redhat.

02 Gitshop

Git - the index

Jiří Jabůrek (jjaburek)
Red Hat 2012

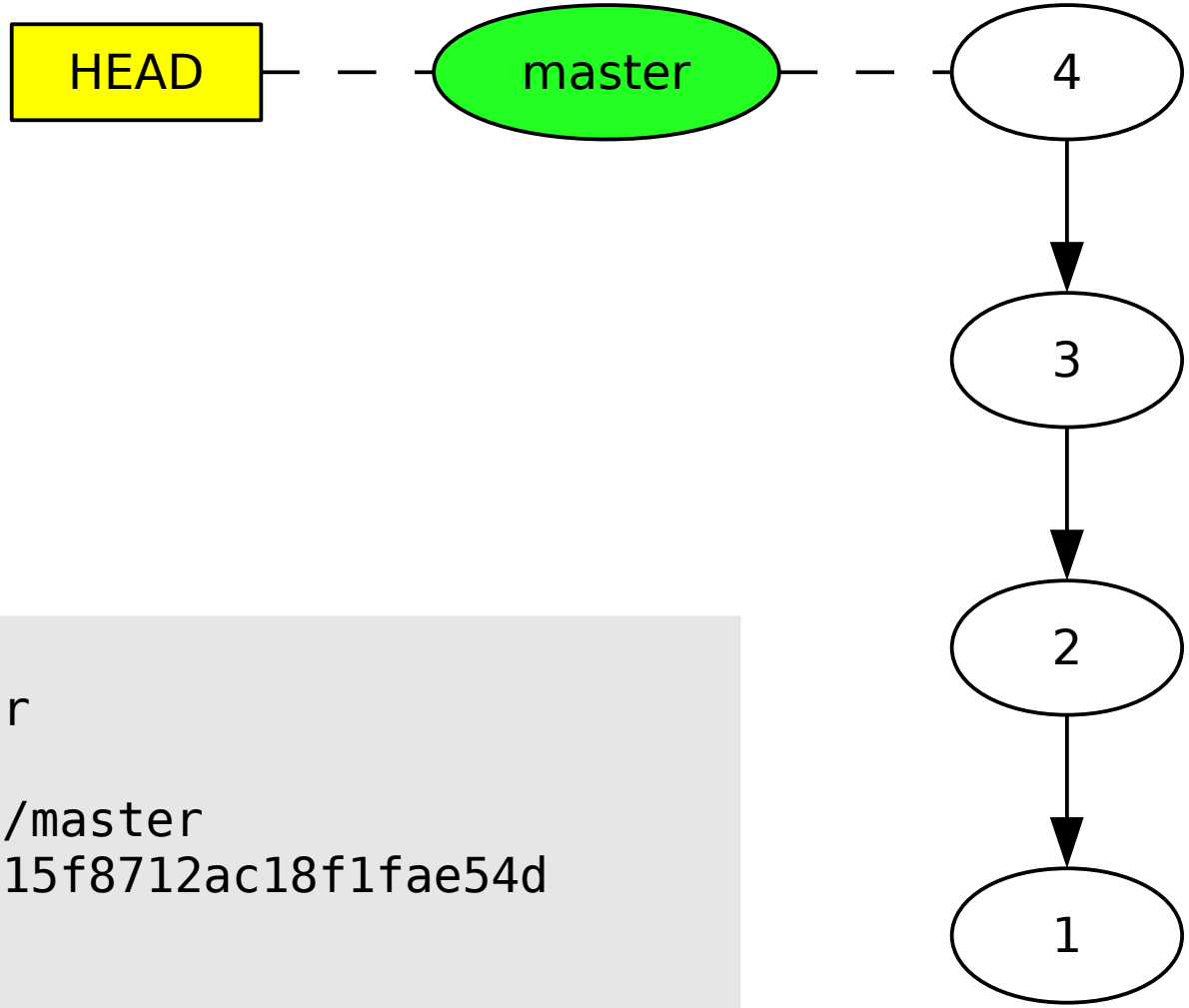
Gitshop 1 - summary

Gitshop 1 - summary

- Objects - blob, tree, commit, tag
  - (meta)data storage
  - dependencies
  - state, not difference
  - tree: file mode not really used
- References - branch, tag, symbolic ref
  - "pointers" to objects / other refs
  - text files
  - branch "moves" with new commits, tag does not

HEAD - a symbolic ref (mostly)

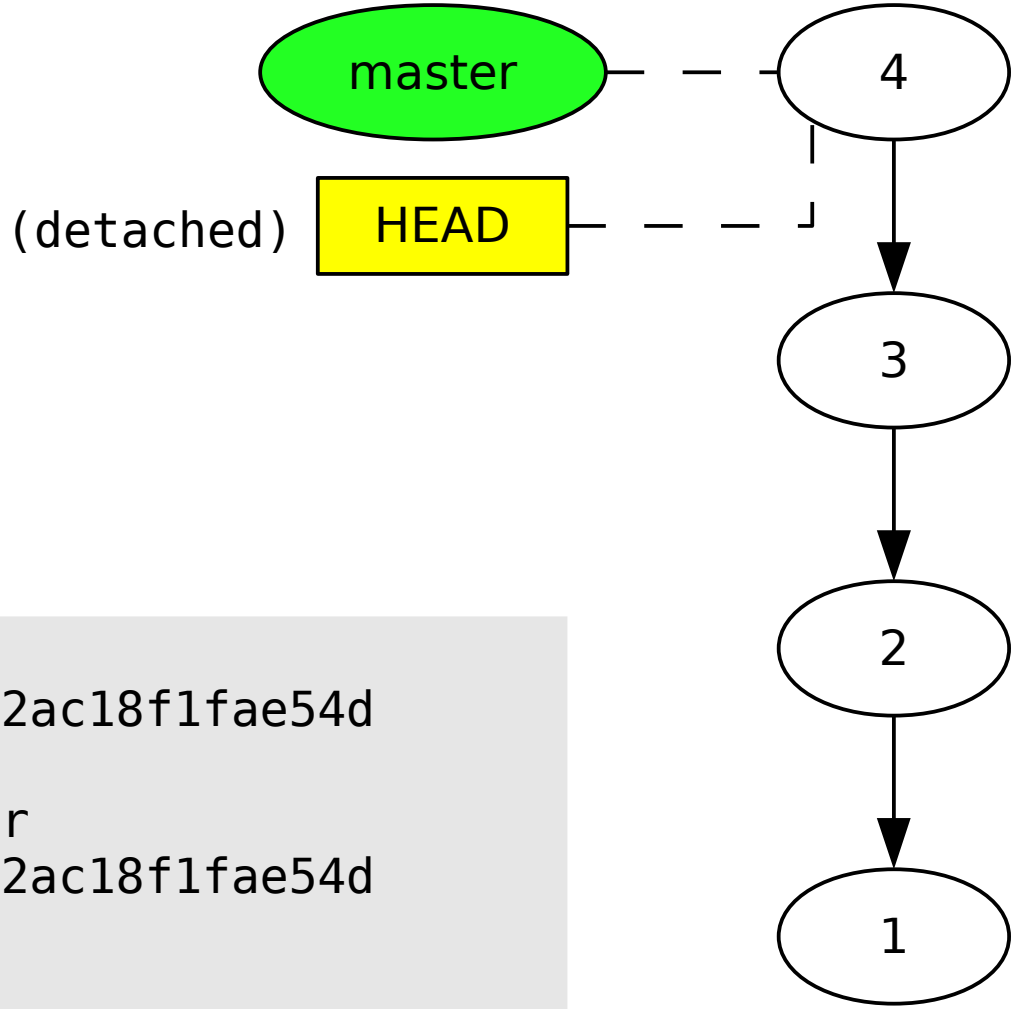
HEAD - a symbolic ref (mostly)




```
$ cat .git/HEAD
ref: refs/heads/master

$ cat .git/refs/heads/master
517b55a563b364fb890b315f8712ac18f1fae54d
```

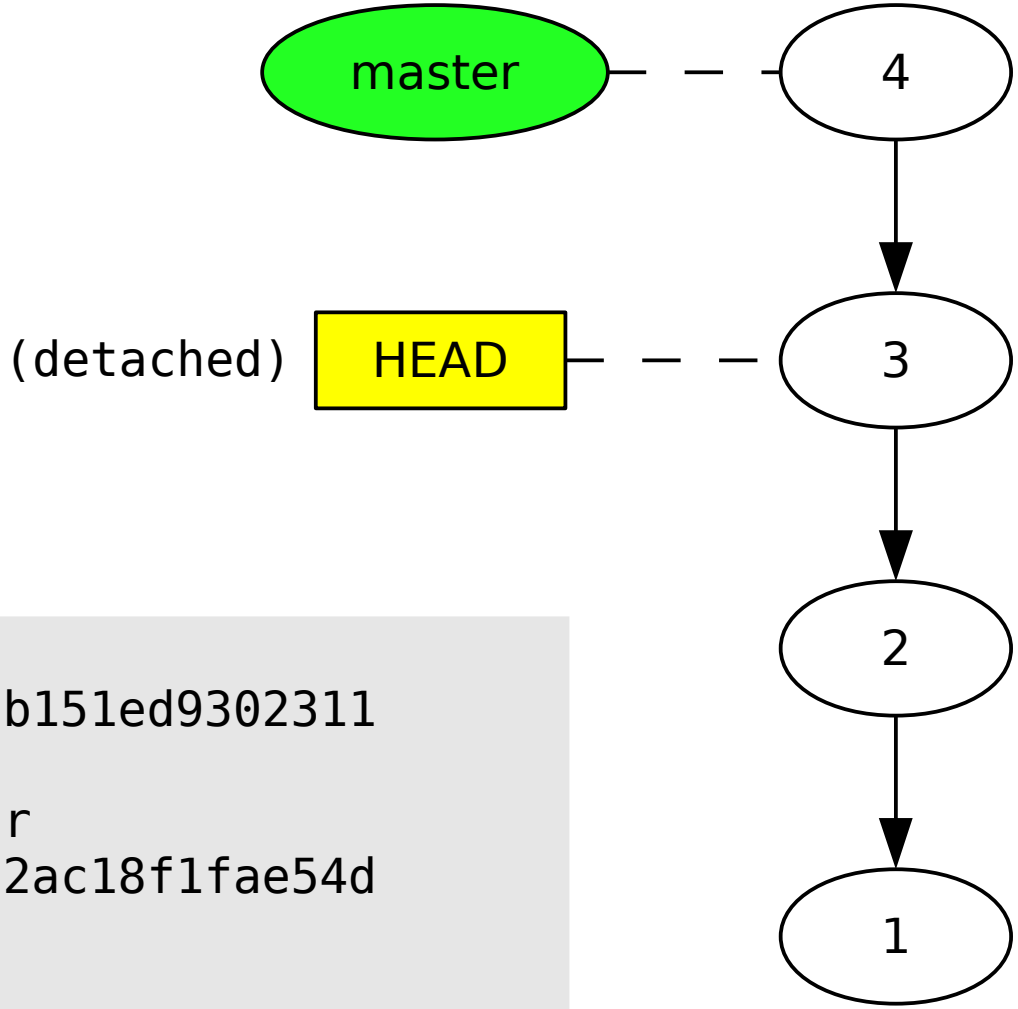
HEAD - a symbolic ref (mostly)




```
$ cat .git/HEAD
517b55a563b364fb890b315f8712ac18f1fae54d

$ cat .git/refs/heads/master
517b55a563b364fb890b315f8712ac18f1fae54d
```

HEAD - a symbolic ref (mostly)

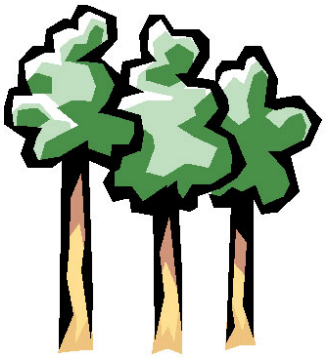



```
$ cat .git/HEAD
3c8a2ee44b6e29fb574655d0ae7b151ed9302311

$ cat .git/refs/heads/master
517b55a563b364fb890b315f8712ac18f1fae54d
```

The Index

Three trees of git



HEAD

Object database

Index

Working tree

The Index

- Real file - .git/index

```
$ file .git/index
tests/file/.git/index: Git index, version 2, 91 entries
```

- Not a btree/htable/.. implementation
  - not "database index", "array index"
- So-called "staging" area for a commit
  - main purpose: stage changes for a commit
- In older (very old) versions called "cache"
  - timestamps of all repo files, fast modify detection

The Index inside

- "Root tree transcript"
  - and more (last path modify, multiple "stages", ..)
  - initial state: root tree of HEAD commit

```
$ git ls-files --stage
100644 6ef994aa446ae9b4c575b9507c40740482448337 0   file1
100644 5664c8edc3807b6994151b5262859af8856bebc6 0   dir1/dir2/x
120000 7bfeafcad0958a7ebe196ce8b069b00d4043e279 0   somelink
<mode>             <blob sha hash>           <stage> <pathname>
```

- git-update-index creates blobs on the fly

  + adds their hashes to the index

- Removal from index --> loose objects
- Untracked files - not in index, not detected

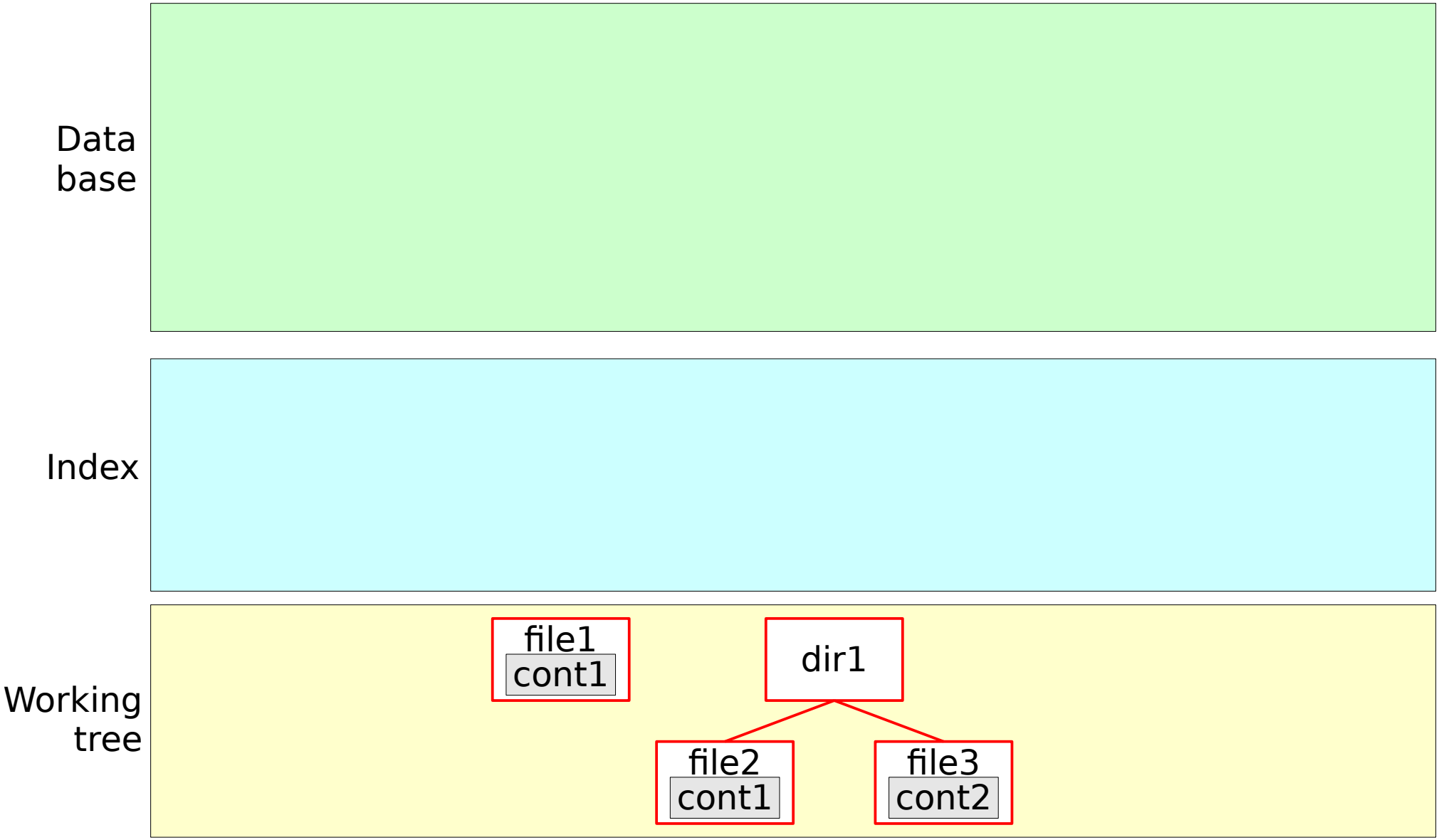
The Index inside

- Plumbing commands involved
  - git-read-tree
    - read (transcribe) tree object into index
    - multiple "stages" (namespaces), usage: merges
  - git-write-tree
    - write tree objects out of the index structure
  - git-update-index
    - perform various operations on the index
  - (git-ls-files --stage)
    - list index entries

Index usage example

The Index inside

```
$ echo "cont1" > file1
$ mkdir dir1
$ echo "cont1" > dir1/file2
$ echo "cont2" > dir1/file3
```



The Index inside

```
$ git add file1
```

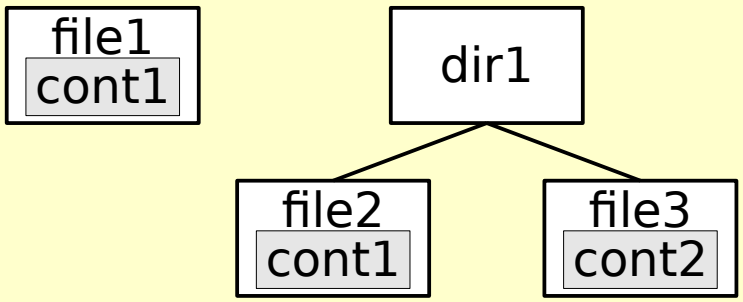
Data base

813f0b4..
"cont1"

Index

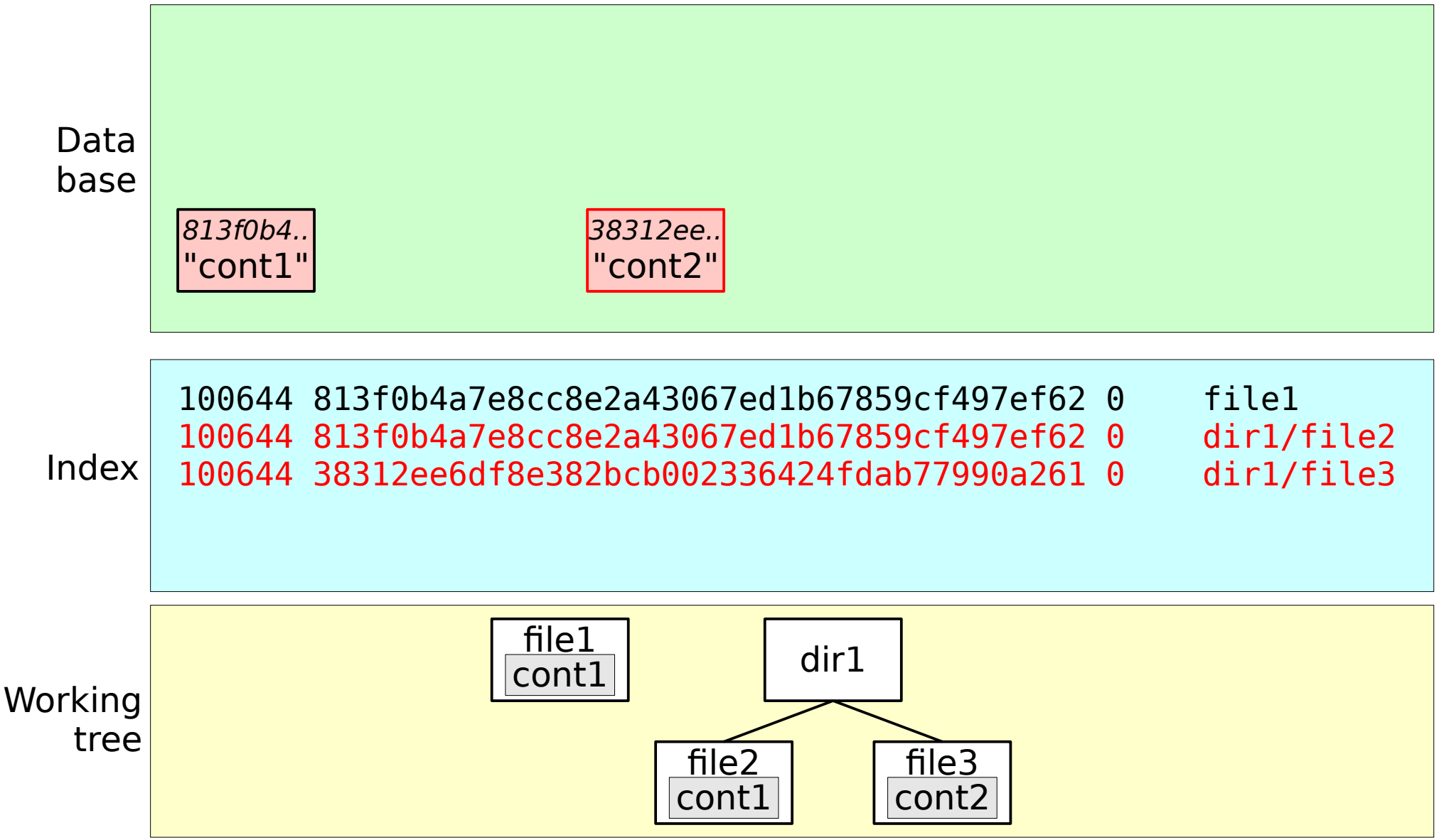
```
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    file1
```

Working tree



The Index inside

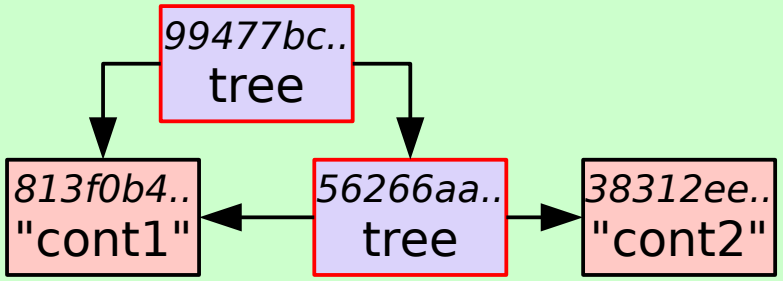
```
$ git add dir1
```



The Index inside

```
$ git write-tree
99477bc621eff4c48b4d34fb902...
```

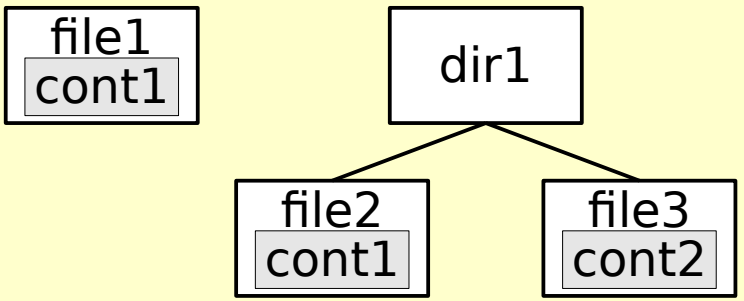
Data base



Index

```
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/file3
```

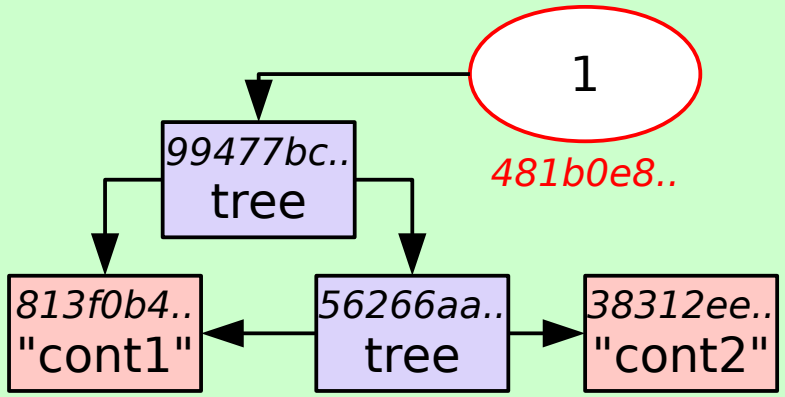
Working tree



The Index inside

```
$ git commit-tree 99477bc62...
example commit msg
^D
481b0e88217187b8a922f21d770...
```

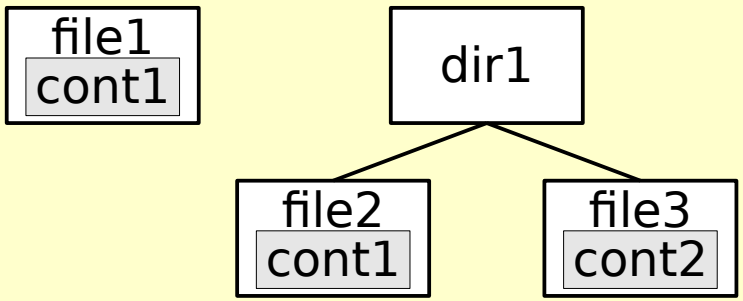
Data base



Index

```
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/file3
```

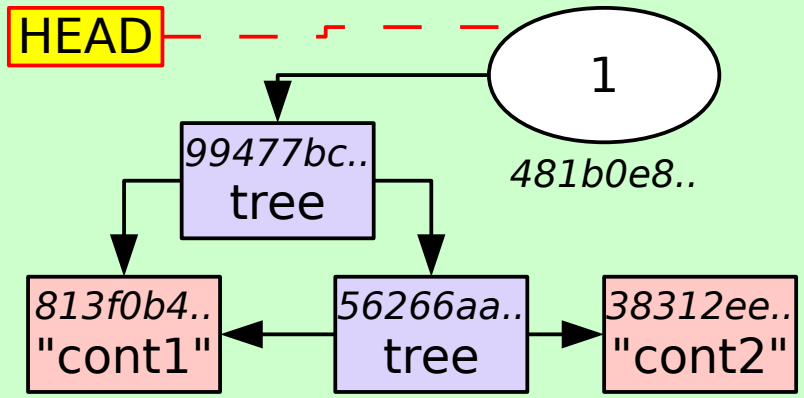
Working tree



The Index inside

```
$ git checkout 481b0e882171...
```

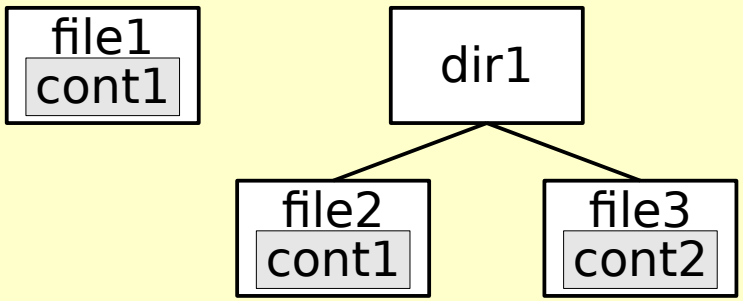
Data base



Index

```
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/file3
```

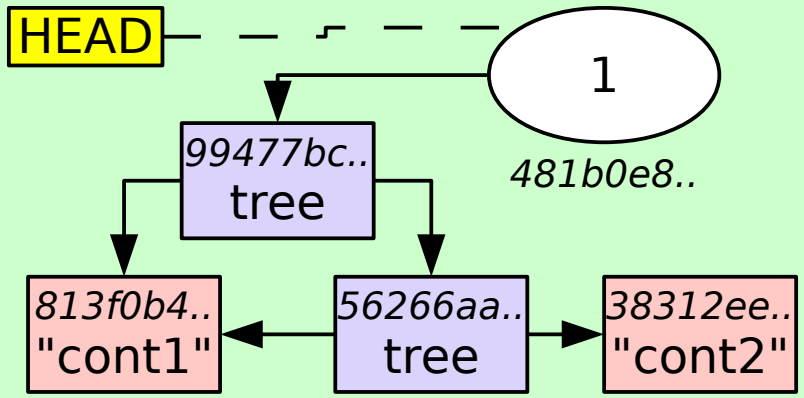
Working tree



The Index inside

```
$ mv dir1/file{3,N}
$ echo "moo" > file1
```

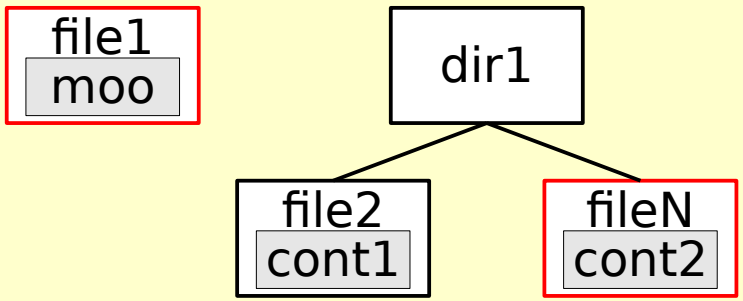
Data base



Index

```
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/file3
```

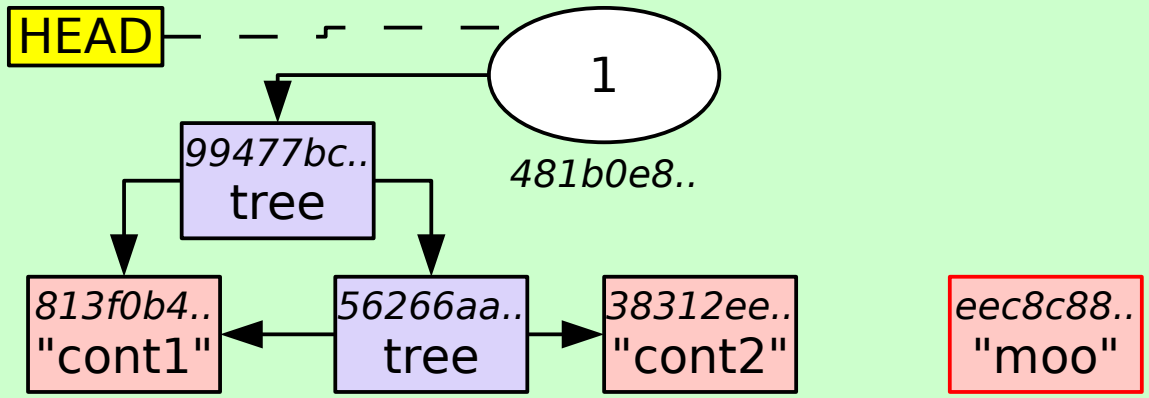
Working tree



The Index inside

```
$ git add .
```

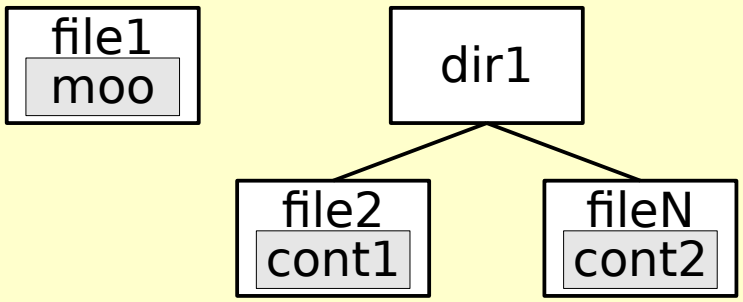
Data base



Index

```
100644 eec8c88a93f6ee1515fb8348f2c122cfda4302cd 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/file3
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/fileN
```

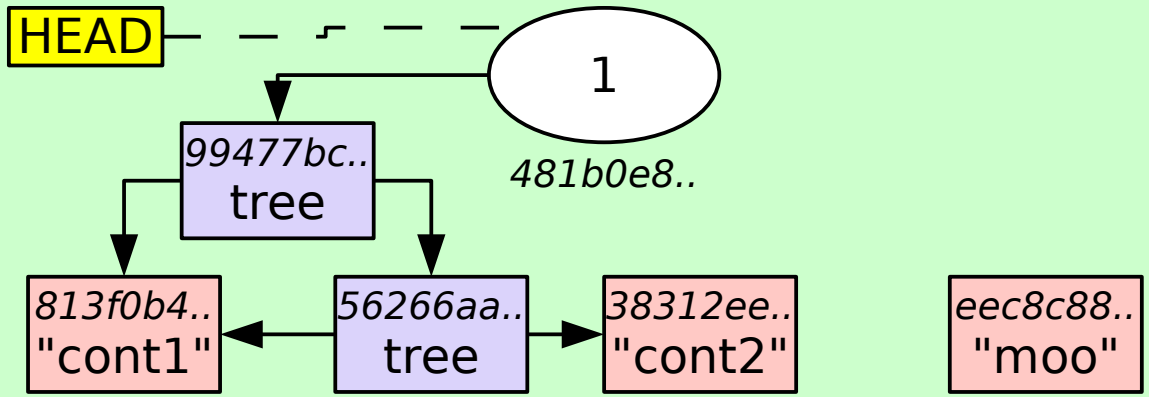
Working tree



The Index inside

```
$ git add -u
```

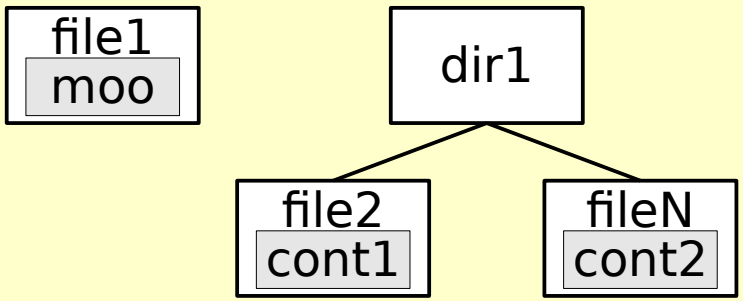
Data base



Index

```
100644 eec8c88a93f6ee1515fb8348f2c122cfda4302cd 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/fileN
```

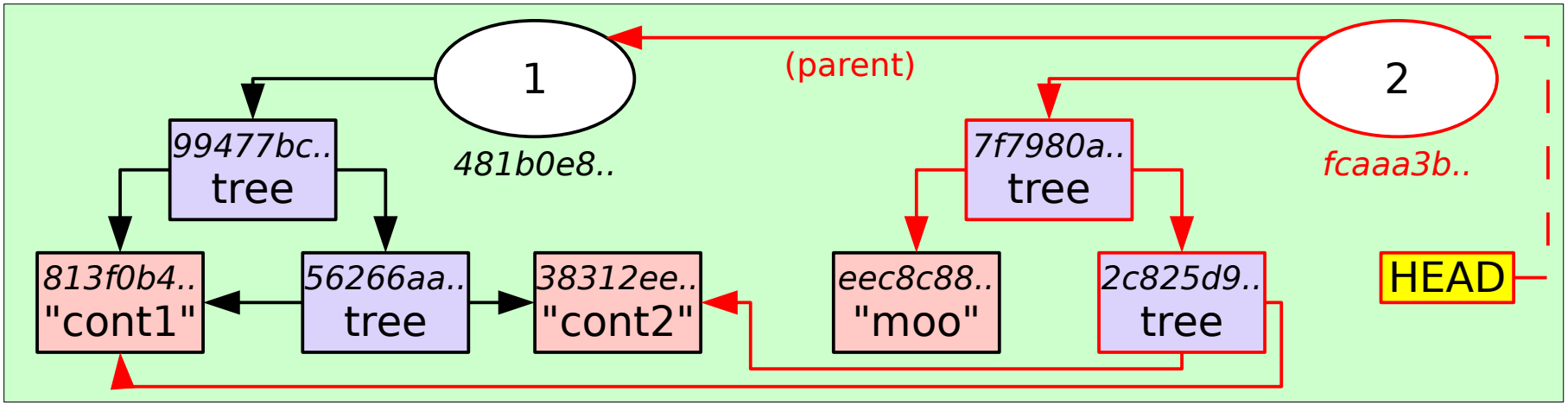
Working tree



The Index inside

```
$ git commit -m "another one"
[detached HEAD fcaaa3b] another one
 2 files changed, 1 insertion(+), 1
deletion(-)
 rename dir1/{file3 => fileN} (100%)
```

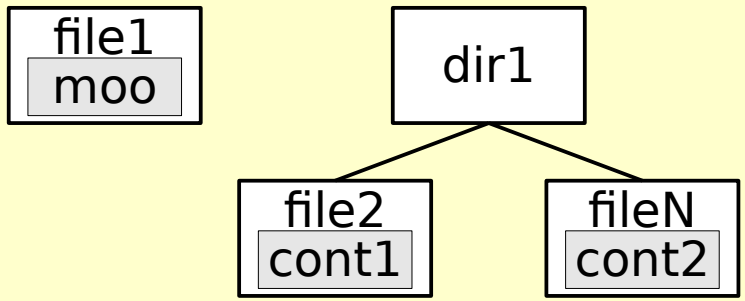
Data base



Index

```
100644 eec8c88a93f6ee1515fb8348f2c122cfda4302cd 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/fileN
```

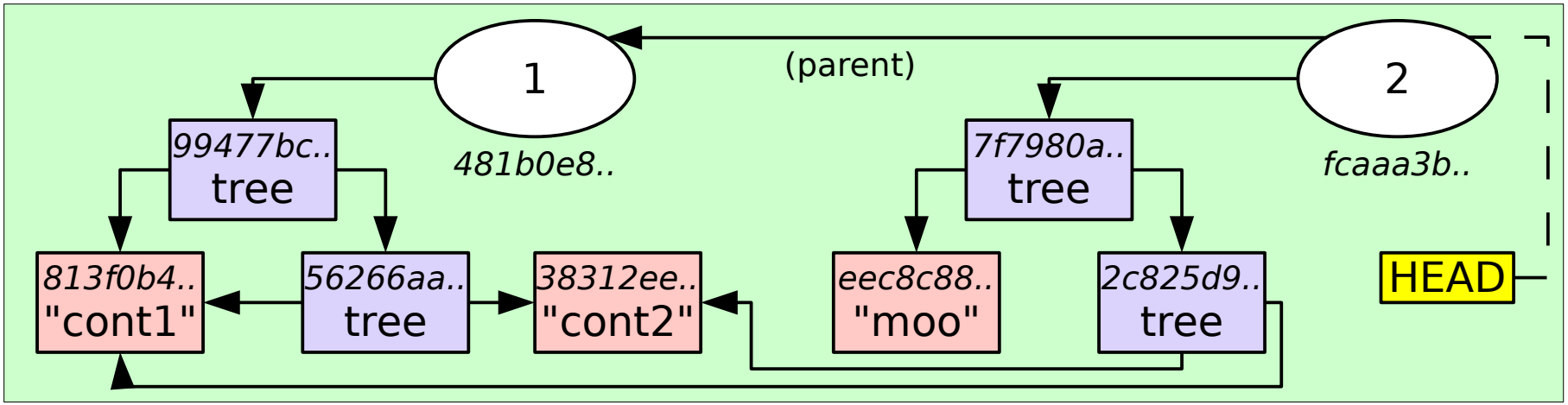
Working tree



The Index inside

```
$
```

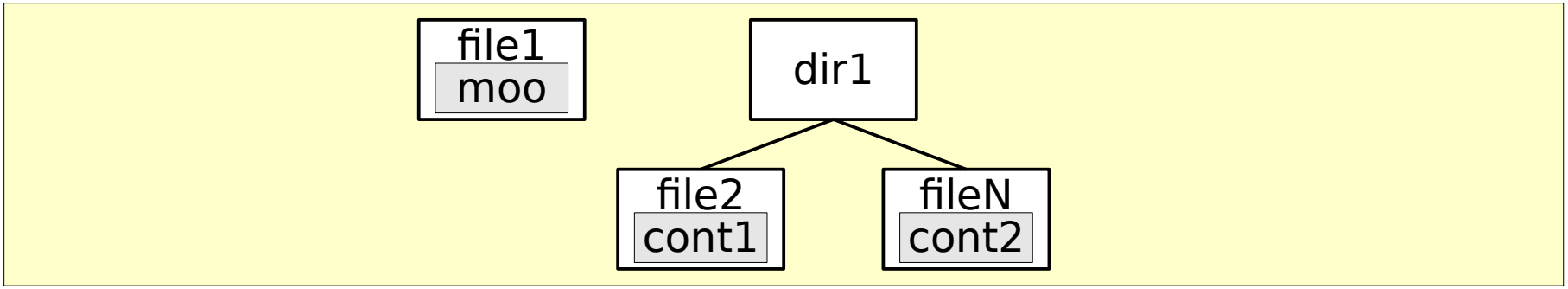
Data base



Index

```
100644 eec8c88a93f6ee1515fb8348f2c122cfda4302cd 0    file1
100644 813f0b4a7e8cc8e2a43067ed1b67859cf497ef62 0    dir1/file2
100644 38312ee6df8e382bcb002336424fdab77990a261 0    dir1/fileN
```

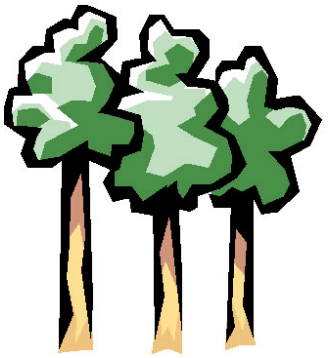
Working tree



Woking with Index

(manpages!)

Three trees of git



HEAD

Object database

Index

Working tree

Working with Index

- git-diff
  - git diff -- [path/]
    - prints out a diff of the Working tree, relative to the Index
  - git diff --cached [sha] -- [path/]
    - prints out a diff of the Index, relative to [sha] (or HEAD)
  - git diff [sha] -- [path/]
    - prints out a diff between the Working tree, relative to [sha] (or HEAD)

Working with Index

- git-add
  - git add [options] -- path/
    - adds path/ recursively to the Index
  - git add <options>
    - performs the specified option(s), such as -u|--update

Working with Index

- git-reset
  - git reset <--mode> [sha]
    - hard: makes current branch point to [sha] (or HEAD) and forces a checkout to Index and to the Working tree
    - mixed: makes current branch point to [sha] (or HEAD) and resets the Index to the [sha] state, leaving Working tree untouched
    - soft: makes current branch point to [sha] only, Index or Working tree are not touched
  - git reset [sha] -- path/to/file
    - copies path version at [sha] (or HEAD) to the Index, does not affect Working tree

Working with Index

- git-checkout
  - git checkout <branchname>
    - makes HEAD a symbolic link to <branchname>
    - loads root tree object from the referenced commit to the Index
    - calls git-checkout-index to update Working tree to match Index
  - git checkout -- path/to/file
    - calls git-checkout-index on the path
  - git checkout <sha> -- path/to/file
    - bypasses Index and checks out the path at <sha> state

Presentation: Questions?

Workshop >>

use workshop-02 repository/branch for this workshop

Workshop

goal: **split one large commit**,
4 commits with messages,
1 change in each

- See changes
  - git show
  - gitk
- Move HEAD back one commit
  - HEAD~1 means "referenced commit parent"
  - keep Working tree untouched (mixed reset)
- Add one change using `git add --patch`
  - (unstage accidents via `git reset --patch`)
  - split larger hunks with multiple changes using "s"
- Verify / sanity check
  - change to be commited: `git diff --cached`
  - remaining changes: `git diff`
- Commit the one change using git commit -v
  - use descriptive commit message
- Add another change
- View results via `git log -4 -p` or `gitk`

The End

Thanks for listening

Links

- Git-scm.com - docs & reference manual

```
http://git-scm.com/documentation
```

- Old "git book"
  - original @ github

```
https://github.com/schacon/gitbook
```

  - online!

```
http://vrac.cofares.net/git/book/index.html
```

- "Git for computer scientists" + rework

```
http://eagain.net/articles/git-for-computer-scientists/
http://sitaramc.github.com/gcs/
```