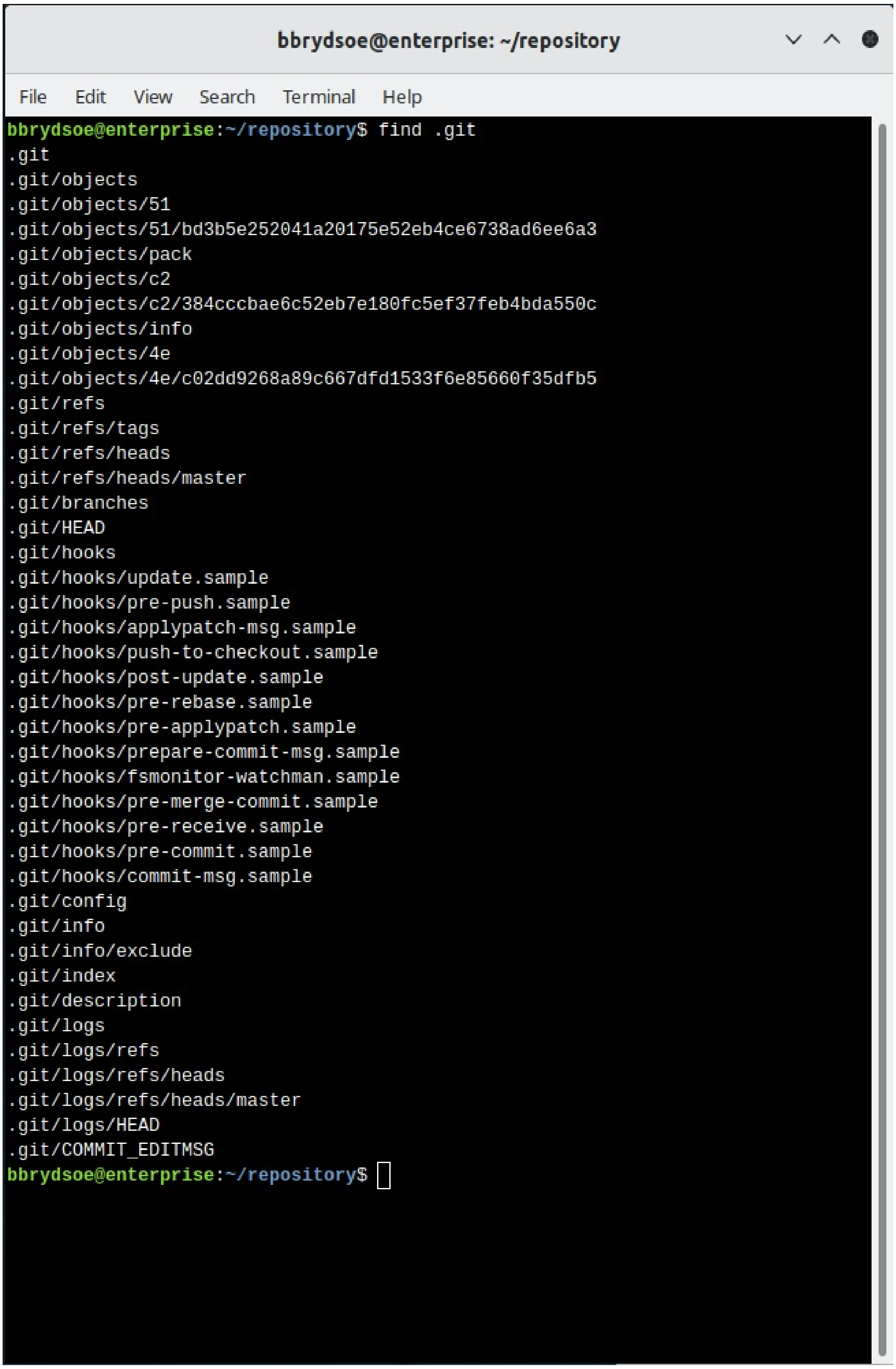
Remember that the two-letter name of the subdirectory under objects-dir is included in the hash

Answer:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

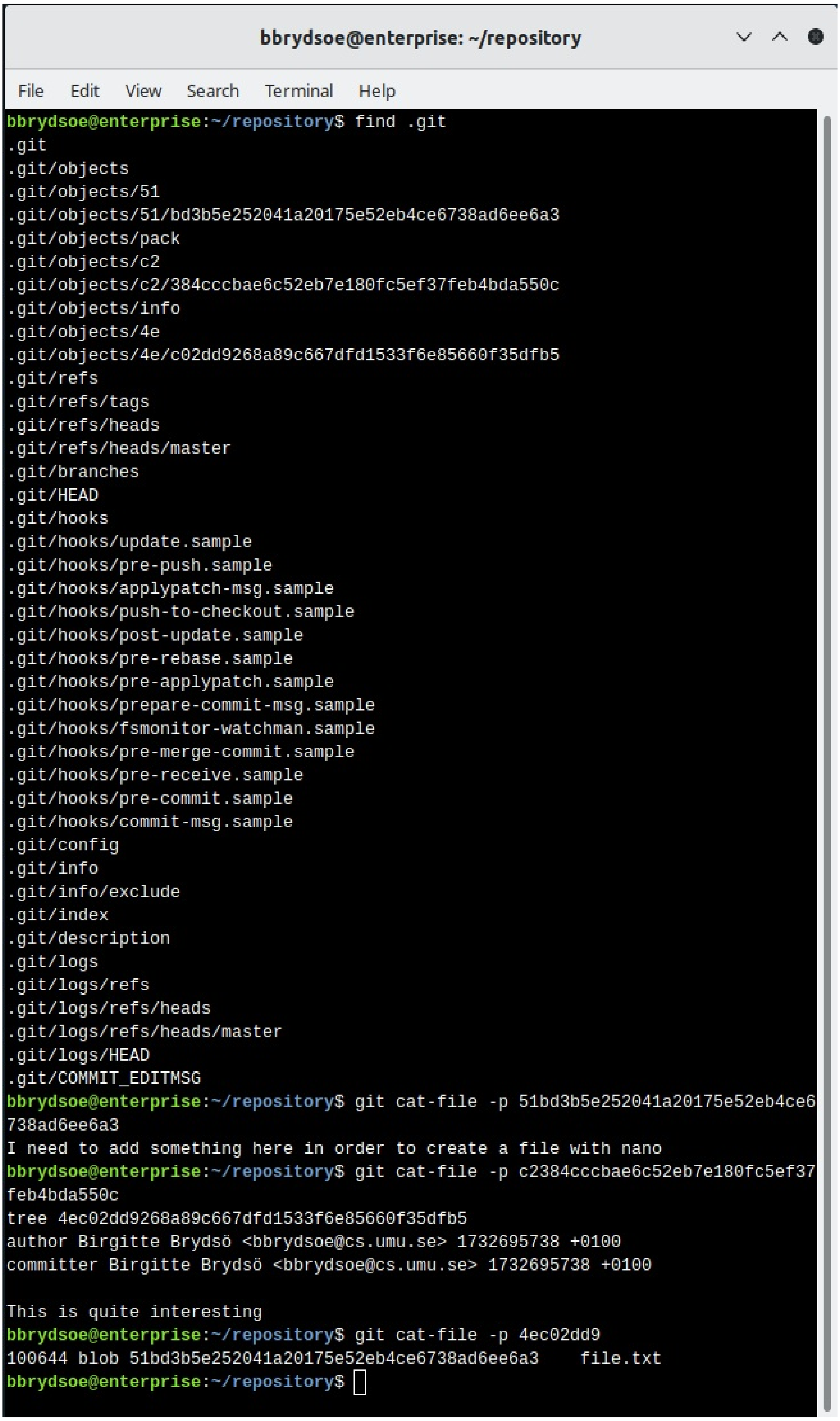
```
bbrydsoe@enterprise:~/repository$ tree .git
.git
├── branches
├── COMMIT_EDITMSG
├── config
├── description
├── HEAD
├── hooks
│   ├── applypatch-msg.sample
│   ├── commit-msg.sample
│   ├── fsmonitor-watchman.sample
│   ├── post-update.sample
│   ├── pre-applypatch.sample
│   ├── pre-commit.sample
│   ├── pre-merge-commit.sample
│   ├── prepare-commit-msg.sample
│   ├── pre-push.sample
│   ├── pre-rebase.sample
│   ├── pre-receive.sample
│   ├── push-to-checkout.sample
│   └── update.sample
├── index
├── info
│   └── exclude
├── logs
│   ├── HEAD
│   └── refs
│       └── heads
│           └── master
├── objects
│   ├── 4e
│   │   └── c02dd9268a89c667dfd1533f6e85660f35dfb5
│   ├── 51
│   │   └── bd3b5e252041a20175e52eb4ce6738ad6ee6a3
│   ├── c2
│   │   └── 384cccbae6c52eb7e180fc5ef37feb4bda550c
│   ├── info
│   └── pack
└── refs
    ├── heads
    │   └── master
    └── tags

15 directories, 25 files
bbrydsoe@enterprise:~/repository$ 
```

If you do not have `tree` on your system, you can use `find`.


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
bbrydsoe@enterprise:~/repository$ find .git
.git
.git/objects
.git/objects/51
.git/objects/51/bd3b5e252041a20175e52eb4ce6738ad6ee6a3
.git/objects/pack
.git/objects/c2
.git/objects/c2/384cccbae6c52eb7e180fc5ef37feb4bda550c
.git/objects/info
.git/objects/4e
.git/objects/4e/c02dd9268a89c667dfd1533f6e85660f35dfb5
.git/refs
.git/refs/tags
.git/refs/heads
.git/refs/heads/master
.git/branches
.git/HEAD
.git/hooks
.git/hooks/update.sample
.git/hooks/pre-push.sample
.git/hooks/applypatch-msg.sample
.git/hooks/push-to-checkout.sample
.git/hooks/post-update.sample
.git/hooks/pre-rebase.sample
.git/hooks/pre-applypatch.sample
.git/hooks/prepare-commit-msg.sample
.git/hooks/fsmonitor-watchman.sample
.git/hooks/pre-merge-commit.sample
.git/hooks/pre-receive.sample
.git/hooks/pre-commit.sample
.git/hooks/commit-msg.sample
.git/config
.git/info
.git/info/exclude
.git/index
.git/description
.git/logs
.git/logs/refs
.git/logs/refs/heads
.git/logs/refs/heads/master
.git/logs/HEAD
.git/COMMIT_EDITMSG
bbrydsoe@enterprise:~/repository$ 
```

Then do cat-file on a hash:



```
bbrydsoe@enterprise:~/repository$ find .git
.git
.git/objects
.git/objects/51
.git/objects/51/bd3b5e252041a20175e52eb4ce6738ad6ee6a3
.git/objects/pack
.git/objects/c2
.git/objects/c2/384cccbae6c52eb7e180fc5ef37feb4bda550c
.git/objects/info
.git/objects/4e
.git/objects/4e/c02dd9268a89c667dfd1533f6e85660f35dfb5
.git/refs
.git/refs/tags
.git/refs/heads
.git/refs/heads/master
.git/branches
.git/HEAD
.git/hooks
.git/hooks/update.sample
.git/hooks/pre-push.sample
.git/hooks/applypatch-msg.sample
.git/hooks/push-to-checkout.sample
.git/hooks/post-update.sample
.git/hooks/pre-rebase.sample
.git/hooks/pre-applypatch.sample
.git/hooks/prepare-commit-msg.sample
.git/hooks/fsmonitor-watchman.sample
.git/hooks/pre-merge-commit.sample
.git/hooks/pre-receive.sample
.git/hooks/pre-commit.sample
.git/hooks/commit-msg.sample
.git/config
.git/info
.git/info/exclude
.git/index
.git/description
.git/logs
.git/logs/refs
.git/logs/refs/heads
.git/logs/refs/heads/master
.git/logs/HEAD
.git/COMMIT_EDITMSG
bbrydsoe@enterprise:~/repository$ git cat-file -p 51bd3b5e252041a20175e52eb4ce6
738ad6ee6a3
I need to add something here in order to create a file with nano
bbrydsoe@enterprise:~/repository$ git cat-file -p c2384cccbae6c52eb7e180fc5ef37
feb4bda550c
tree 4ec02dd9268a89c667dfd1533f6e85660f35dfb5
author Birgitte Brydsö <bbrydsoe@cs.umu.se> 1732695738 +0100
committer Birgitte Brydsö <bbrydsoe@cs.umu.se> 1732695738 +0100

This is quite interesting
bbrydsoe@enterprise:~/repository$ git cat-file -p 4ec02dd9
100644 blob 51bd3b5e252041a20175e52eb4ce6738ad6ee6a3    file.txt
bbrydsoe@enterprise:~/repository$ 
```

!!! note

`Note that you include the two-letter name of the subdirectory in the hash!`

---

1. Make some changes.

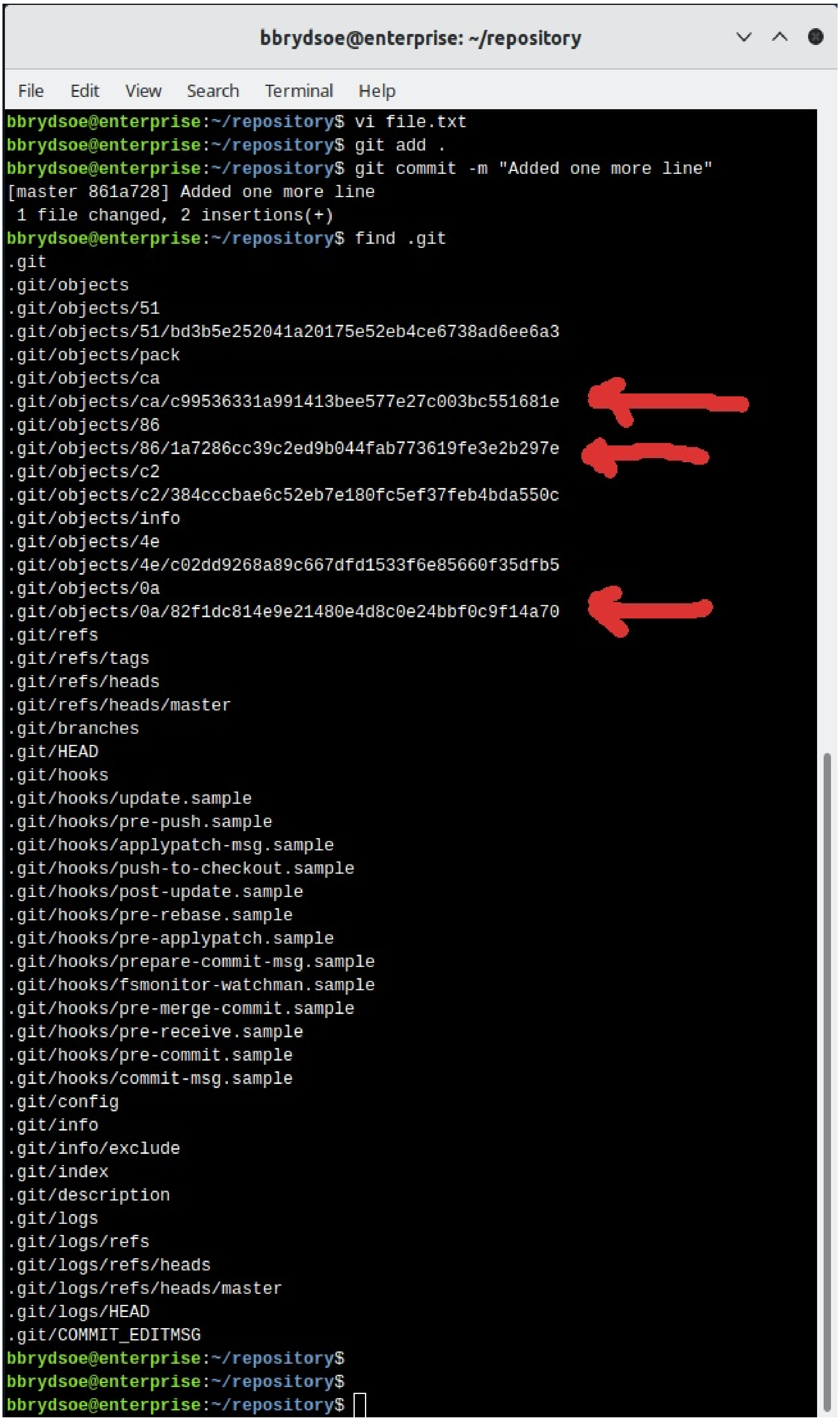
Change one or more files. Add and commit. Make some changes. Add and commit.

Look at what happens in the .git tree

Answer:

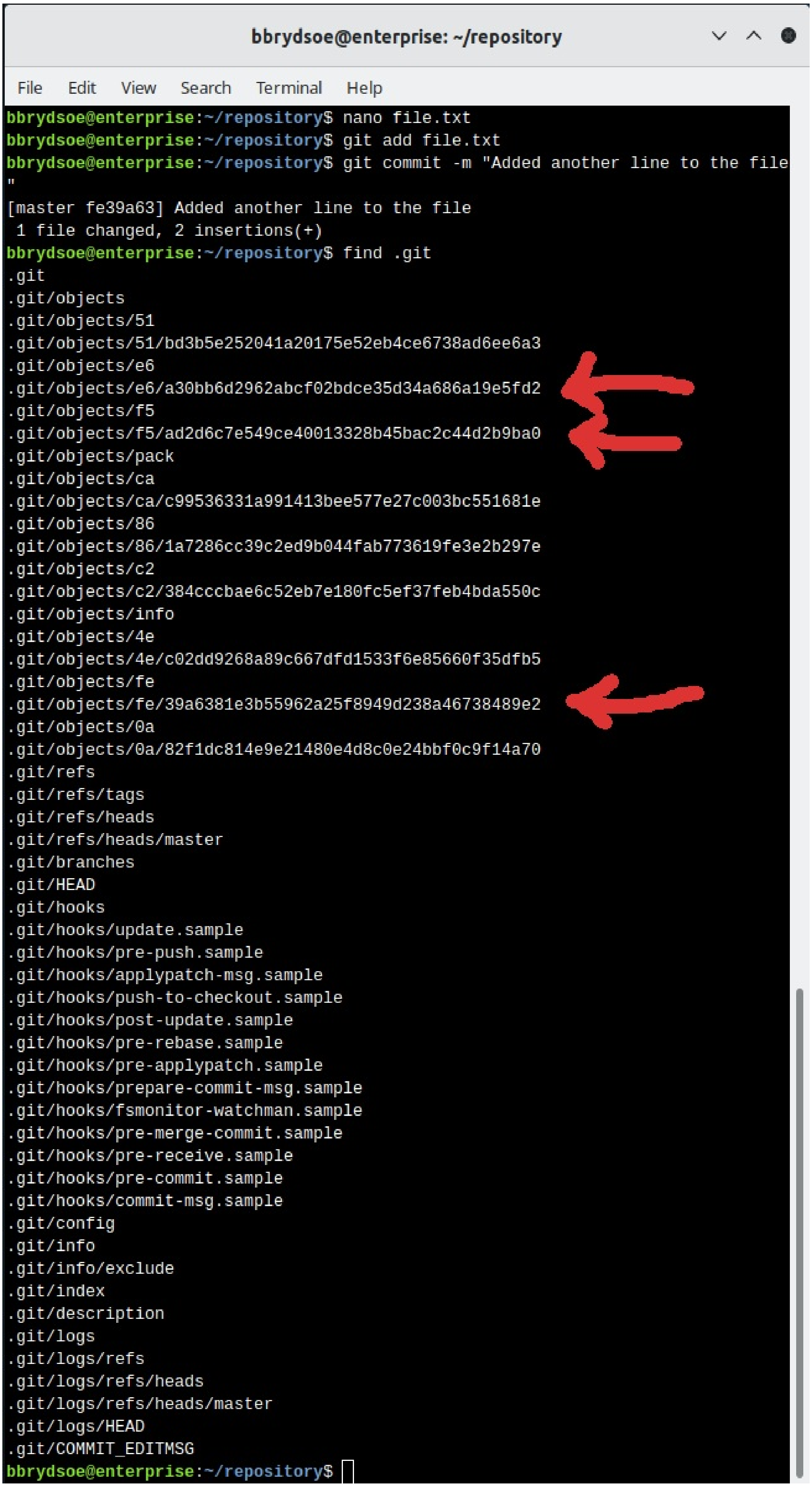
I will be using `find`

Making a change:



```
bbrydsoe@enterprise:~/repository$ vi file.txt
bbrydsoe@enterprise:~/repository$ git add .
bbrydsoe@enterprise:~/repository$ git commit -m "Added one more line"
[master 861a728] Added one more line
 1 file changed, 2 insertions(+)
bbrydsoe@enterprise:~/repository$ find .git
.git
.git/objects
.git/objects/51
.git/objects/51/bd3b5e252041a20175e52eb4ce6738ad6ee6a3
.git/objects/pack
.git/objects/ca
.git/objects/ca/c99536331a991413bee577e27c003bc551681e
.git/objects/86
.git/objects/86/1a7286cc39c2ed9b044fab773619fe3e2b297e
.git/objects/c2
.git/objects/c2/384cccbae6c52eb7e180fc5ef37feb4bda550c
.git/objects/info
.git/objects/4e
.git/objects/4e/c02dd9268a89c667dfd1533f6e85660f35dfb5
.git/objects/0a
.git/objects/0a/82f1dc814e9e21480e4d8c0e24bbf0c9f14a70
.git/refs
.git/refs/tags
.git/refs/heads
.git/refs/heads/master
.git/branches
.git/HEAD
.git/hooks
.git/hooks/update.sample
.git/hooks/pre-push.sample
.git/hooks/applypatch-msg.sample
.git/hooks/push-to-checkout.sample
.git/hooks/post-update.sample
.git/hooks/pre-rebase.sample
.git/hooks/pre-applypatch.sample
.git/hooks/prepare-commit-msg.sample
.git/hooks/fsmonitor-watchman.sample
.git/hooks/pre-merge-commit.sample
.git/hooks/pre-receive.sample
.git/hooks/pre-commit.sample
.git/hooks/commit-msg.sample
.git/config
.git/info
.git/info/exclude
.git/index
.git/description
.git/logs
.git/logs/refs
.git/logs/refs/heads
.git/logs/refs/heads/master
.git/logs/HEAD
.git/COMMIT_EDITMSG
bbrydsoe@enterprise:~/repository$
bbrydsoe@enterprise:~/repository$
bbrydsoe@enterprise:~/repository$
```

Making another change:



```
bbrydsoe@enterprise:~/repository$ nano file.txt
bbrydsoe@enterprise:~/repository$ git add file.txt
bbrydsoe@enterprise:~/repository$ git commit -m "Added another line to the file
"
[master fe39a63] Added another line to the file
 1 file changed, 2 insertions(+)
bbrydsoe@enterprise:~/repository$ find .git
.git
.git/objects
.git/objects/51
.git/objects/51/bd3b5e252041a20175e52eb4ce6738ad6ee6a3
.git/objects/e6
.git/objects/e6/a30bb6d2962abcf02bdce35d34a686a19e5fd2
.git/objects/f5
.git/objects/f5/ad2d6c7e549ce40013328b45bac2c44d2b9ba0
.git/objects/pack
.git/objects/ca
.git/objects/ca/c99536331a991413bee577e27c003bc551681e
.git/objects/86
.git/objects/86/1a7286cc39c2ed9b044fab773619fe3e2b297e
.git/objects/c2
.git/objects/c2/384cccbae6c52eb7e180fc5ef37feb4bda550c
.git/objects/info
.git/objects/4e
.git/objects/4e/c02dd9268a89c667dfd1533f6e85660f35dfb5
.git/objects/fe
.git/objects/fe/39a6381e3b55962a25f8949d238a46738489e2
.git/objects/0a
.git/objects/0a/82f1dc814e9e21480e4d8c0e24bbf0c9f14a70
.git/refs
.git/refs/tags
.git/refs/heads
.git/refs/heads/master
.git/branches
.git/HEAD
.git/hooks
.git/hooks/update.sample
.git/hooks/pre-push.sample
.git/hooks/applypatch-msg.sample
.git/hooks/push-to-checkout.sample
.git/hooks/post-update.sample
.git/hooks/pre-rebase.sample
.git/hooks/pre-applypatch.sample
.git/hooks/prepare-commit-msg.sample
.git/hooks/fsmonitor-watchman.sample
.git/hooks/pre-merge-commit.sample
.git/hooks/pre-receive.sample
.git/hooks/pre-commit.sample
.git/hooks/commit-msg.sample
.git/config
.git/info
.git/info/exclude
.git/index
.git/description
.git/logs
.git/logs/refs
.git/logs/refs/heads
.git/logs/refs/heads/master
.git/logs/HEAD
.git/COMMIT_EDITMSG
bbrydsoe@enterprise:~/repository$
```

1. Go to earlier commits and make a new branch

Steps:

Try go back to an earlier commit with “git checkout ”

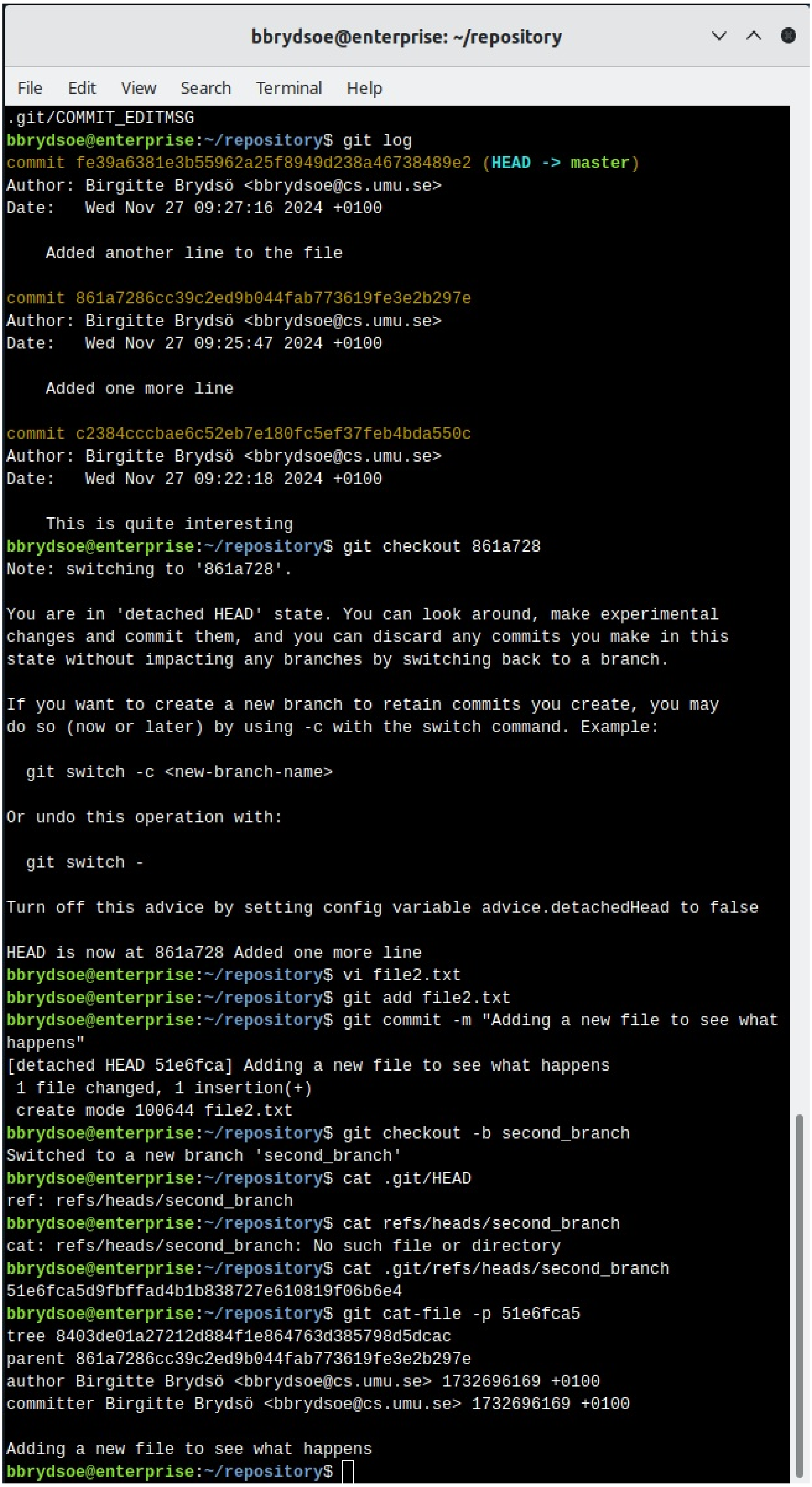
Make come changes to a file there, or try adding a new file (and add+commit)

Create a new branch from that commit: “git checkout -b second_branch”

Find the hash of the commit the HEAD points to now

```
cat .git/HEAD
cat <path you got above>
git cat-file -p <hash you just got from above command>
```

Answer:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
.git/COMMIT_EDITMSG
bbrydsoe@enterprise:~/repository$ git log
commit fe39a6381e3b55962a25f8949d238a46738489e2 (HEAD -> master)
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:27:16 2024 +0100

    Added another line to the file

commit 861a7286cc39c2ed9b044fab773619fe3e2b297e
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:25:47 2024 +0100

    Added one more line

commit c2384cccbae6c52eb7e180fc5ef37feb4bda550c
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:22:18 2024 +0100

    This is quite interesting
bbrydsoe@enterprise:~/repository$ git checkout 861a728
Note: switching to '861a728'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 861a728 Added one more line
bbrydsoe@enterprise:~/repository$ vi file2.txt
bbrydsoe@enterprise:~/repository$ git add file2.txt
bbrydsoe@enterprise:~/repository$ git commit -m "Adding a new file to see what
happens"
[detached HEAD 51e6fca] Adding a new file to see what happens
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt
bbrydsoe@enterprise:~/repository$ git checkout -b second_branch
Switched to a new branch 'second_branch'
bbrydsoe@enterprise:~/repository$ cat .git/HEAD
ref: refs/heads/second_branch
bbrydsoe@enterprise:~/repository$ cat refs/heads/second_branch
cat: refs/heads/second_branch: No such file or directory
bbrydsoe@enterprise:~/repository$ cat .git/refs/heads/second_branch
51e6fca5d9fbffad4b1b838727e610819f06b6e4
bbrydsoe@enterprise:~/repository$ git cat-file -p 51e6fca5
tree 8403de01a27212d884f1e864763d385798d5dcac
parent 861a7286cc39c2ed9b044fab773619fe3e2b297e
author Birgitte Brydsö <bbrydsoe@cs.umu.se> 1732696169 +0100
committer Birgitte Brydsö <bbrydsoe@cs.umu.se> 1732696169 +0100

Adding a new file to see what happens
bbrydsoe@enterprise:~/repository$ 
```

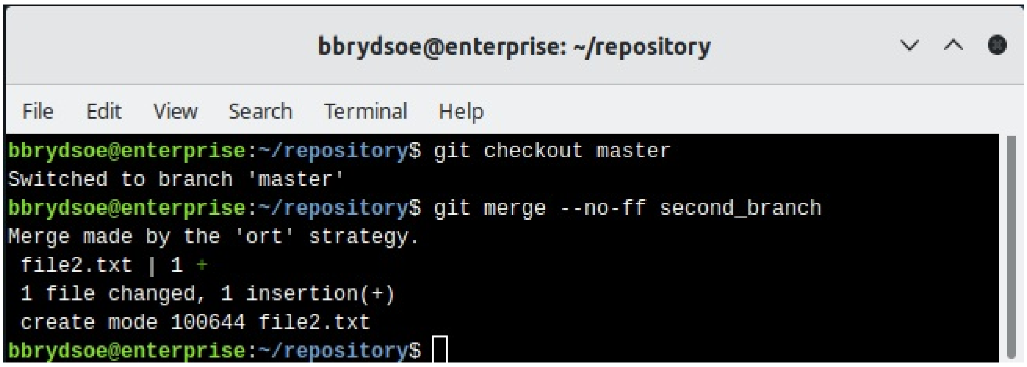
1. Go back on primary branch and merge the branches

Steps:

Go back to the primary branch (main or master) with “git checkout main” or “git checkout master”

Merge the new branch you created to master with “git merge –no-ff second_branch” (or whatever you called the new branch)

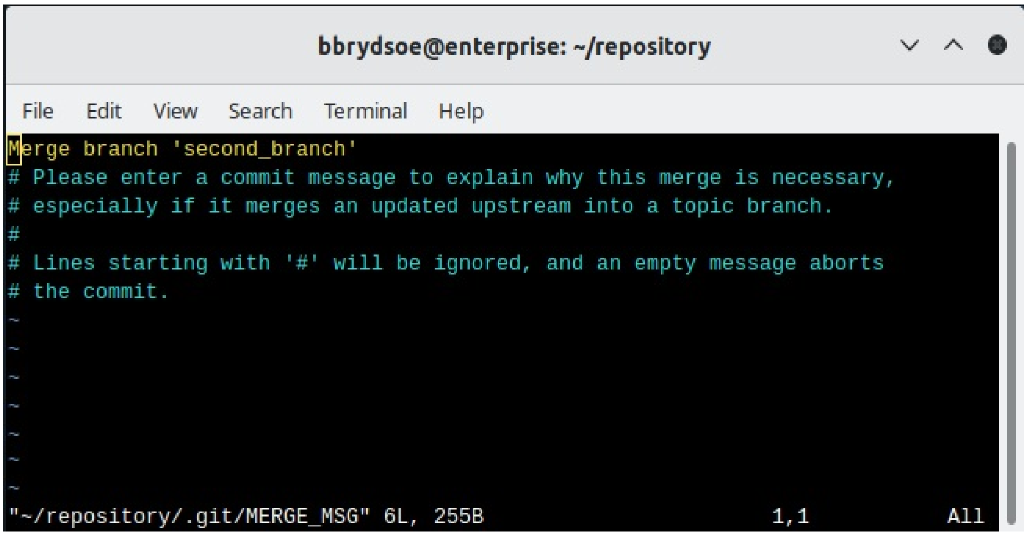
Answer:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
bbrydsoe@enterprise:~/repository$ git checkout master
Switched to branch 'master'
bbrydsoe@enterprise:~/repository$ git merge --no-ff second_branch
Merge made by the 'ort' strategy.
 file2.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt
bbrydsoe@enterprise:~/repository$
```

Note that here there was no conflict.

And this is how it looked when it opened the merge message (in vim):

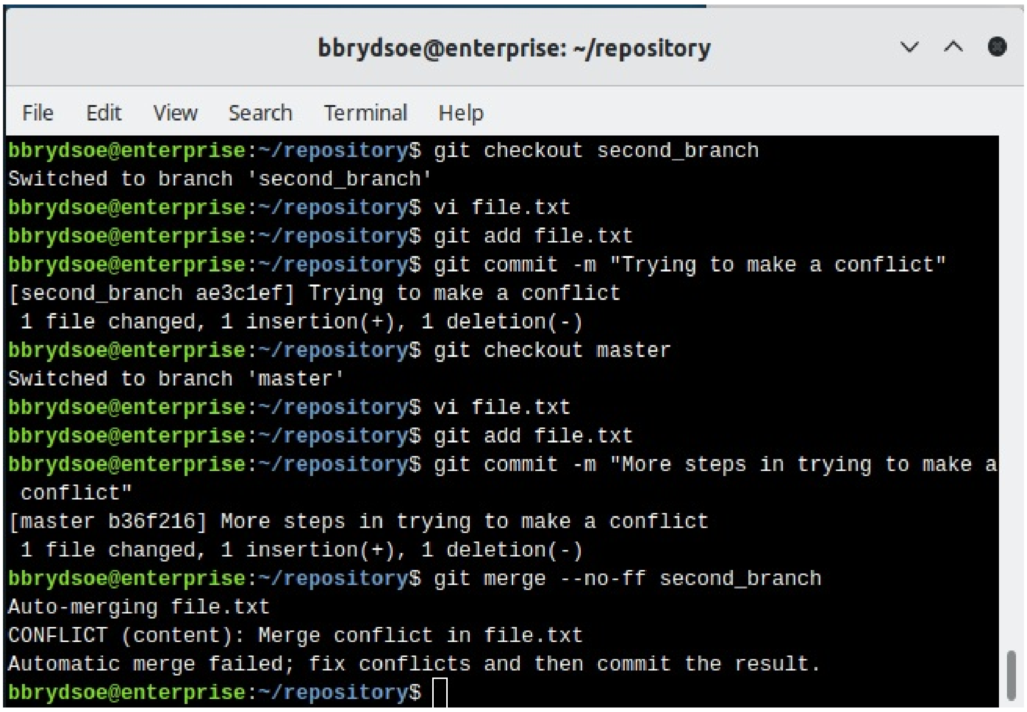

bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
Merge branch 'second_branch'
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
"~/repository/.git/MERGE_MSG" 6L, 255B                     1,1           All
```

I make no change, just save/close the editor and the merge continues as shown in image above this image.

An easy way to get conflicts is to edit the same file in the same line on both branches, then adding and committing on both branches before trying to merge second_branch to master (on master branch).

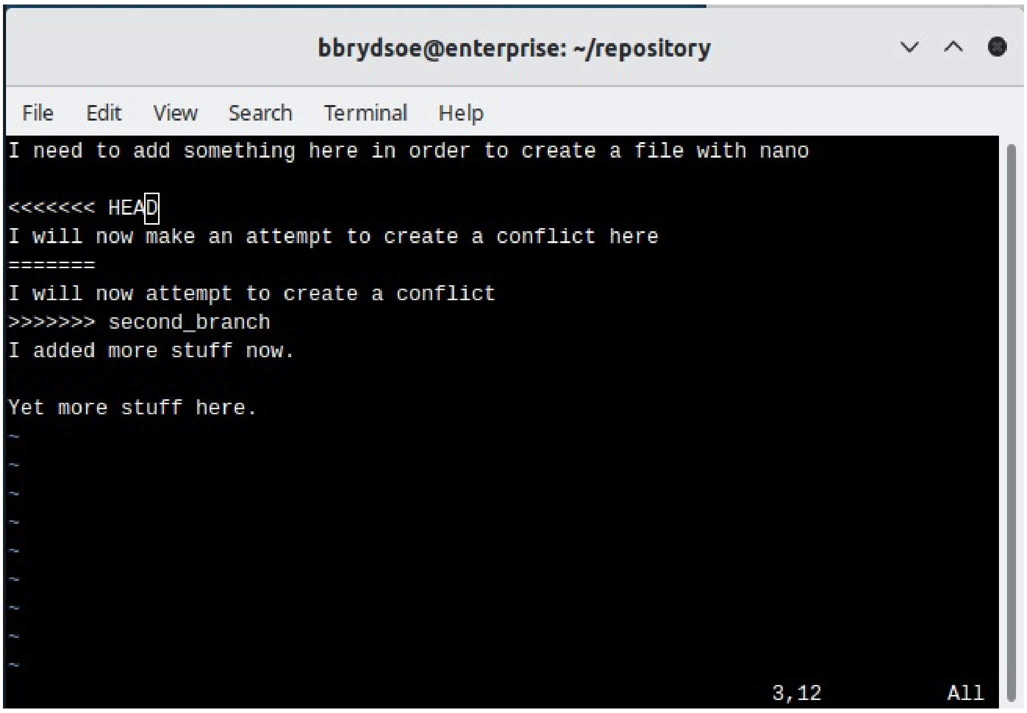
This is how it looks when there are conflicts:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
bbrydsoe@enterprise:~/repository$ git checkout second_branch
Switched to branch 'second_branch'
bbrydsoe@enterprise:~/repository$ vi file.txt
bbrydsoe@enterprise:~/repository$ git add file.txt
bbrydsoe@enterprise:~/repository$ git commit -m "Trying to make a conflict"
[second_branch ae3c1ef] Trying to make a conflict
 1 file changed, 1 insertion(+), 1 deletion(-)
bbrydsoe@enterprise:~/repository$ git checkout master
Switched to branch 'master'
bbrydsoe@enterprise:~/repository$ vi file.txt
bbrydsoe@enterprise:~/repository$ git add file.txt
bbrydsoe@enterprise:~/repository$ git commit -m "More steps in trying to make a
 conflict"
[master b36f216] More steps in trying to make a conflict
 1 file changed, 1 insertion(+), 1 deletion(-)
bbrydsoe@enterprise:~/repository$ git merge --no-ff second_branch
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
bbrydsoe@enterprise:~/repository$ 
```

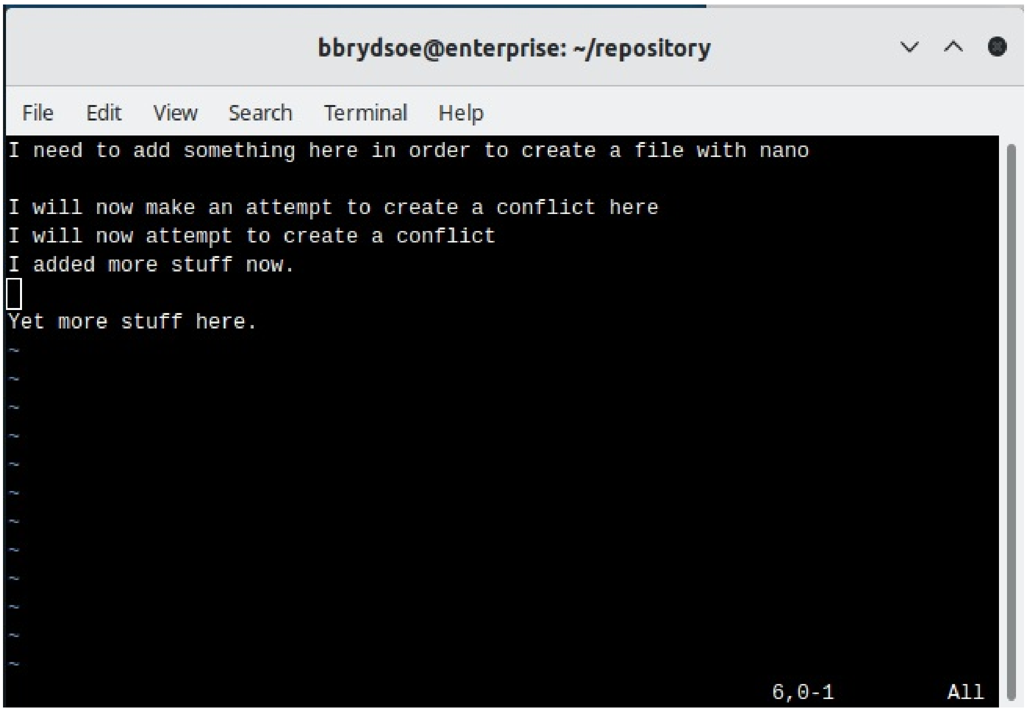
Fix any conflict by going into the conflicting file(s) and removing the stuff you don't want and also the git markers set there

This is how it looks in the file, when there are conflicts:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
I need to add something here in order to create a file with nano

<<<<<<< HEAD
I will now make an attempt to create a conflict here
=======
I will now attempt to create a conflict
>>>>>>> second_branch
I added more stuff now.

Yet more stuff here.
~
~
~
~
~
~
~
~
~
                                                              3,12          All
```

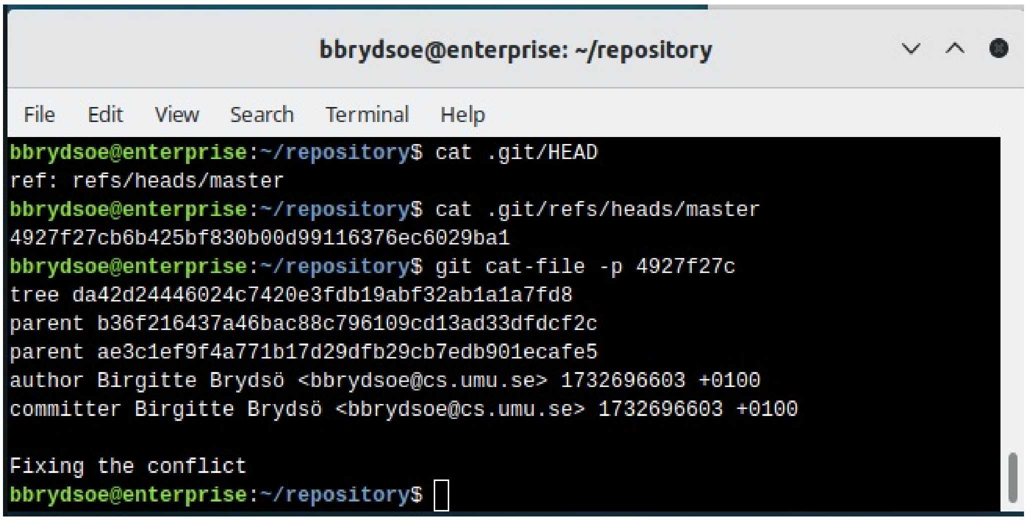
Remove the git markers and also fix the file as you want it to be (which lines should be there, etc.). I did like this:



```
I need to add something here in order to create a file with nano

I will now make an attempt to create a conflict here
I will now attempt to create a conflict
I added more stuff now.

Yet more stuff here.
~
~
~
~
~
~
~
~
~
~
~
~
                                                              6,0-1         All
```

Add the fixed file and do “git merge –continue"

Now look at what the HEAD is pointing to



```
bbrydsoe@enterprise:~/repository$ cat .git/HEAD
ref: refs/heads/master
bbrydsoe@enterprise:~/repository$ cat .git/refs/heads/master
4927f27cb6b425bf830b00d99116376ec6029ba1
bbrydsoe@enterprise:~/repository$ git cat-file -p 4927f27c
tree da42d24446024c7420e3fdb19abf32ab1a1a7fd8
parent b36f216437a46bac88c796109cd13ad33dfdcf2c
parent ae3c1ef9f4a771b17d29dfb29cb7edb901ecafe5
author Birgitte Brydsö <bbrydsoe@cs.umu.se> 1732696603 +0100
committer Birgitte Brydsö <bbrydsoe@cs.umu.se> 1732696603 +0100

Fixing the conflict
bbrydsoe@enterprise:~/repository$ 
```

---

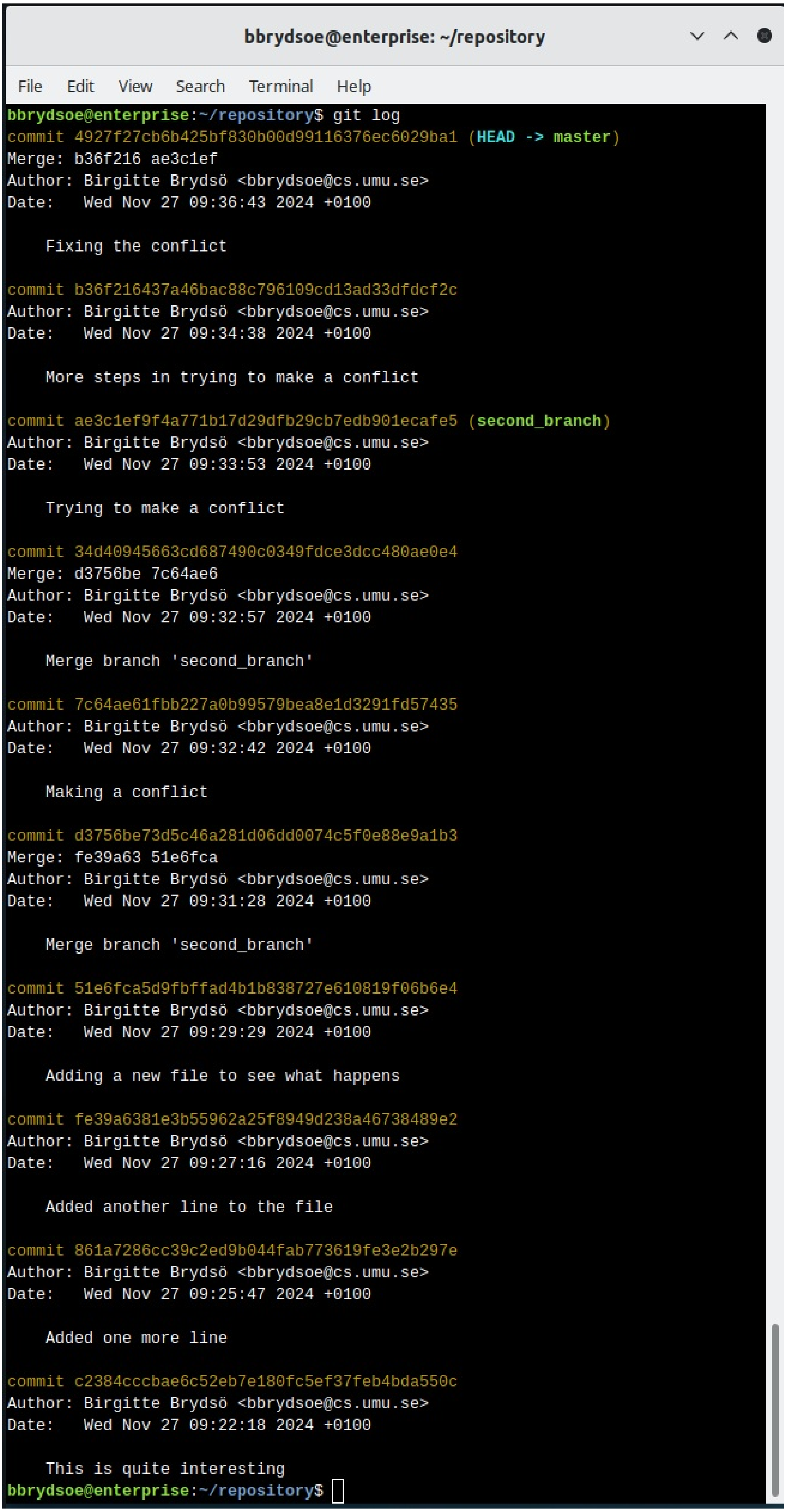
1. Check out the commit tree

You can also just try jump around in the commit tree with “git checkout ” and see how your working tree looks in various places (see that files are there or not, and with different content).

You get the list of hash’s and commit messages with “git log”

Answer:

This is how it looks for me:


bbrydsoe@enterprise: ~/repository
File Edit View Search Terminal Help

```
bbrydsoe@enterprise:~/repository$ git log
commit 4927f27cb6b425bf830b00d99116376ec6029ba1 (HEAD -> master)
Merge: b36f216 ae3c1ef
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:36:43 2024 +0100

    Fixing the conflict

commit b36f216437a46bac88c796109cd13ad33dfdcf2c
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:34:38 2024 +0100

    More steps in trying to make a conflict

commit ae3c1ef9f4a771b17d29dfb29cb7edb901ecafe5 (second_branch)
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:33:53 2024 +0100

    Trying to make a conflict

commit 34d40945663cd687490c0349fdce3dcc480ae0e4
Merge: d3756be 7c64ae6
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:32:57 2024 +0100

    Merge branch 'second_branch'

commit 7c64ae61fbb227a0b99579bea8e1d3291fd57435
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:32:42 2024 +0100

    Making a conflict

commit d3756be73d5c46a281d06dd0074c5f0e88e9a1b3
Merge: fe39a63 51e6fca
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:31:28 2024 +0100

    Merge branch 'second_branch'

commit 51e6fca5d9fbffad4b1b838727e610819f06b6e4
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:29:29 2024 +0100

    Adding a new file to see what happens

commit fe39a6381e3b55962a25f8949d238a46738489e2
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:27:16 2024 +0100

    Added another line to the file

commit 861a7286cc39c2ed9b044fab773619fe3e2b297e
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:25:47 2024 +0100

    Added one more line

commit c2384cccbae6c52eb7e180fc5ef37feb4bda550c
Author: Birgitte Brydsö <bbrydsoe@cs.umu.se>
Date:   Wed Nov 27 09:22:18 2024 +0100

    This is quite interesting
bbrydsoe@enterprise:~/repository$ 
```

To look at a specific commit, do

```
git checkout <hash>
```

for one of the hash'es.

You could then use

```
git cat-file -p <hash>
```

to check on the files.