

Version Control with **git**

Advanced Programming
Techniques
Fall 2013

What is version control?

- ❖ Track the history of a collection of files—most often source code.
- ❖ Allows us to:
 - ⊗ See what files changed and when they changed.
 - ▮ Compare (diff) two or more versions.
 - ▮ Recover (check out) old version.
 - ▮ Experiment with new ideas without the risk of losing existing work (branching).
- ❖ Greatly facilitates collaboration.

Why version control?

- ❖ Have you ever taken a working program and introduced a bug?
- ❖ Have you ever taken a working program and introduced a bug *but not noticed immediately*?

Version control with **git**

- ◆ Many version control systems: bazaar, cvs, darcs, mercurial, Perforce, subversion, Visual SourceSafe, etc.
- ◆ But **git** has (largely) won.
 - Developed to manage Linux kernel source code.
 - Popularized by github.
 - Widely used

Google

facebook

Microsoft

twitter

LinkedIn

NETFLIX



PostgreSQL

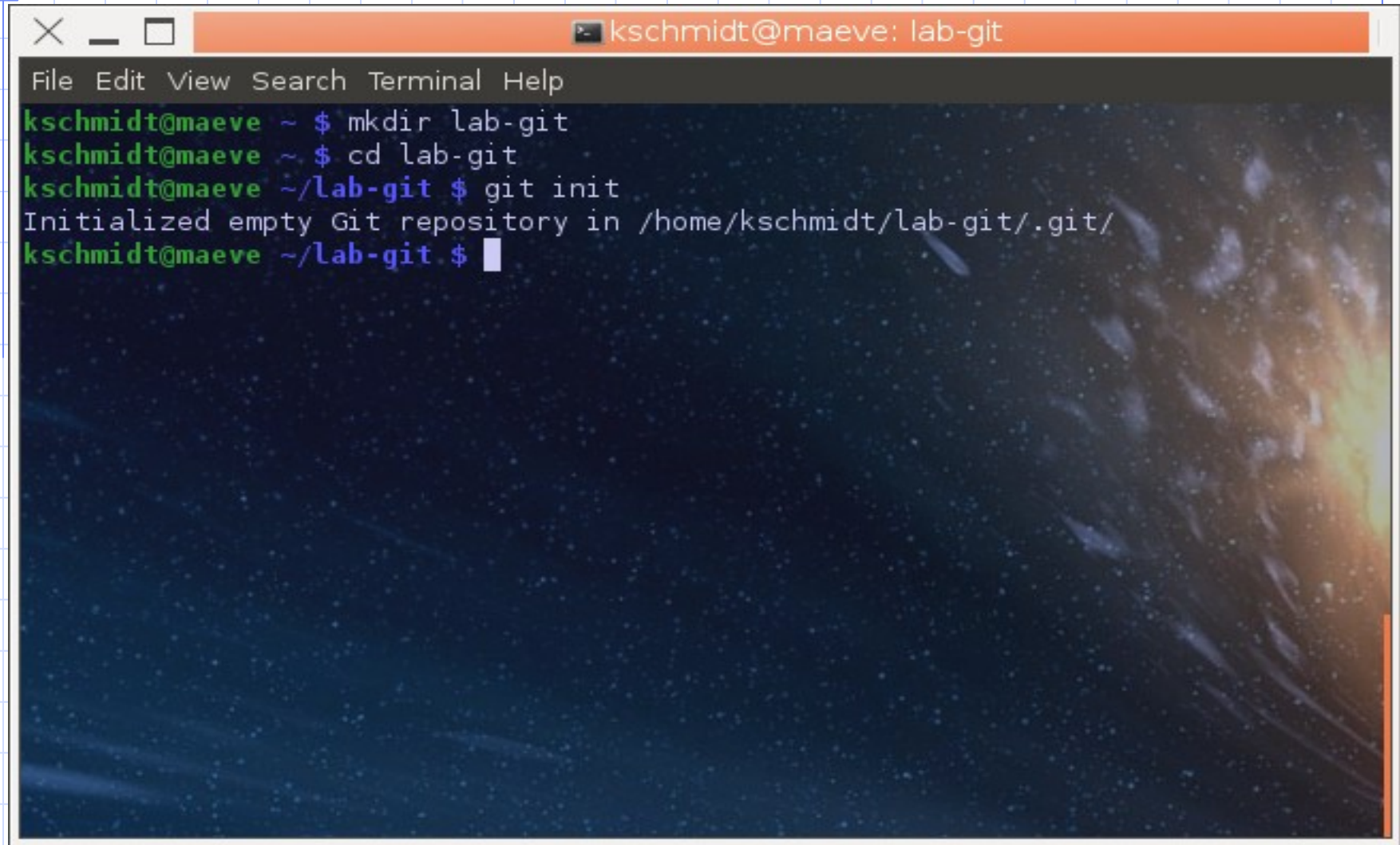


git mini-lab – HOLD (TODO)



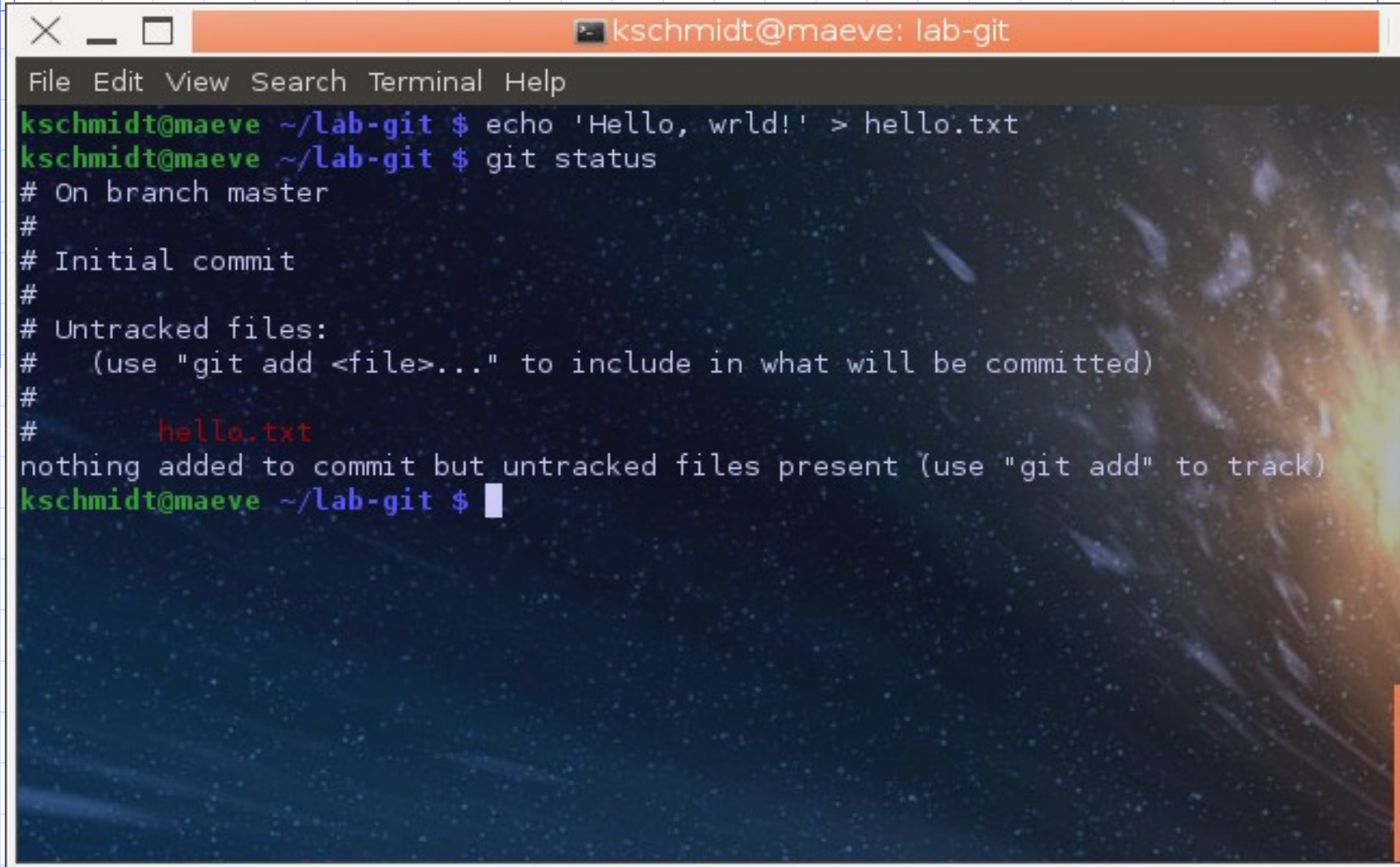
◆ For *this portion of the lab only*, you will submit your work by checking it into a **git** repository that lives on **tux**.

Creating a Repository – git init

A terminal window titled 'kschmidt@maeve: lab-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The background of the terminal is a dark space with a bright nebula on the right. The terminal shows the following commands and output:

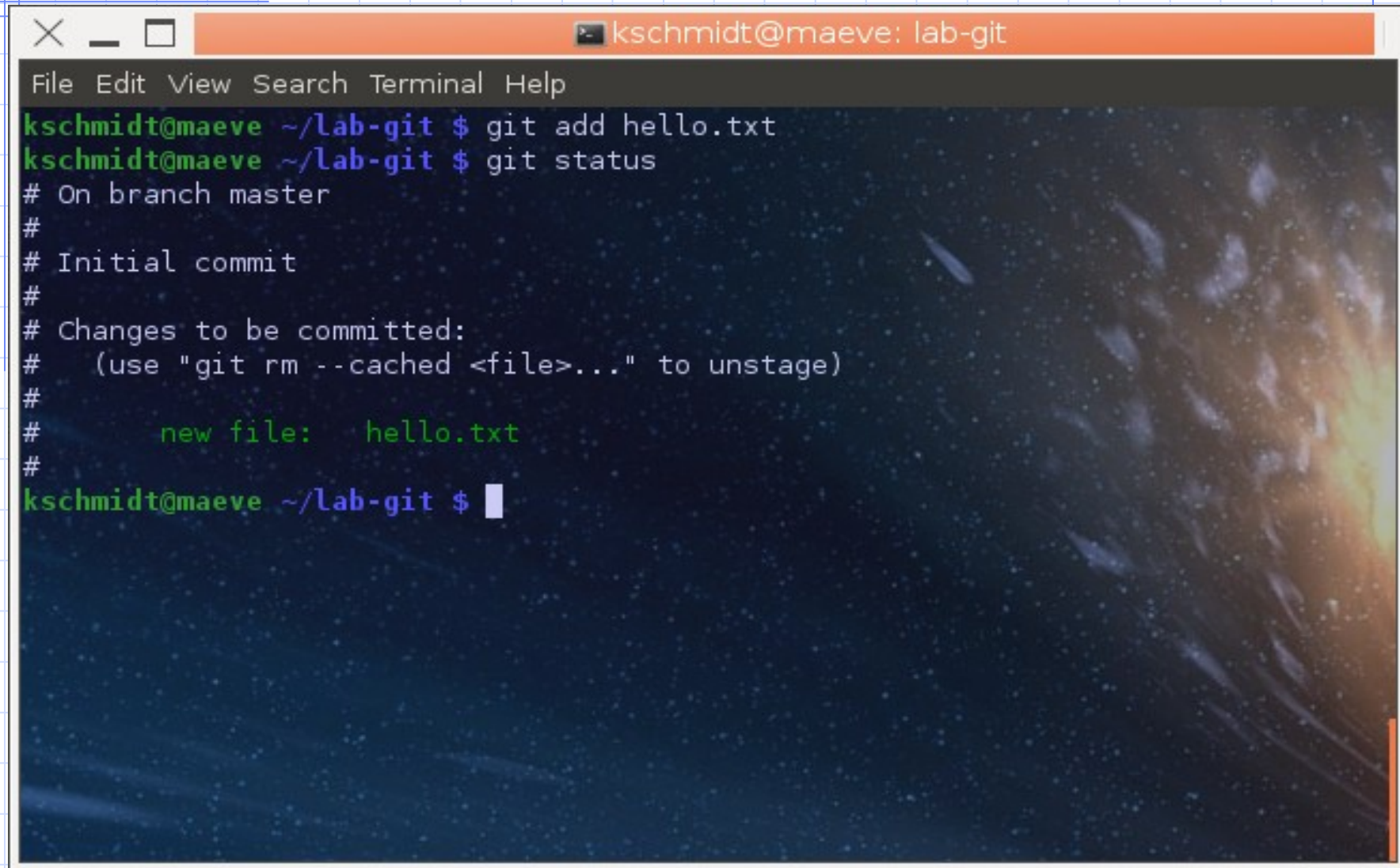
```
kschmidt@maeve ~ $ mkdir lab-git
kschmidt@maeve ~ $ cd lab-git
kschmidt@maeve ~/lab-git $ git init
Initialized empty Git repository in /home/kschmidt/lab-git/.git/
kschmidt@maeve ~/lab-git $
```

Adding a file – git add

A terminal window titled 'kschmidt@maeve: lab-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'echo 'Hello, wrld!' > hello.txt' and 'git status'. The output of 'git status' indicates an initial commit on the master branch with one untracked file, 'hello.txt'.

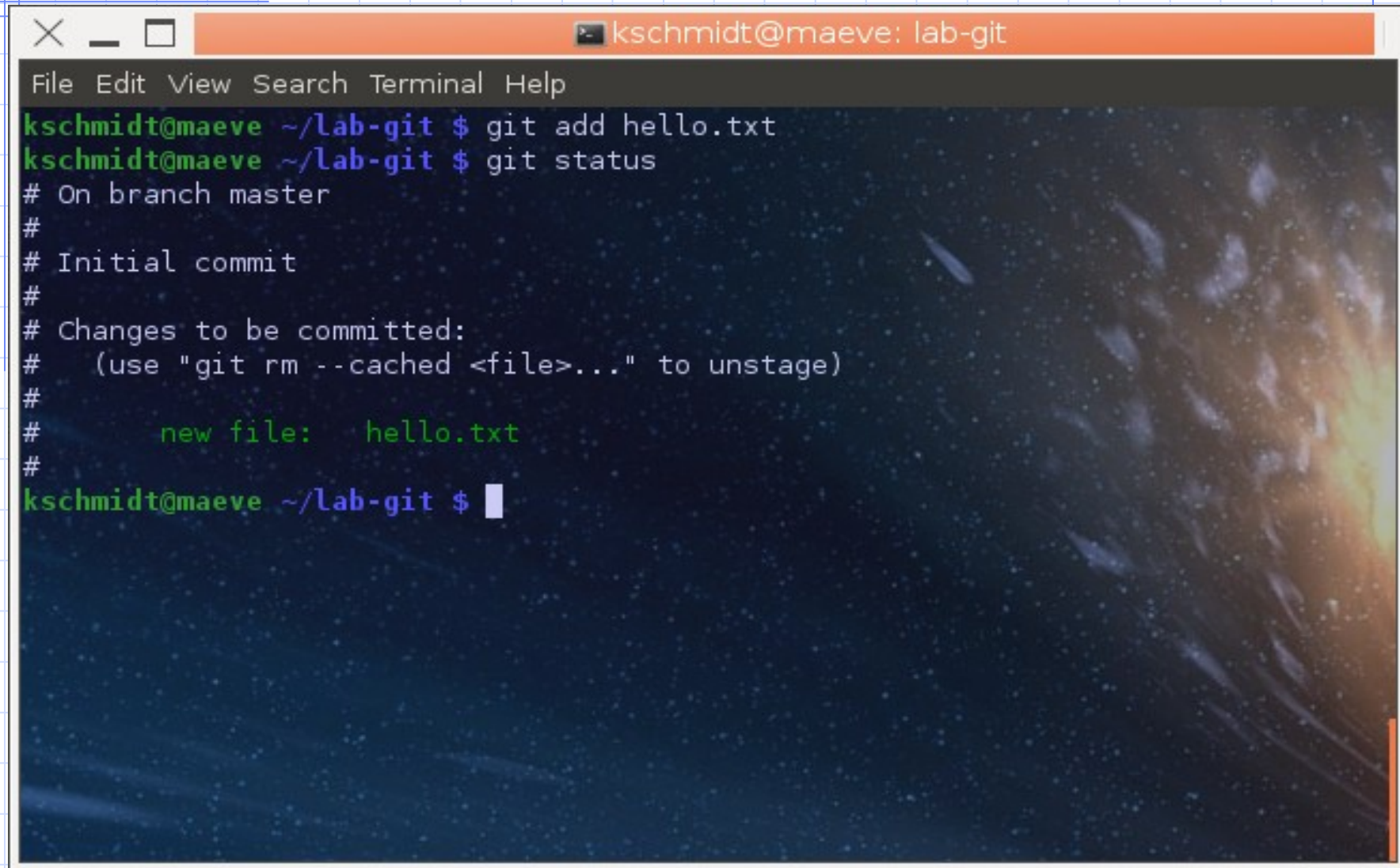
```
kschmidt@maeve ~/lab-git $ echo 'Hello, wrld!' > hello.txt
kschmidt@maeve ~/lab-git $ git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       hello.txt
nothing added to commit but untracked files present (use "git add" to track)
kschmidt@maeve ~/lab-git $
```


Adding a file (cont'd)

A terminal window titled 'kschmidt@maeve: lab-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'git add hello.txt' and 'git status'. The status output indicates an initial commit on the master branch with one new file staged for commit. The background of the terminal is a dark space with a bright nebula on the right side.

```
kschmidt@maeve ~/lab-git $ git add hello.txt
kschmidt@maeve ~/lab-git $ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   hello.txt
#
kschmidt@maeve ~/lab-git $
```


Adding a file (cont'd)

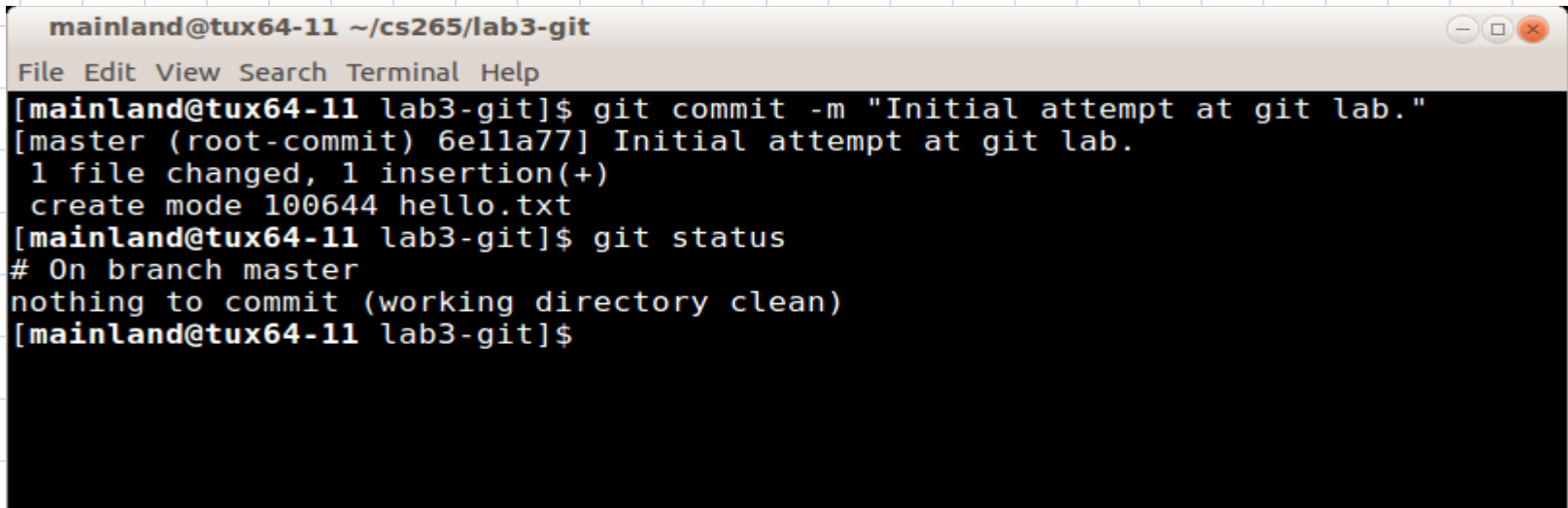
A terminal window titled 'kschmidt@maeve: lab-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'git add hello.txt' and 'git status'. The status output indicates an initial commit on the master branch with one new file staged for commit: 'hello.txt'.

```
kschmidt@maeve ~/lab-git $ git add hello.txt
kschmidt@maeve ~/lab-git $ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   hello.txt
#
kschmidt@maeve ~/lab-git $
```

Committing our work

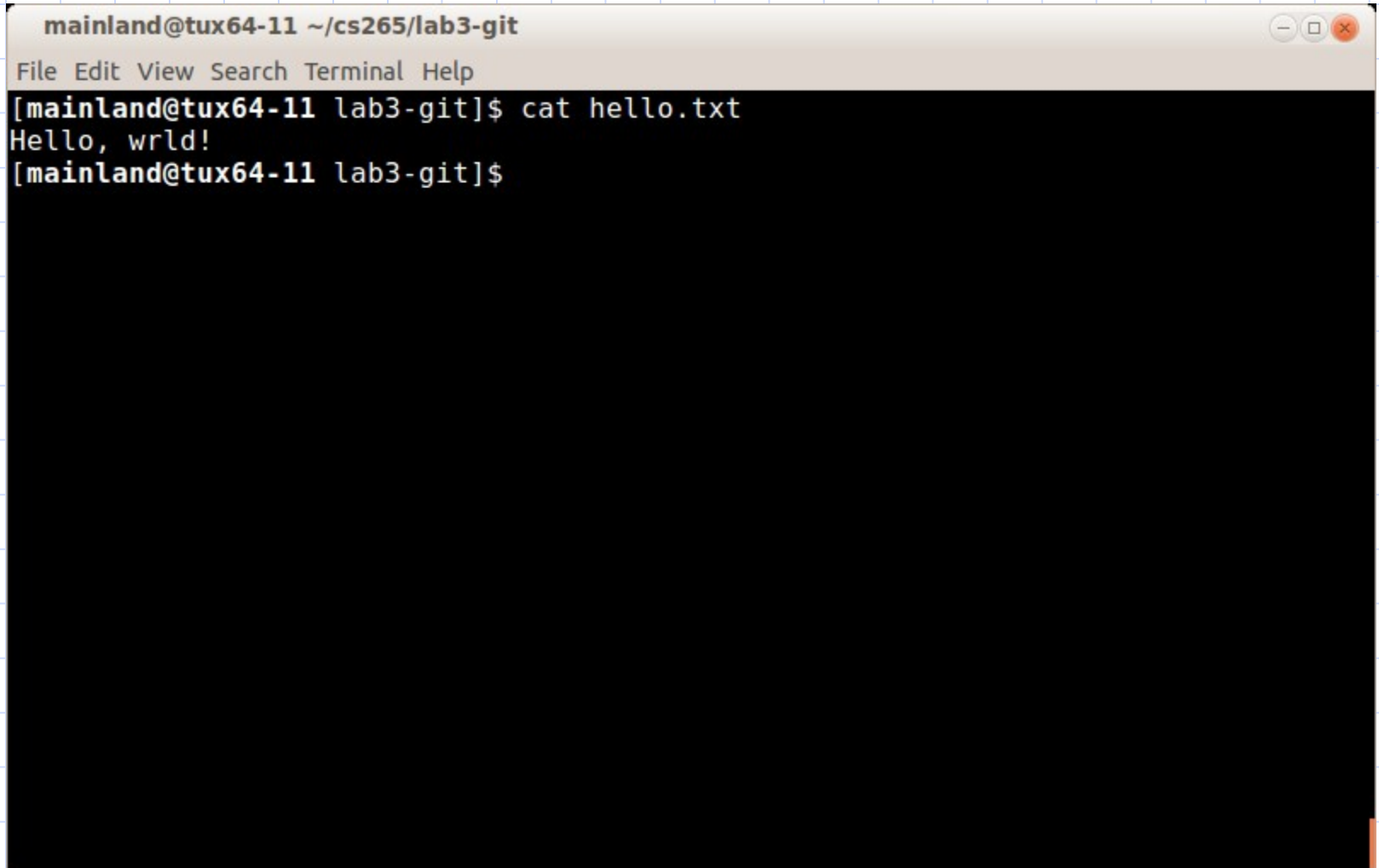
`git commit [-m <msg>]`

- Commits changes to the working directory into the repository
- If a msg is not supplied, the log will be opened in an editor

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the execution of 'git commit -m "Initial attempt at git lab."' which results in a commit on the master branch with hash 6ella77. It then shows 'git status' which reports that the working directory is clean.

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ git commit -m "Initial attempt at git lab."
[master (root-commit) 6ella77] Initial attempt at git lab.
1 file changed, 1 insertion(+)
create mode 100644 hello.txt
[mainland@tux64-11 lab3-git]$ git status
# On branch master
nothing to commit (working directory clean)
[mainland@tux64-11 lab3-git]$
```

Oops...

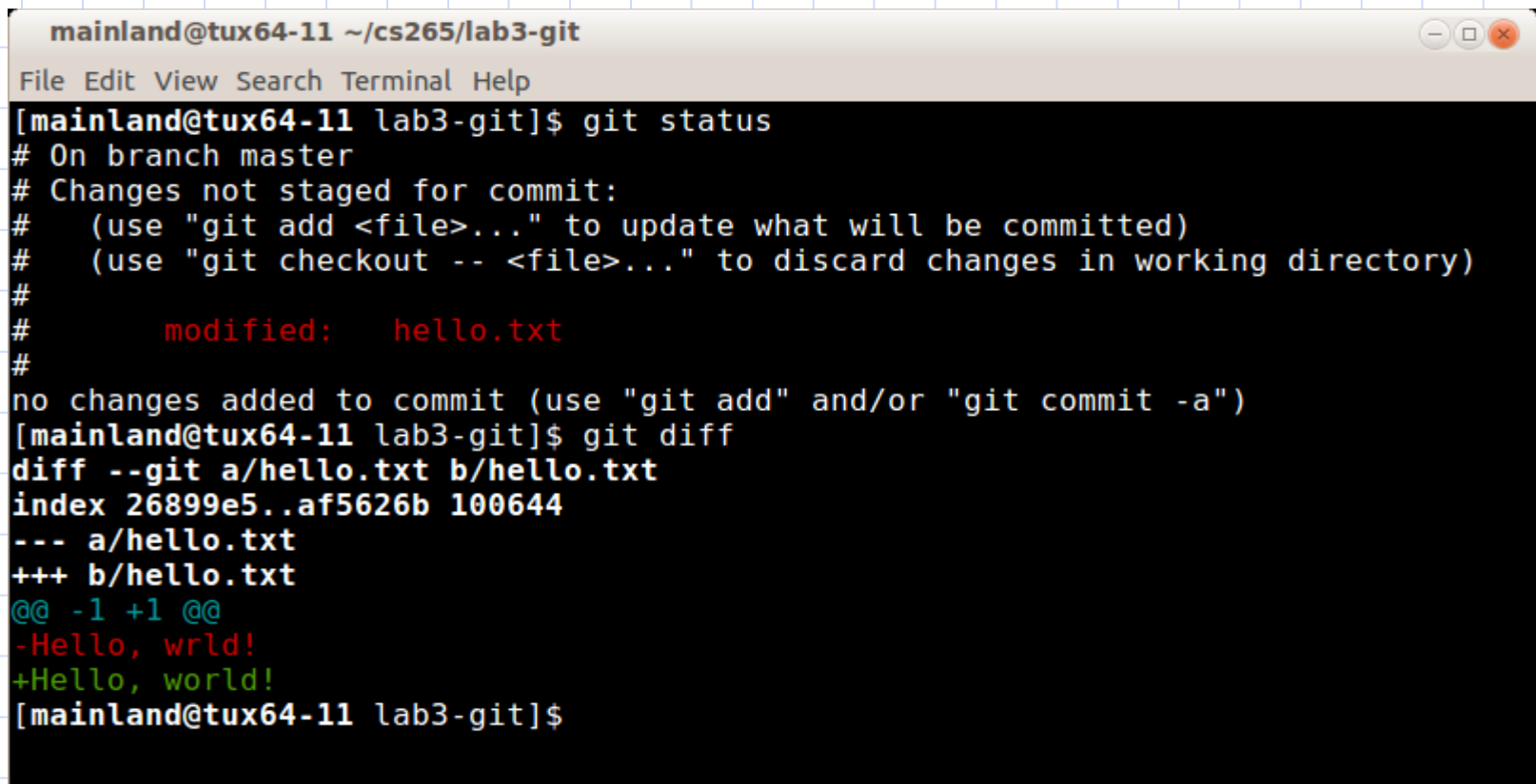


```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ cat hello.txt
Hello, wrld!
[mainland@tux64-11 lab3-git]$
```


That's better...

- Changes to current working directory can be compared to the repository

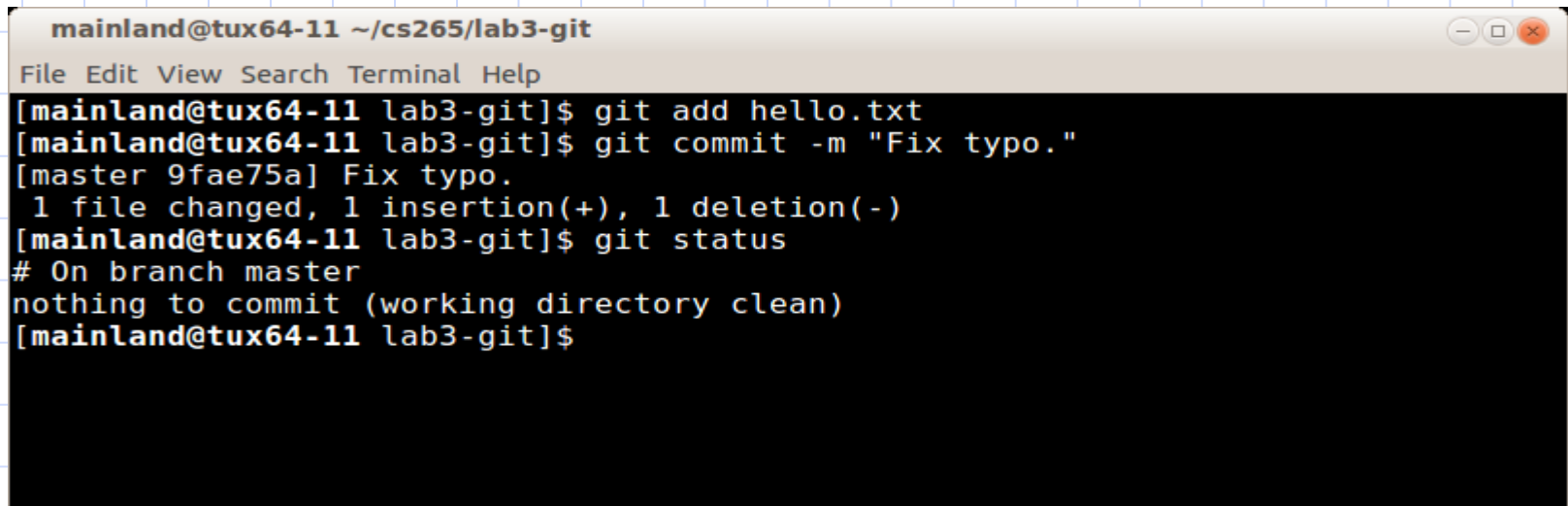
`git diff`

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with standard window controls. The terminal shows the output of 'git status' and 'git diff'. The 'git status' output indicates that 'hello.txt' is modified. The 'git diff' output shows a line-by-line comparison of 'a/hello.txt' and 'b/hello.txt', highlighting the change from 'Hello, wrld!' to 'Hello, world!'.

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   hello.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
[mainland@tux64-11 lab3-git]$ git diff
diff --git a/hello.txt b/hello.txt
index 26899e5..af5626b 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,1 @@
-Hello, wrld!
+Hello, world!
[mainland@tux64-11 lab3-git]$
```

Committing changes

- Changes to files must be added to the index
- Changes to known files can also be added with the update flag
`git add -u`
- Then commit changes

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[mainland@tux64-11 lab3-git]$ git add hello.txt
[mainland@tux64-11 lab3-git]$ git commit -m "Fix typo."
[master 9fae75a] Fix typo.
 1 file changed, 1 insertion(+), 1 deletion(-)
[mainland@tux64-11 lab3-git]$ git status
# On branch master
nothing to commit (working directory clean)
[mainland@tux64-11 lab3-git]$
```


Examining what we've done.



```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ git log
commit 9fae75a3fc26aeda7b899542c805e5e7627668d7
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:34:43 2013 -0400

    Fix typo.

commit 6e11a7762d6faa215a4e0d4009d1170fef4730ee
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:32:37 2013 -0400

    Initial attempt at git lab.
[mainland@tux64-11 lab3-git]$
```

Examining what we've done.

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ git log
commit 9fae75a3fc26aeda7b899542c805e5e7627668d7
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:34:43 2013 -0400

    Fix typo.

commit 6e11a7762d6faa215a4e0d4009d1170fef4730ee
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:32:37 2013 -0400

    Initial attempt at git lab.
[mainland@tux64-11 lab3-git]$ git diff 6e11a7..9fae75
diff --git a/hello.txt b/hello.txt
index 26899e5..af5626b 100644
--- a/hello.txt
+++ b/hello.txt
@@ -1,1 @@
-Hello, wrld!
+Hello, world!
[mainland@tux64-11 lab3-git]$
```

Summary

Initialize a new repo	<code>git init</code>
Check repo status	<code>git status</code>
Add a file	<code>git add <file></code>
Commit changes	<code>git commit</code> <code>git commit -m</code> <code><msg></code>
See commit history	<code>git log</code>
Diff versions	<code>git diff</code> <code>git diff A..B</code>

Getting help in **git**



`git help <topic>`

◆ git will display help for this topic

Some **git** concepts

- ◆ The **working directory** is the directory where you've checked out code.
- ◆ The **index** is the set of changes that are **staged** and will become the next commit.
 - The index may be different from the working directory!

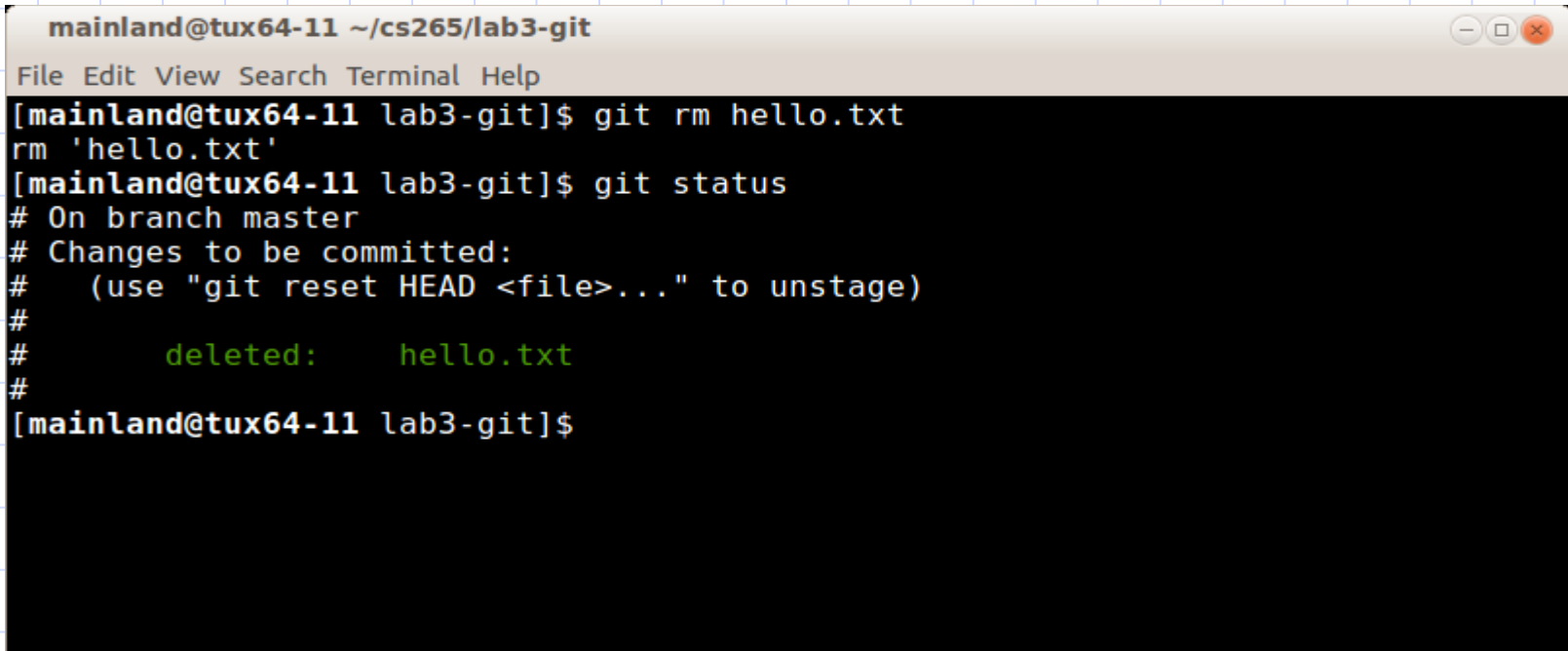
Removing a file

- ❖ Just delete the file, add the change to the index, then commit, OR....

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ rm -f hello.txt
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    hello.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
[mainland@tux64-11 lab3-git]$ git add -u
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       deleted:    hello.txt
#
[mainland@tux64-11 lab3-git]$
```


Removing a file, take 2

- ◆ Use git to remove the file
`git rm <file>`
- ◆ (Then commit the changes to the repository)

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with standard window controls. The terminal shows the execution of 'git rm hello.txt' and 'git status'. The status output indicates that 'hello.txt' has been deleted and is staged for commit. The prompt returns to the shell.

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ git rm hello.txt
rm 'hello.txt'
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       deleted:    hello.txt
#
[mainland@tux64-11 lab3-git]$
```

Undoing mistakes

- ◆ git reset removes all pending changes (clears the index)

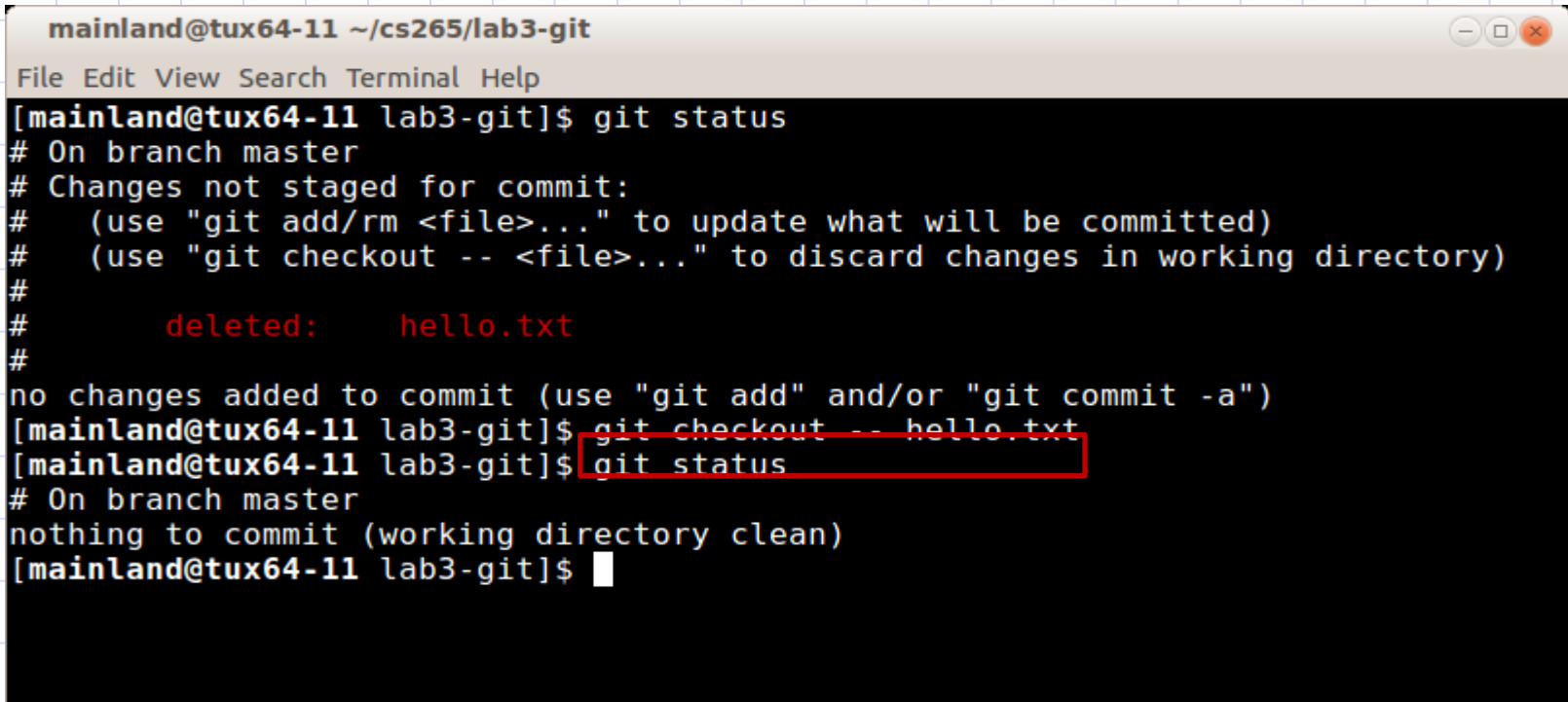
Does *not* touch the files in the working directory

```
mainland@tux64-11 ~7cs265/lab3-git
File Edit View Search Terminal Help

[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       deleted:    hello.txt
#
[mainland@tux64-11 lab3-git]$ git reset
Unstaged changes after reset:
D       hello.txt
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    hello.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
[mainland@tux64-11 lab3-git]$
```

Undoing mistakes, cont'd

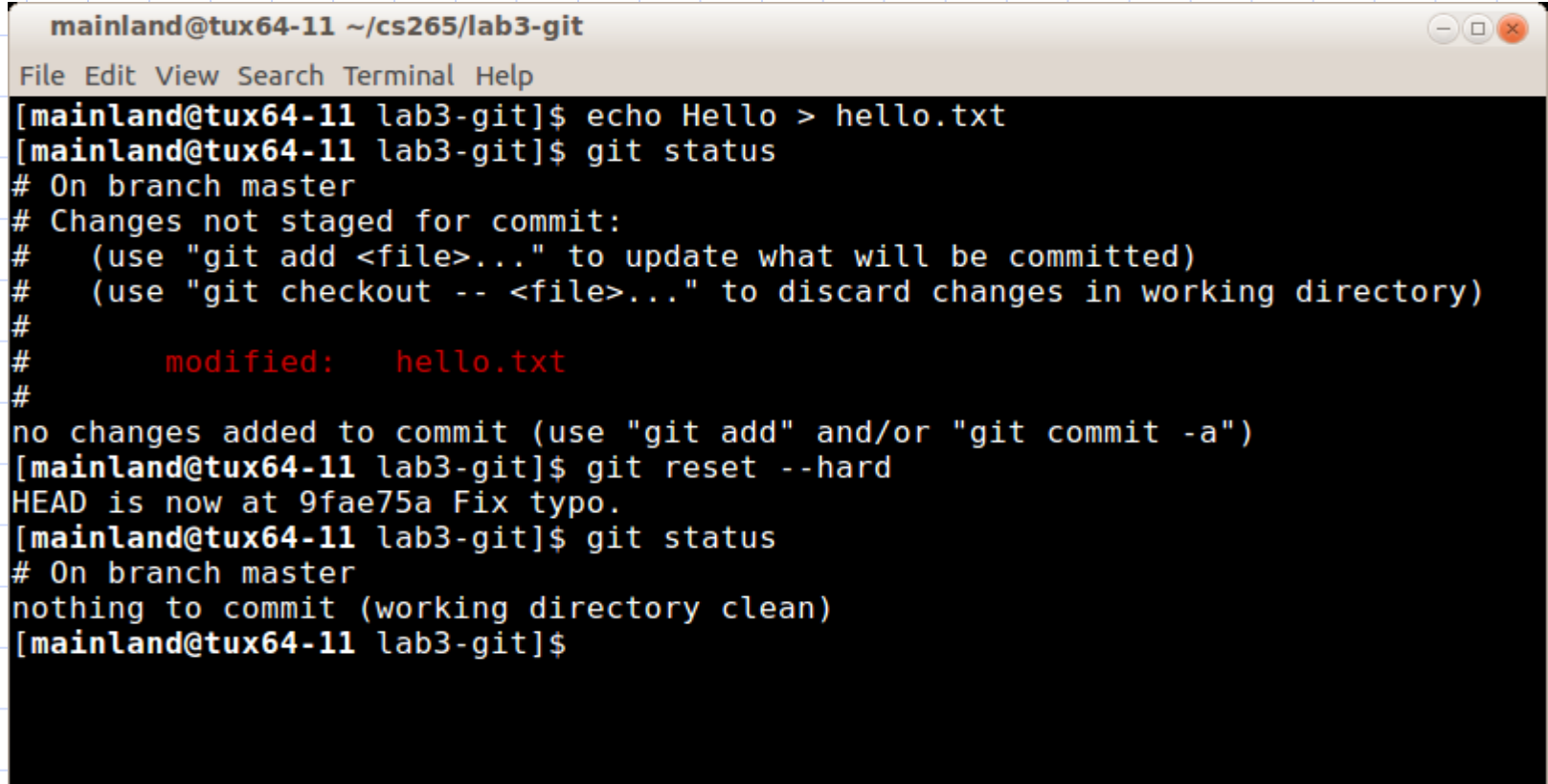
- ◆ git checkout can be used to pull the last (committed) version from the repository to the working directory

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the output of 'git status' indicating a deletion of 'hello.txt'. Then, the command 'git checkout -- hello.txt' is entered and highlighted with a red box. Finally, 'git status' is entered, showing the working directory is clean.

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:      hello.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
[mainland@tux64-11 lab3-git]$ git checkout -- hello.txt
[mainland@tux64-11 lab3-git]$ git status
# On branch master
nothing to commit (working directory clean)
[mainland@tux64-11 lab3-git]$
```

Reverting all changes

- ◆ `git reset --hard` puts the repository and working directory back to the last committed state

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following sequence of commands and output:

```
[mainland@tux64-11 lab3-git]$ echo Hello > hello.txt
[mainland@tux64-11 lab3-git]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   hello.txt
#
no changes added to commit (use "git add" and/or "git commit -a")
[mainland@tux64-11 lab3-git]$ git reset --hard
HEAD is now at 9fae75a Fix typo.
[mainland@tux64-11 lab3-git]$ git status
# On branch master
nothing to commit (working directory clean)
[mainland@tux64-11 lab3-git]$
```

Summary

Remove a file

```
git rm <FILE>
```

Add changes to
index

```
git add -u
```

Reset index

```
git reset
```

Discard changes
to working
directory

```
git checkout -- <FILE>
```

Discard all
changes

```
git reset --hard
```

git is distributed

- ◆ Peer-to-peer vs. client-server.
- ◆ There is no “central” repository (except by convention).
- ◆ Each developer's copy of the code is a real repository.
- ◆ Synchronize by exchanging *patches* (sets of changes) with other developers.

Repositories and Branches

- ❖ A repository contains one or more *branches*
- ❖ `git branch [-a|-r]`
- ❖ Separate branches are useful
 - Maintaining several releases of a product
 - Experiment, add features, take it for a spin before merging new feature into release
- ❖ **Not** this week. Nor this term. Plenty of tutorials out there.

Creating a **bare** repository

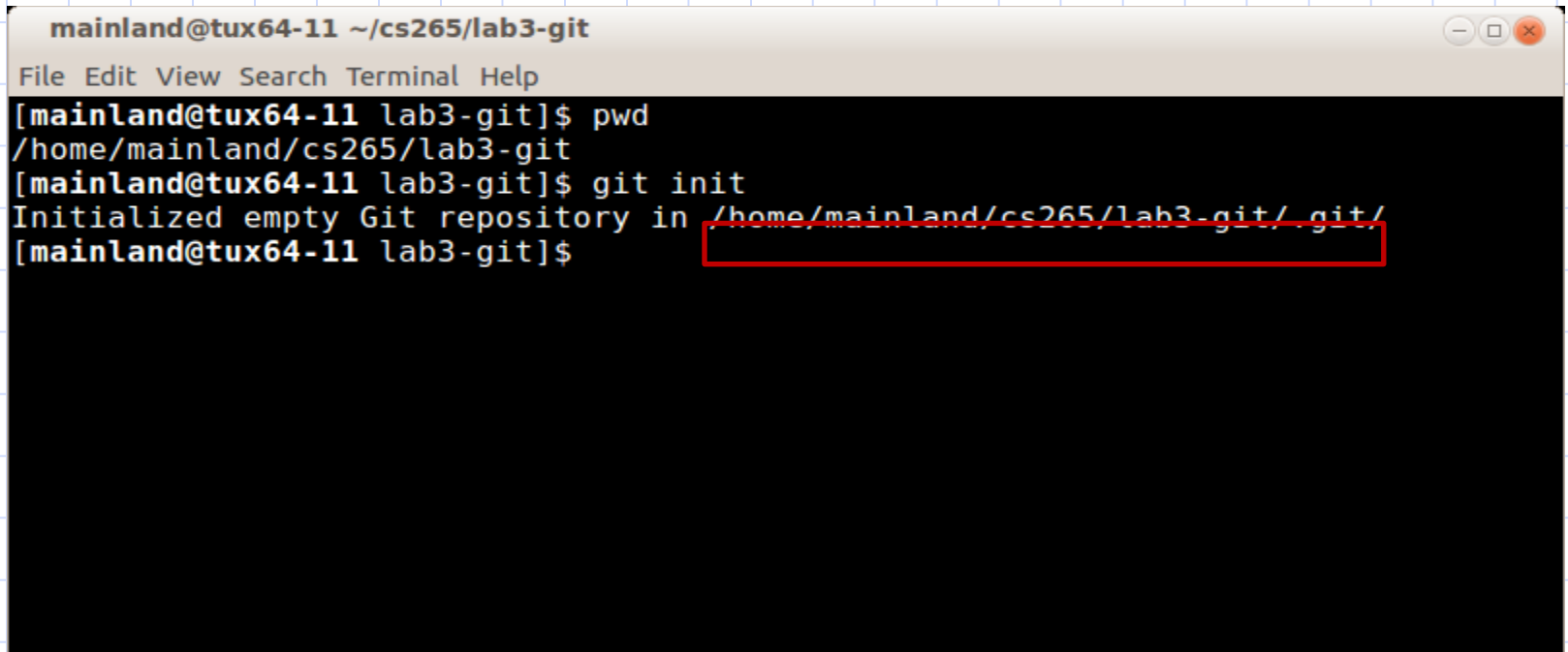
- ◆ Bare means no working directory.
- ◆ Useful for coordinating the movement of patches between other repositories.
- ◆ In this example, this will be considered to be the main

(authoritative) repository

```
mainland@tux64-11 ~/cs265
File Edit View Search Terminal Help
[mainland@tux64-11 cs265]$ pwd
/home/mainland/cs265
[mainland@tux64-11 cs265]$ ls
lab3-git
[mainland@tux64-11 cs265]$ git init --bare lab3-git-bare
Initialized empty Git repository in /home/mainland/cs265/lab3-git-bare/
[mainland@tux64-11 cs265]$
```

Let's compare...

- ◆ Remember when we created lab3-git

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with standard window controls. The terminal shows the following commands and output:

```
[mainland@tux64-11 lab3-git]$ pwd
/home/mainland/cs265/lab3-git
[mainland@tux64-11 lab3-git]$ git init
Initialized empty Git repository in /home/mainland/cs265/lab3-git/.git/
[mainland@tux64-11 lab3-git]$
```

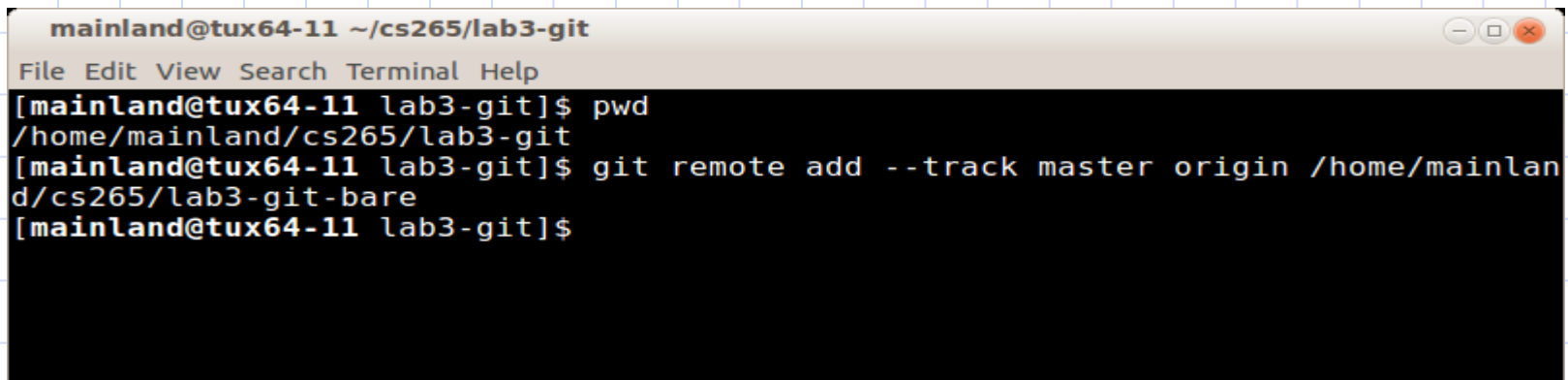
The path '/home/mainland/cs265/lab3-git/.git/' in the output of 'git init' is highlighted with a red rectangular box.

Adding a remote

- Distant repository to synch to
 - Others can pull from this, add work

`git remote add --track <branch> <label> <location>`

- <branch> is the distant branch to be tied to the current local one (master)
- <label> is the local name (alias) of the distant repository
- <location> is the path, URL, etc.

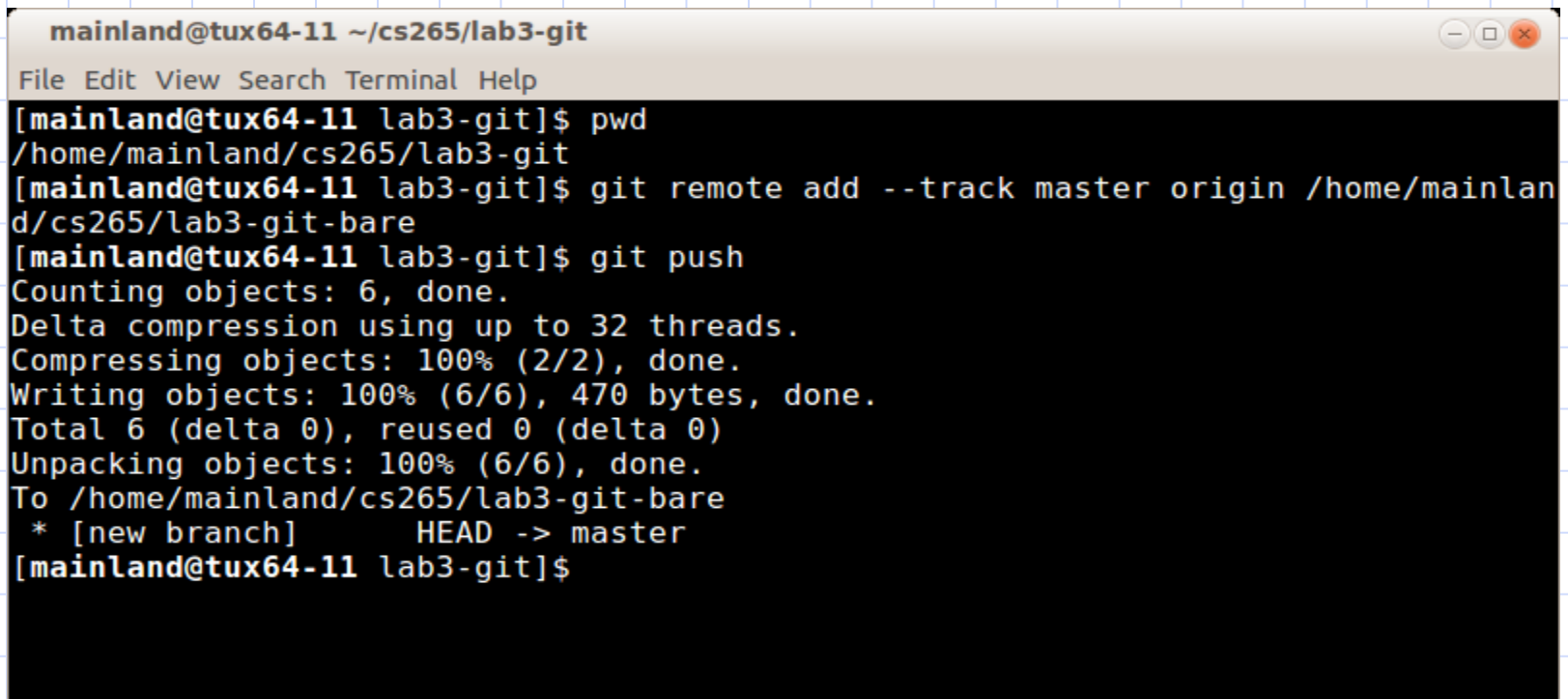
A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[mainland@tux64-11 lab3-git]$ pwd
/home/mainland/cs265/lab3-git
[mainland@tux64-11 lab3-git]$ git remote add --track master origin /home/mainland/cs265/lab3-git-bare
[mainland@tux64-11 lab3-git]$
```

Pushing to a remote

- git push origin master

- ◆ To make the remote branch the upstream branch for local:
git branch --set-upstream origin/master

A terminal window titled 'mainland@tux64-11 ~/cs265/lab3-git' with standard window controls. The terminal shows the following commands and output:

```
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git]$ pwd
/home/mainland/cs265/lab3-git
[mainland@tux64-11 lab3-git]$ git remote add --track master origin /home/mainland/cs265/lab3-git-bare
[mainland@tux64-11 lab3-git]$ git push
Counting objects: 6, done.
Delta compression using up to 32 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 470 bytes, done.
Total 6 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (6/6), done.
To /home/mainland/cs265/lab3-git-bare
 * [new branch]      HEAD -> master
[mainland@tux64-11 lab3-git]$
```

git branches

- ◆ Branches allow multiple lines of development to occur in the same repository.
- ◆ The default branch is named **master**.
- ◆ The **--track master** argument to **git remote add** makes **master** the default remote branch for pushing.

Cloning a repository

□ If somebody else, e.g., wants to join the project

```
mainland@tux64-11 ~/lab3-git-copy
File Edit View Search Terminal Help
[mainland@tux64-11 ~]$ git clone /home/mainland/cs265/lab3-git-bare lab3-git-copy
Cloning into 'lab3-git-copy'...
done.
[mainland@tux64-11 ~]$ cd lab3-git-copy
[mainland@tux64-11 lab3-git-copy]$ ls
hello.txt
[mainland@tux64-11 lab3-git-copy]$ git log
commit 9fae75a3fc26aeda7b899542c805e5e7627668d7
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:34:43 2013 -0400

    Fix typo.

commit 6e11a7762d6faa215a4e0d4009d1170fef4730ee
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:32:37 2013 -0400

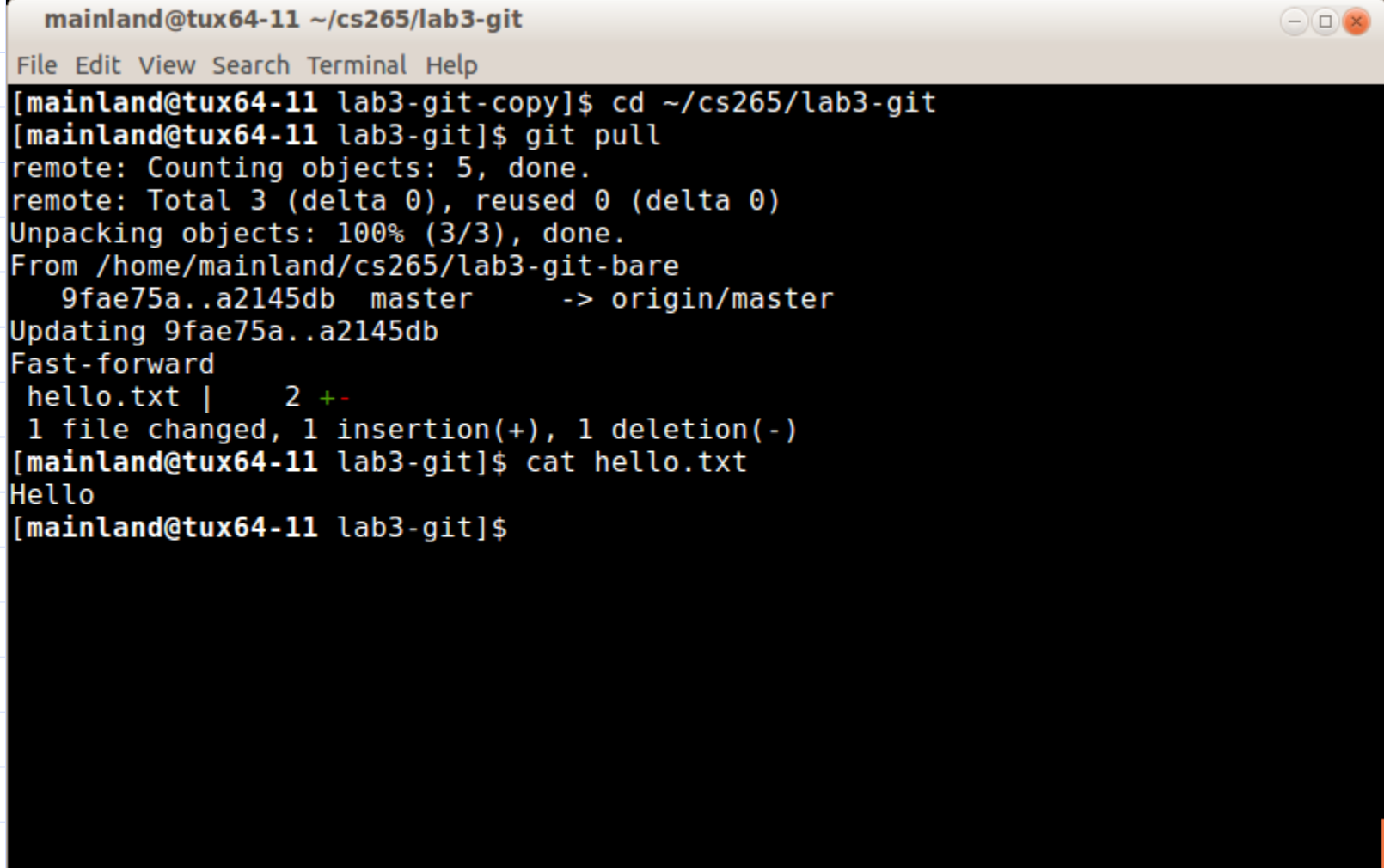
    Initial attempt at git lab.
[mainland@tux64-11 lab3-git-copy]$
```

Pushing changes



```
mainland@tux64-11 ~/lab3-git-copy
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git-copy]$ echo "Hello" >hello.txt
[mainland@tux64-11 lab3-git-copy]$ git add -u
[mainland@tux64-11 lab3-git-copy]$ git commit -m "Pithy."
[master a2145db] Pithy.
 1 file changed, 1 insertion(+), 1 deletion(-)
[mainland@tux64-11 lab3-git-copy]$ git push
Counting objects: 5, done.
Writing objects: 100% (3/3), 249 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
To /home/mainland/cs265/lab3-git-bare
 9fae75a..a2145db HEAD -> master
[mainland@tux64-11 lab3-git-copy]$
```

Pulling changes



```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help
[mainland@tux64-11 lab3-git-copy]$ cd ~/cs265/lab3-git
[mainland@tux64-11 lab3-git]$ git pull
remote: Counting objects: 5, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From /home/mainland/cs265/lab3-git-bare
  9fae75a..a2145db  master    -> origin/master
Updating 9fae75a..a2145db
Fast-forward
 hello.txt |      2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
[mainland@tux64-11 lab3-git]$ cat hello.txt
Hello
[mainland@tux64-11 lab3-git]$
```

Summary



Initialize a new bare repo **git init --bare**

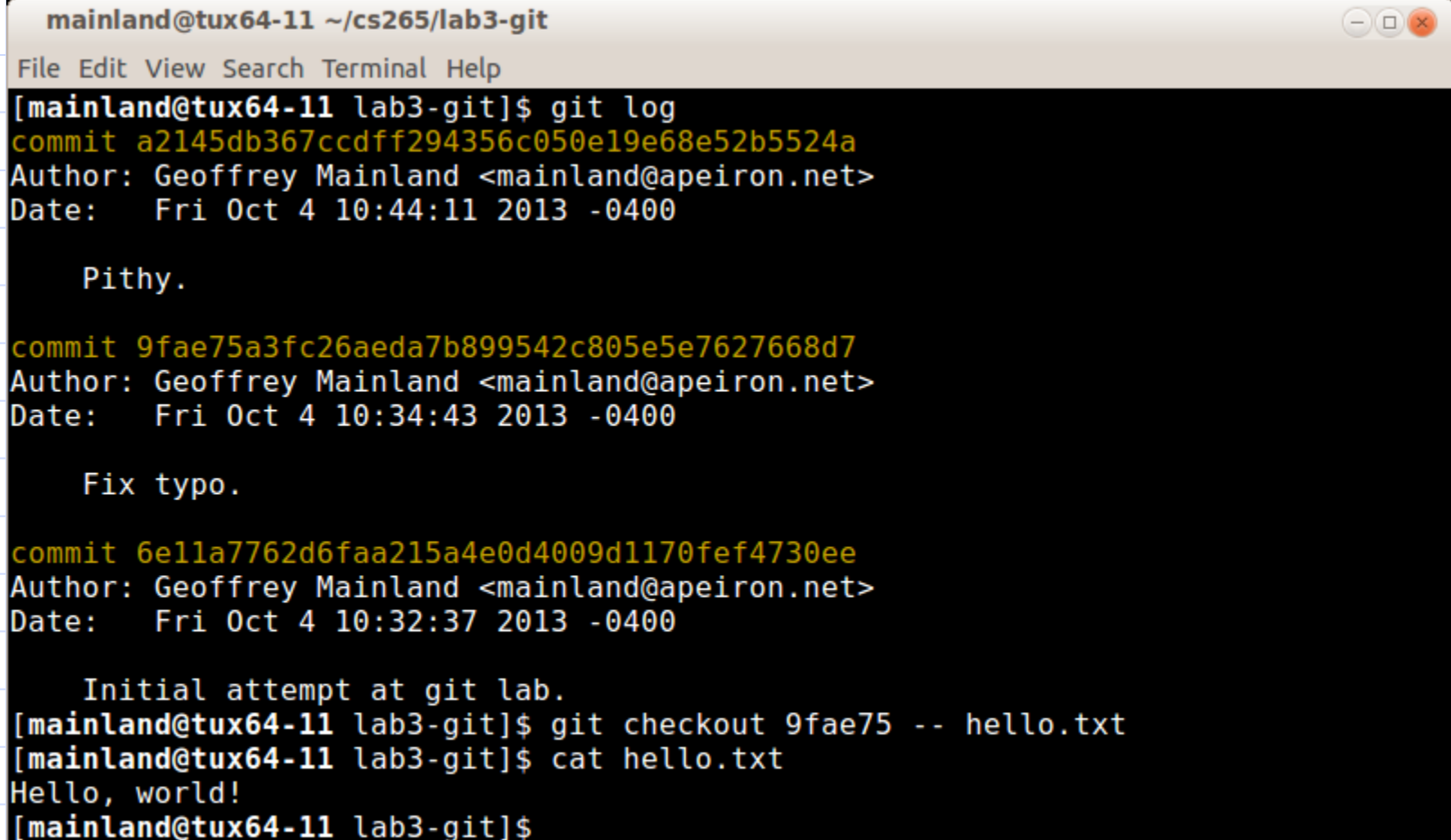
Add a remote repo **git remote add**

Push changes to remote **git push**

Pull changes from remote **git pull**

Clone a repository **git clone**

Checking out a specific version



A terminal window titled "mainland@tux64-11 ~/cs265/lab3-git" with standard window controls. The terminal shows the output of a `git log` command, listing three commits by Geoffrey Mainland. The first commit is the initial lab setup, the second is a typo fix, and the third is an initial attempt at a git lab. Finally, the user runs `git checkout 9fae75 -- hello.txt` and `cat hello.txt`, resulting in the output "Hello, world!".

```
mainland@tux64-11 ~/cs265/lab3-git
File Edit View Search Terminal Help

[mainland@tux64-11 lab3-git]$ git log
commit a2145db367ccdf294356c050e19e68e52b5524a
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:44:11 2013 -0400

    Pithy.

commit 9fae75a3fc26aeda7b899542c805e5e7627668d7
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:34:43 2013 -0400

    Fix typo.

commit 6e11a7762d6faa215a4e0d4009d1170fef4730ee
Author: Geoffrey Mainland <mainland@apeiron.net>
Date:   Fri Oct 4 10:32:37 2013 -0400

    Initial attempt at git lab.
[mainland@tux64-11 lab3-git]$ git checkout 9fae75 -- hello.txt
[mainland@tux64-11 lab3-git]$ cat hello.txt
Hello, world!
[mainland@tux64-11 lab3-git]$
```

Naming things in git



By branch	master
-----------	---------------

By hash	9fae75
---------	---------------

Previous commit	HEAD^
-----------------	--------------

<i>n</i> commits in the past	HEAD~<i>n</i>
-------------------------------------	----------------------