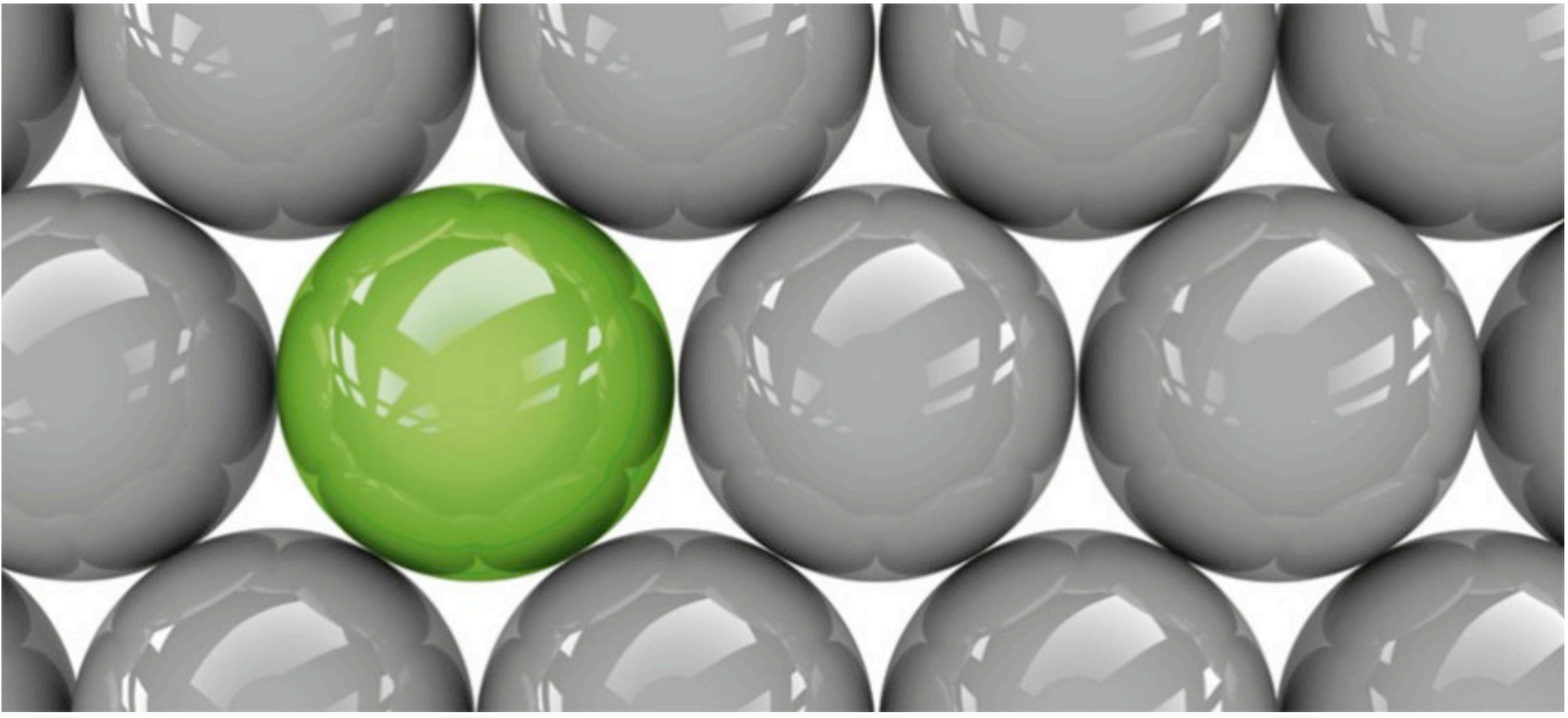
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF



Computational Physics WS15/16

Dr. Götz Lehmann, Inst. f. Theoretische Physik I

Agenda

1. Who is involved?
2. Outline of the lecture
3. Organizational things

Computational Physics

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Who is involved?

- Friedrich Schluck
- Stella Glöckner
- Eckhard Suckow

How to reach us

- Email: simu@tp1.uni-duesseldorf.de
- Office: 25.32.01.43
- ...if it concerns your homework, via GitHub

Find this document later at GitHub: http://github.com/TP1-HHU

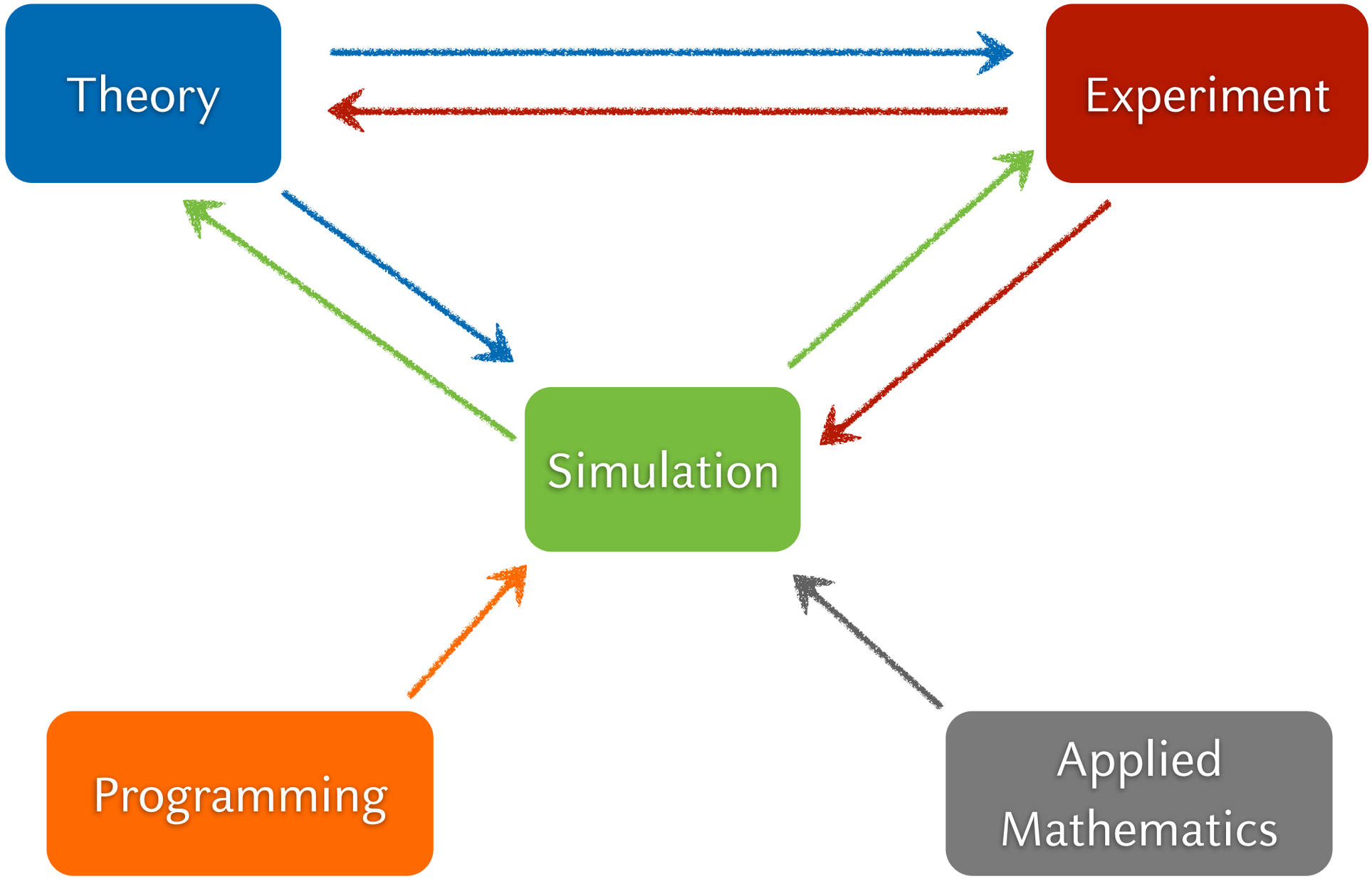
Organization of the class

- 2h lectures + 2h lab/tutorial per week + final exam ➠ 6 CP
- No 3CP rule
- Lecture: Wednesday, 8:30-10:30, HS 5J
- Lab classes: To be decided in a few minutes
- Final exam will (probably) be writing a computer code (problem of organization)

- Attendance at lab classes is mandatory

- Homework: To be handed in via GitHub, 2/3 of problems have to be solved

Computational Physics



Topics of the lecture

- Numerical methods
- Programming

Topics of the lecture

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Numerical methods

- Focus will be on initial value problems for
  - ordinary differential equations (ODEs)
  - partial differential equations (PDEs)
- This will require dealing with
  - finite difference approximations to derivatives
  - systems of linear equations
  - root finding of nonlinear equations
  - discrete Fourier transformation

Topics of the lecture

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Problems to be discussed

ODEs

$$\frac{d\vec{y}}{dt} = f(\vec{y}(t), t)$$

One-step methods: Runge-Kutta
Multi-Step methods: Adams Methods
Verlet Methods

$$\dot{p} = -\frac{\partial H(p,q)}{\partial q}$$
$$\dot{q} = \frac{\partial H(p,q)}{\partial p}$$

Symplectic methods for Hamiltonian systems

Topics of the lecture

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Problems to be discussed

PDEs

$\nabla^2 \phi = -\varrho$ Poisson eq.

$\frac{\partial^2}{\partial t^2} f - \nabla^2 f = 0$ Wave eq.

$\frac{\partial}{\partial t} f + \nabla \cdot j = 0$ Continuity eq.

$\frac{\partial}{\partial t} f - D \nabla^2 f = 0$ Diffusion eq.

$\frac{\partial}{\partial t} u - u \frac{\partial}{\partial x} u = 0$ Burgers eq.

$\mathrm{i} \frac{\partial}{\partial t} \psi + q \frac{\partial^2}{\partial x^2} \psi + |\psi|^2 \psi = 0$ NLSE

Topics of the lecture

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Programming

- C++, the language
  - as the better C
    - datatypes, functions, pointers, references
  - beyond C: Object oriented programming
    - classes, namespaces
    - external libraries: Avoid *Not invented here* syndrome
    - The basics of multi-core applications
    - You will need a C++ compiler for your homework!
- C++, the eco-system
  - Git as an example for a version control system
  - Makefiles / CMake
  - Valgrind

How to get your hands on C++

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

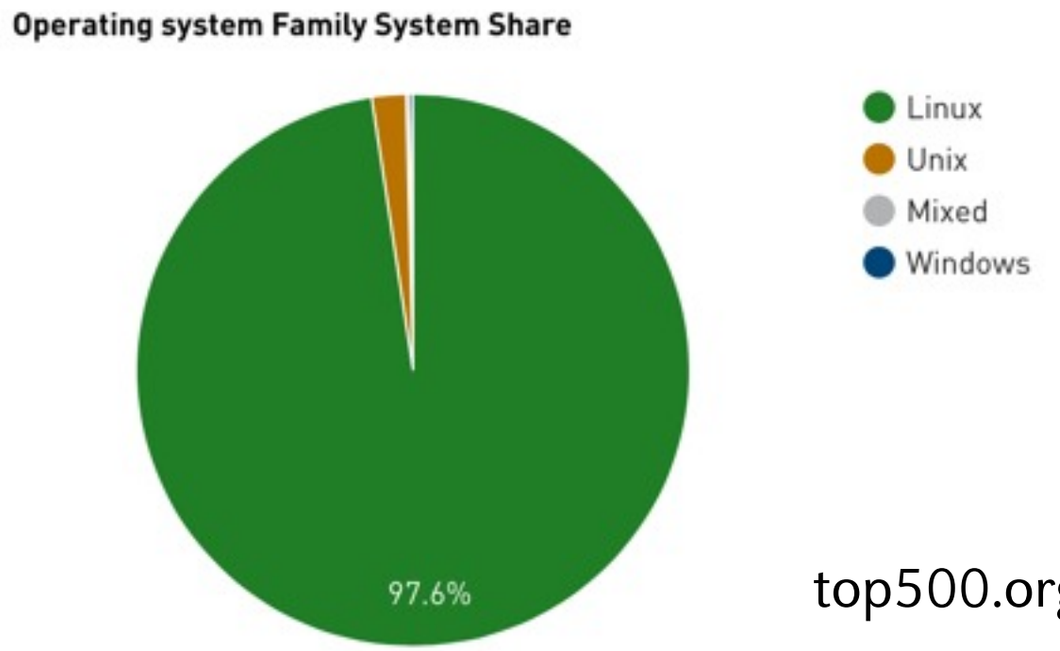
- Windows:
  - MS Visual Studio via www.dreamspark.com
    - ...you‘re on your own
- Linux:
  - Lab class will use Linux
    - proper installation next to Windows (if your advanced, you know what you do)
    - live USB (fairly easy, use Linux Live USB creator to create bootable USB stick)
      - http://www.linuxliveusb.com (choose a “persistant“ capable distribution)
    - virtual machine (medium complexity, best result for the price)
      - Install VirtualBox for Windows https://www.virtualbox.org
      - Download a Linux distribution (Fedora, OpenSUSE, Mint, ...)
      - Install Linux in the virtual machine
- Mac:
  - Install XCode app. Done. Easy as that.

Why Linux?

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Linux is the backbone of computational science

Virtually all big machines use Linux



top500.org

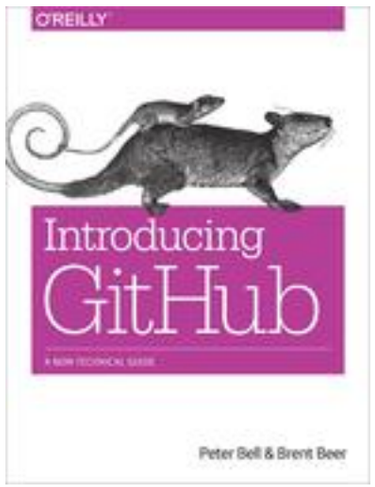
Everything beyond one computer runs on Linux...

SGI UV2000, 512 Cores, ZIM

Managing sources

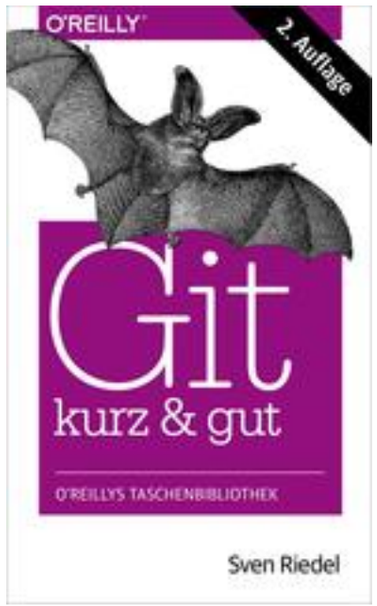
Git & GitHub

- Git is software that allows you to organize your source-code
- We will use GitHub to *hand out & collect* sources for lab classes and homework
- First homework & first lab class will deal with GitHub & Git



Introducing GitHub
Bell & Beer

Preißel, Stachmann



Riedel

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Single User Git scenario - Using Git as a log book

repository

A repository contains a set of files
May reside on a local disk or somewhere on the internet

local work directory

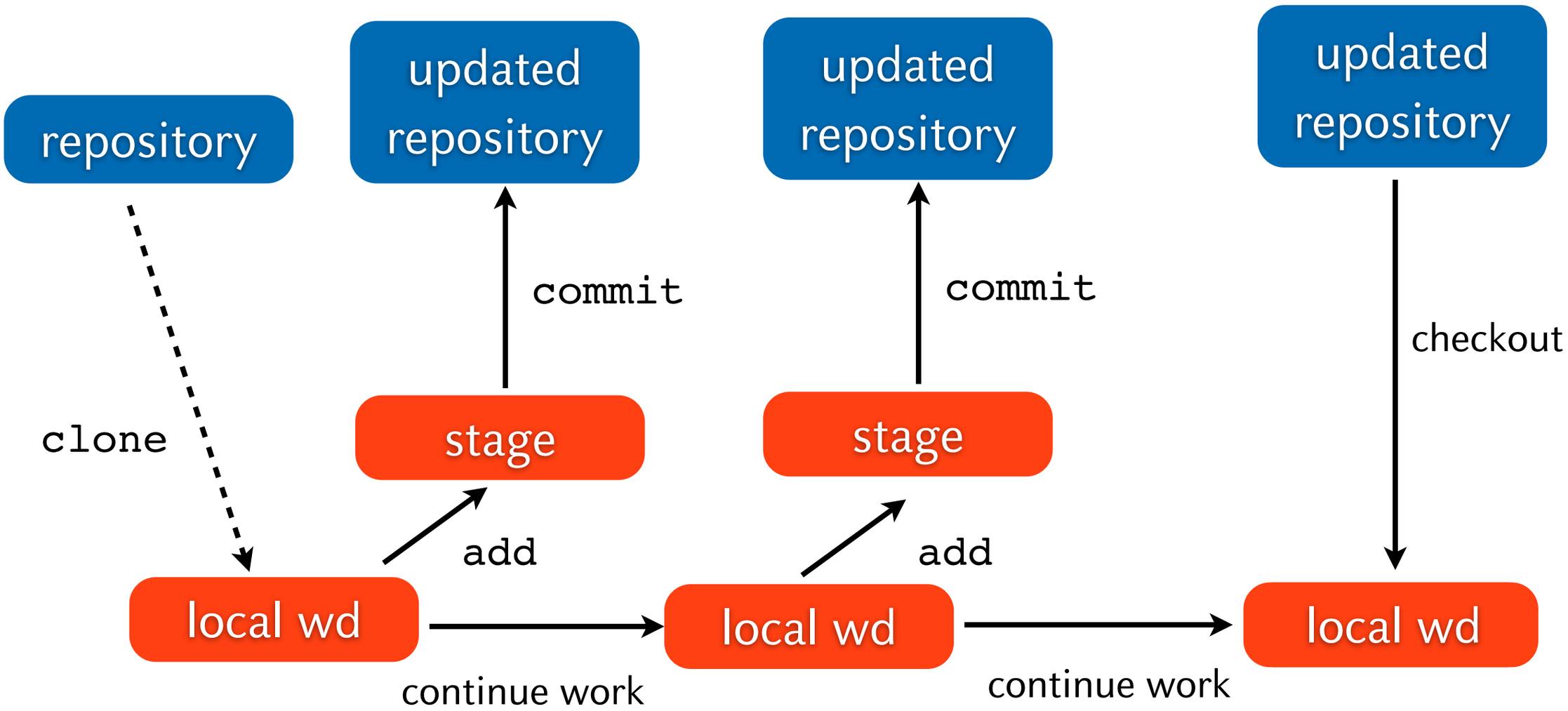
A directory on your hard disk

By `cloning` the repository into your local working directory, you create a local copy of the files contained in the repository.

After editing the files, you `add` the changes to the staging area. This is an intermediate step before sending them actually to the repository by `comitting` them.

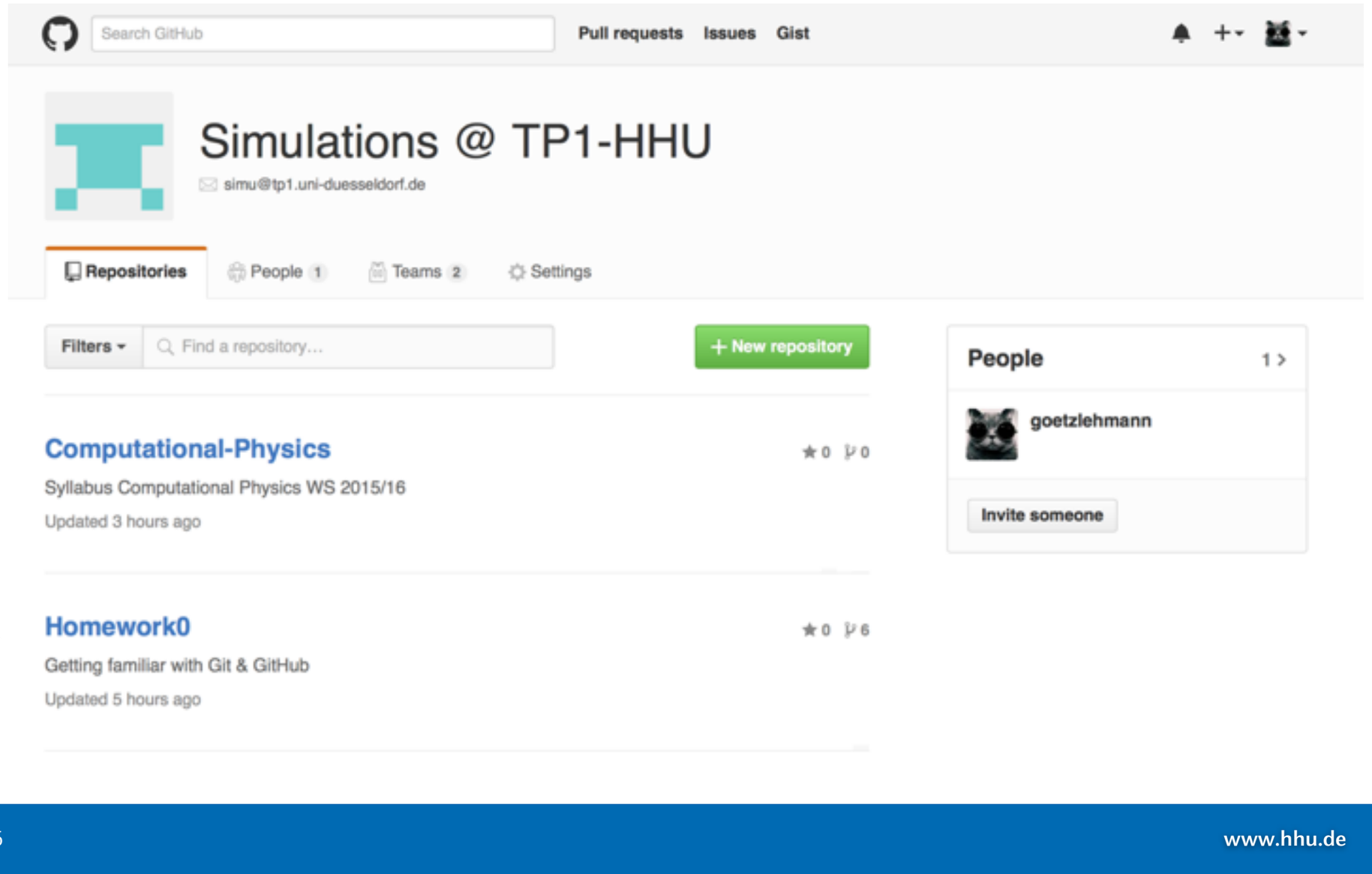
The repository keeps track of all older versions of all files. You can easily compare changes between between different `commits`.

Single User Git scenario - Using Git as a log book



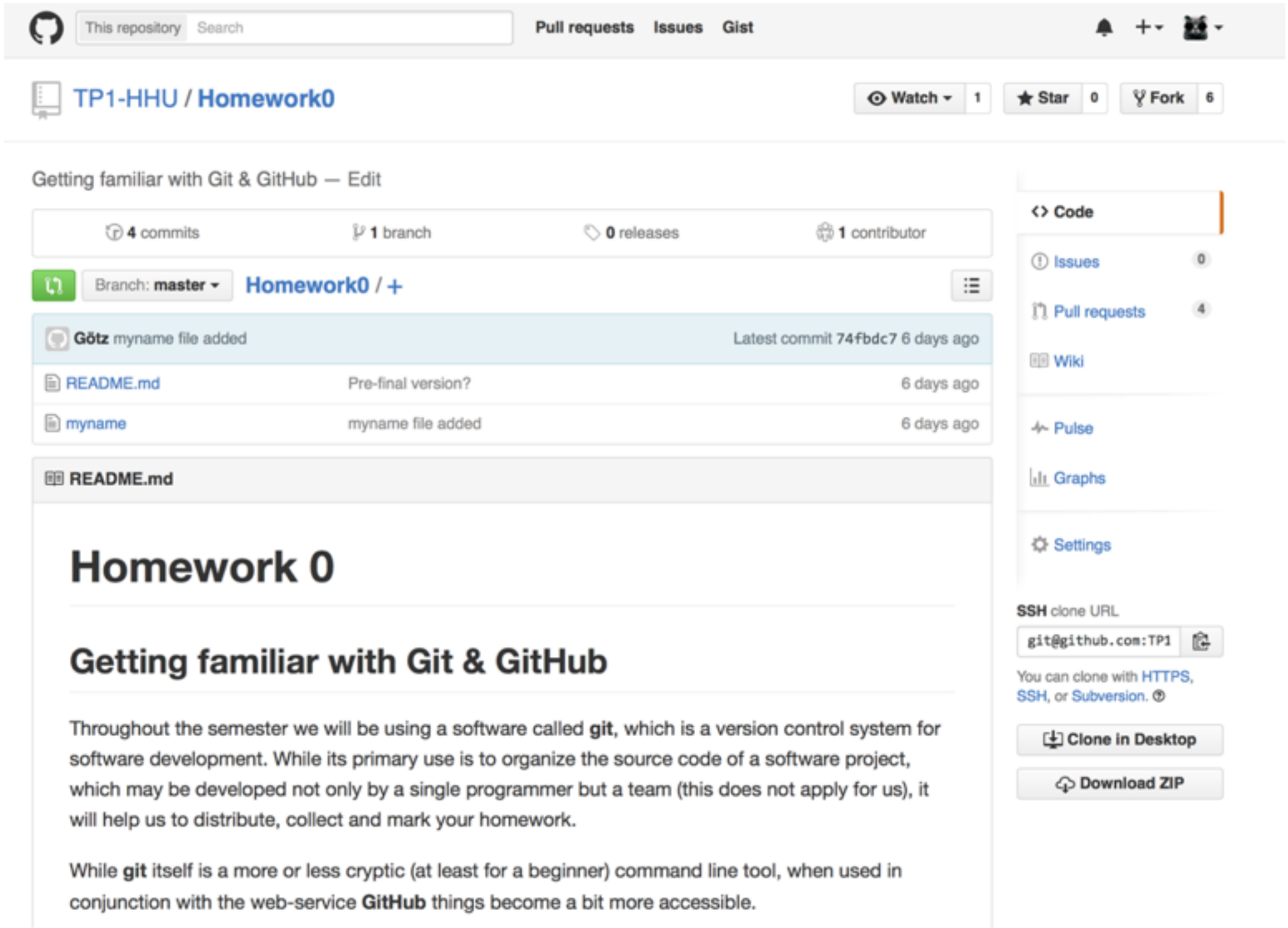
GitHub

- GitHub is a platform which allows to host repositories online
- Accounts are free, you will have to create one
- Our account is named TP1-HHU, thus our address is
  http://github.com/TP1-HHU



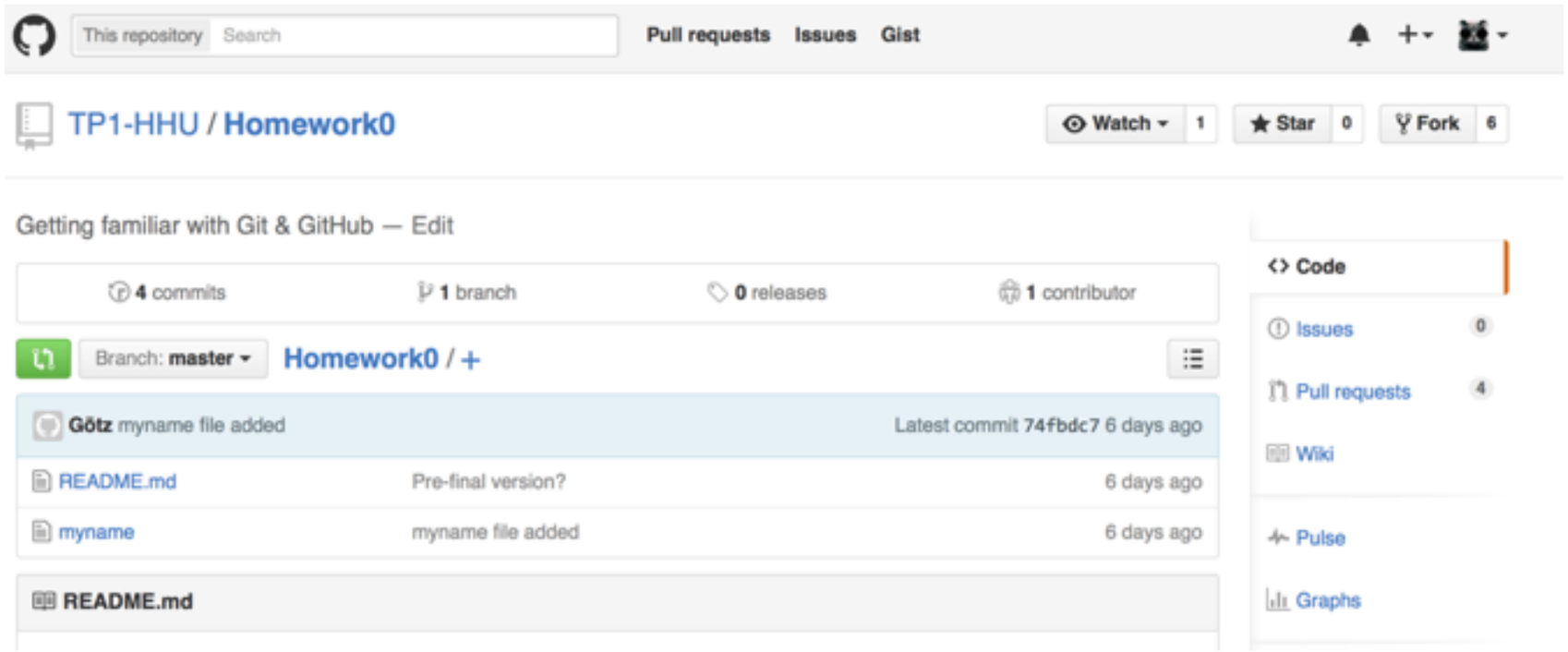
Repositories

GitHub


This repository Search
Pull requests
Issues
Gist
TP1-HHU / Homework0
Watch 1
Star 0
Fork 6
Getting familiar with Git & GitHub — Edit
4 commits
1 branch
0 releases
1 contributor
Branch: master
Homework0 / +
Götz myname file added
Latest commit 74fbdc7 6 days ago
README.md
Pre-final version?
6 days ago
myname
myname file added
6 days ago
Code
Issues 0
Pull requests 4
Wiki
Pulse
Graphs
Settings
SSH clone URL
git@github.com:TP1
You can clone with HTTPS, SSH, or Subversion.
Clone in Desktop
Download ZIP
README.md
Homework 0
Getting familiar with Git & GitHub
Throughout the semester we will be using a software called git, which is a version control system for software development. While its primary use is to organize the source code of a software project, which may be developed not only by a single programmer but a team (this does not apply for us), it will help us to distribute, collect and mark your homework.
While git itself is a more or less cryptic (at least for a beginner) command line tool, when used in conjunction with the web-service GitHub things become a bit more accessible.

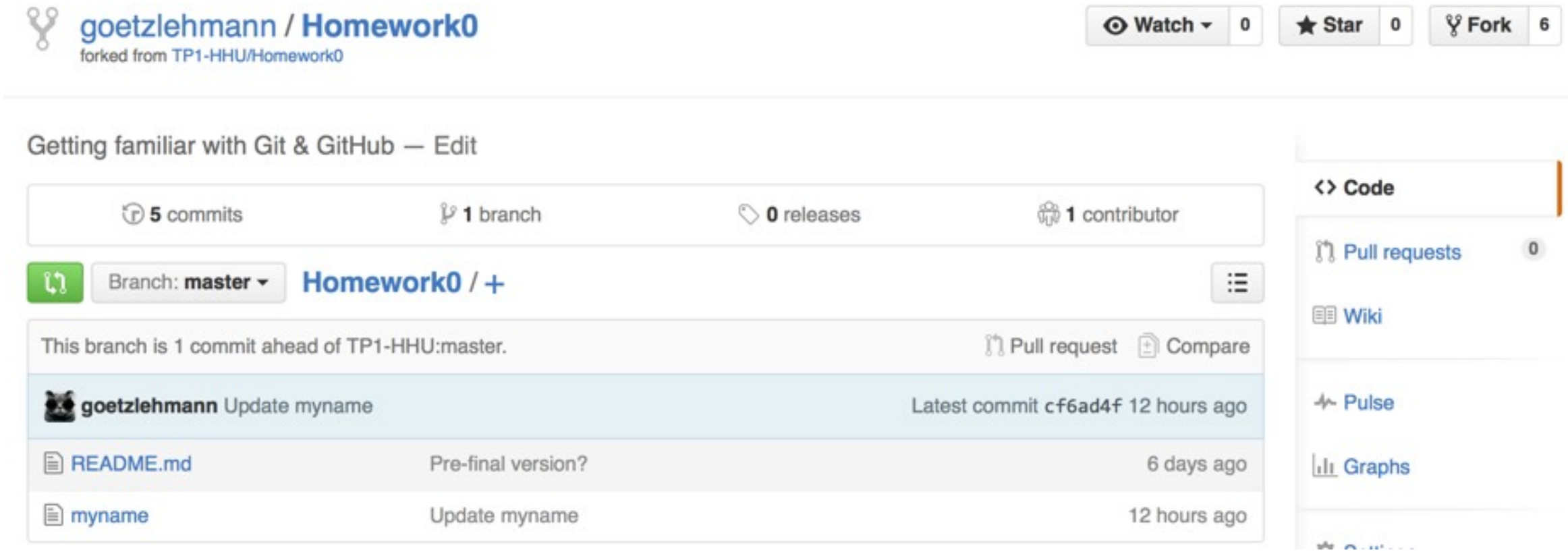
GitHub Homework

- Create a GitHub account for yourself
  - choose any nickname you want
  - the goal of this homework is to associate you to your nick
- Navigate to the Homework0 repository on our account and fork the repository by pressing the **fork** button on the top right



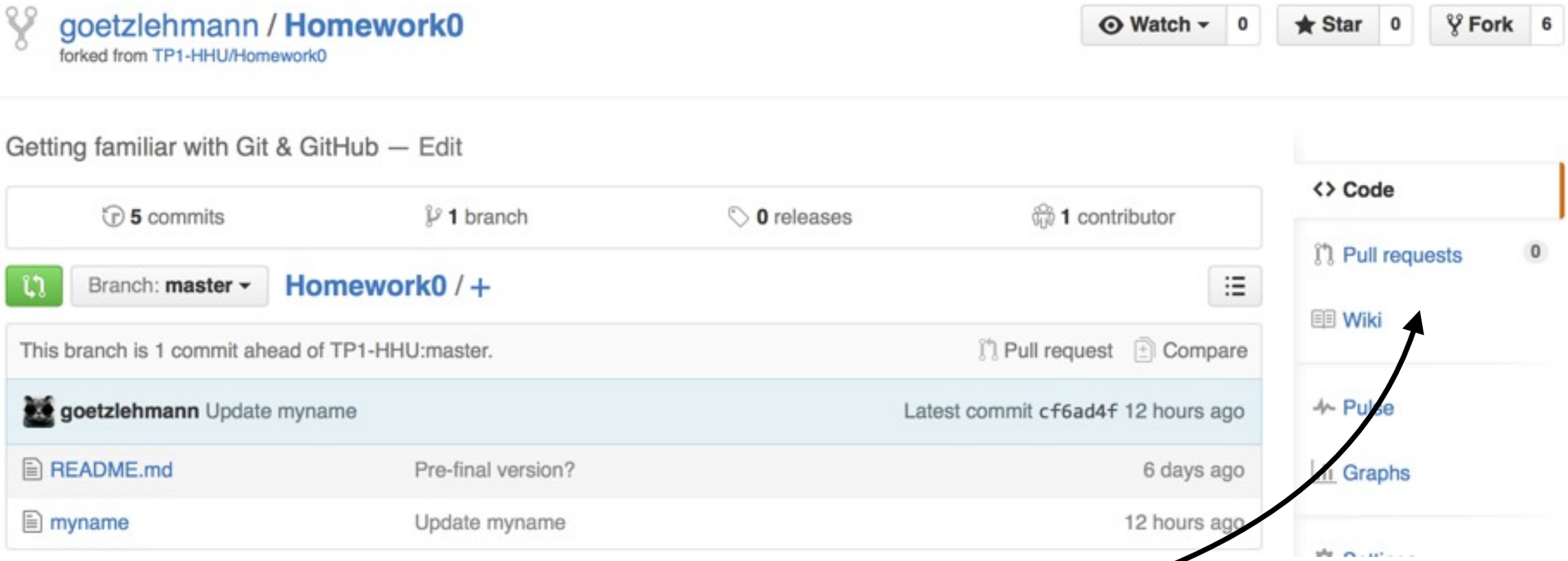
GitHub Homework

- Now a fork of the repository has been created in your own account
- The files in your account belong to you - you can change them



GitHub Homework

- Edit the file myname and write your name into it
- Upon saving the changed file,
  leave Commit directly to the `master` branch unchanged



- Create a pull request

GitHub Homework

Creating the pull request:

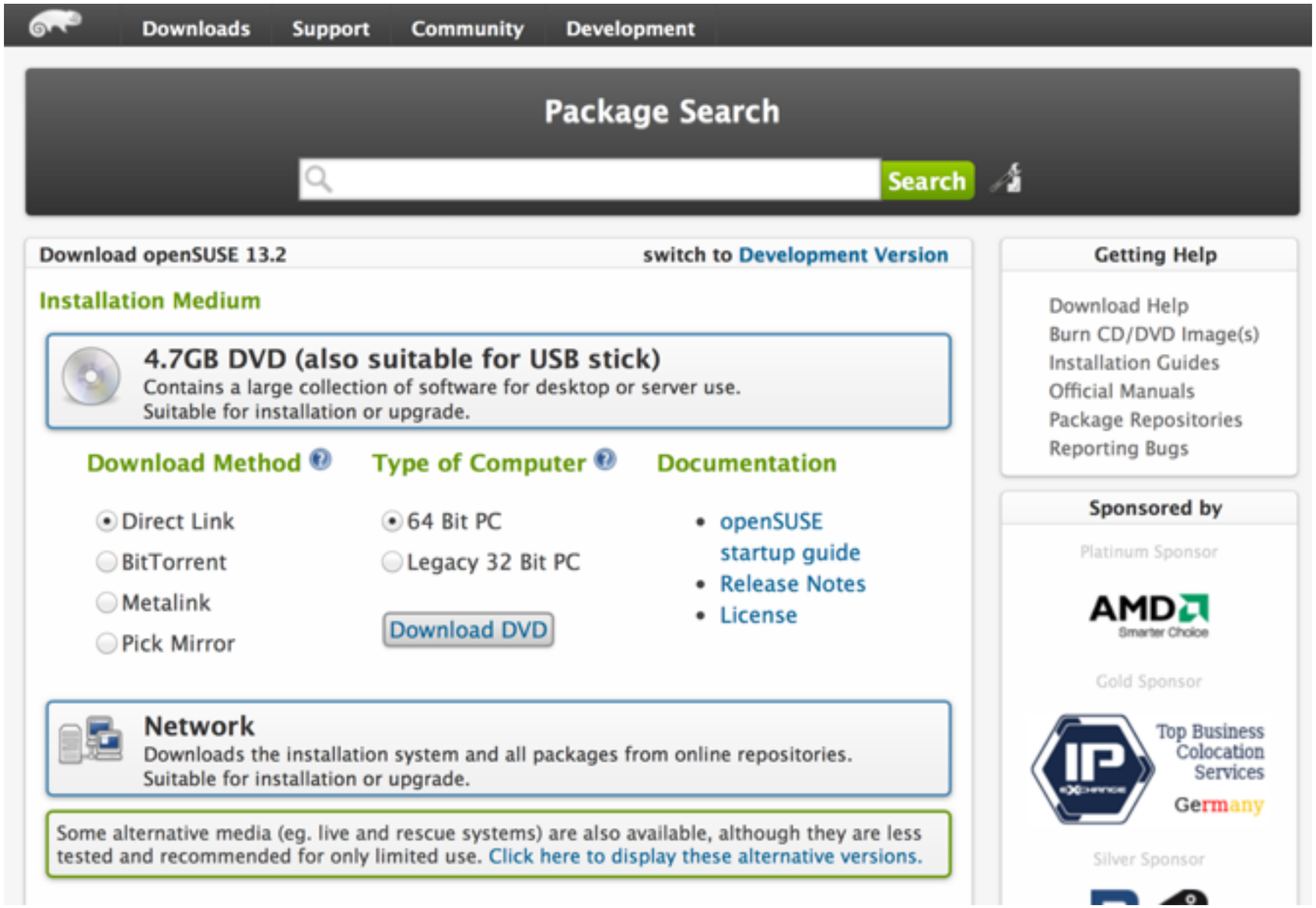
- Green button „new pull request“
- Green button „view pull request“
- Leave your real name in the comment field for the Pull request
- done

Installing Linux in VirtualBox

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Installation example for OpenSUSE

- www.opensuse.org

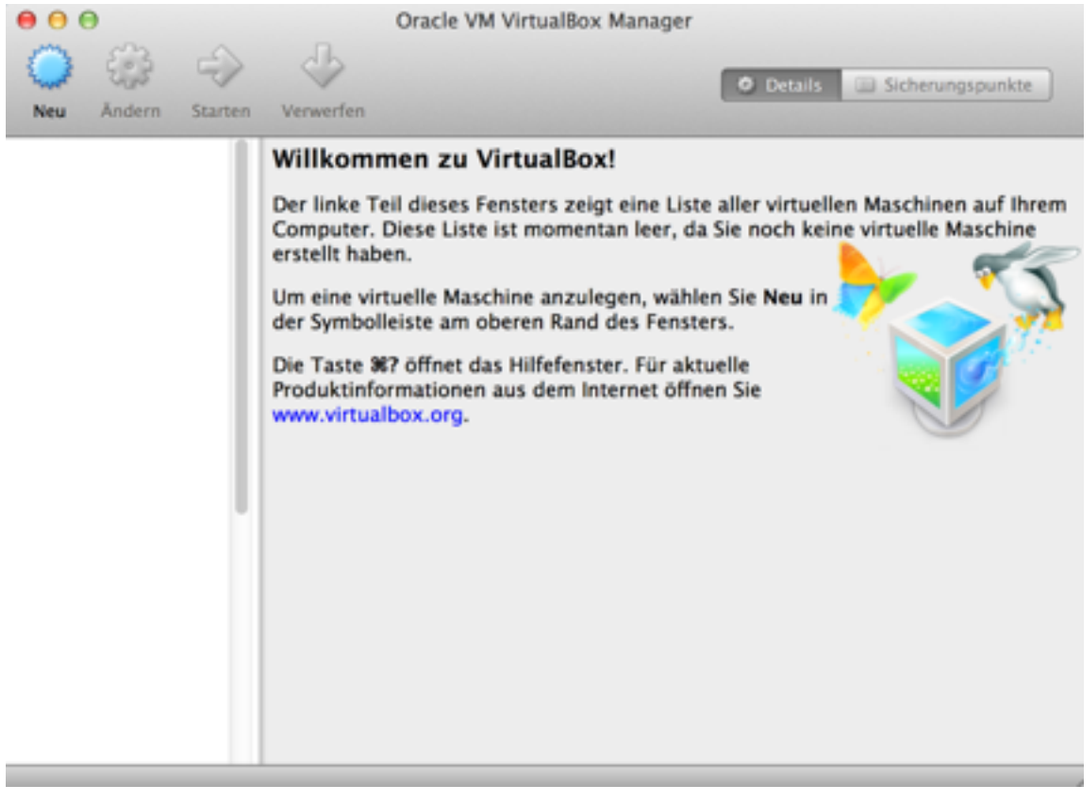


Installing Linux in VirtualBox

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Short summary

- Download & Install VirtualBox (www.virtualbox.org)
- Start VirtualBox



- Click „New“, choose Linux and then the distribution you want to install.
- In this example we will use OpenSuse

Installing Linux in VirtualBox

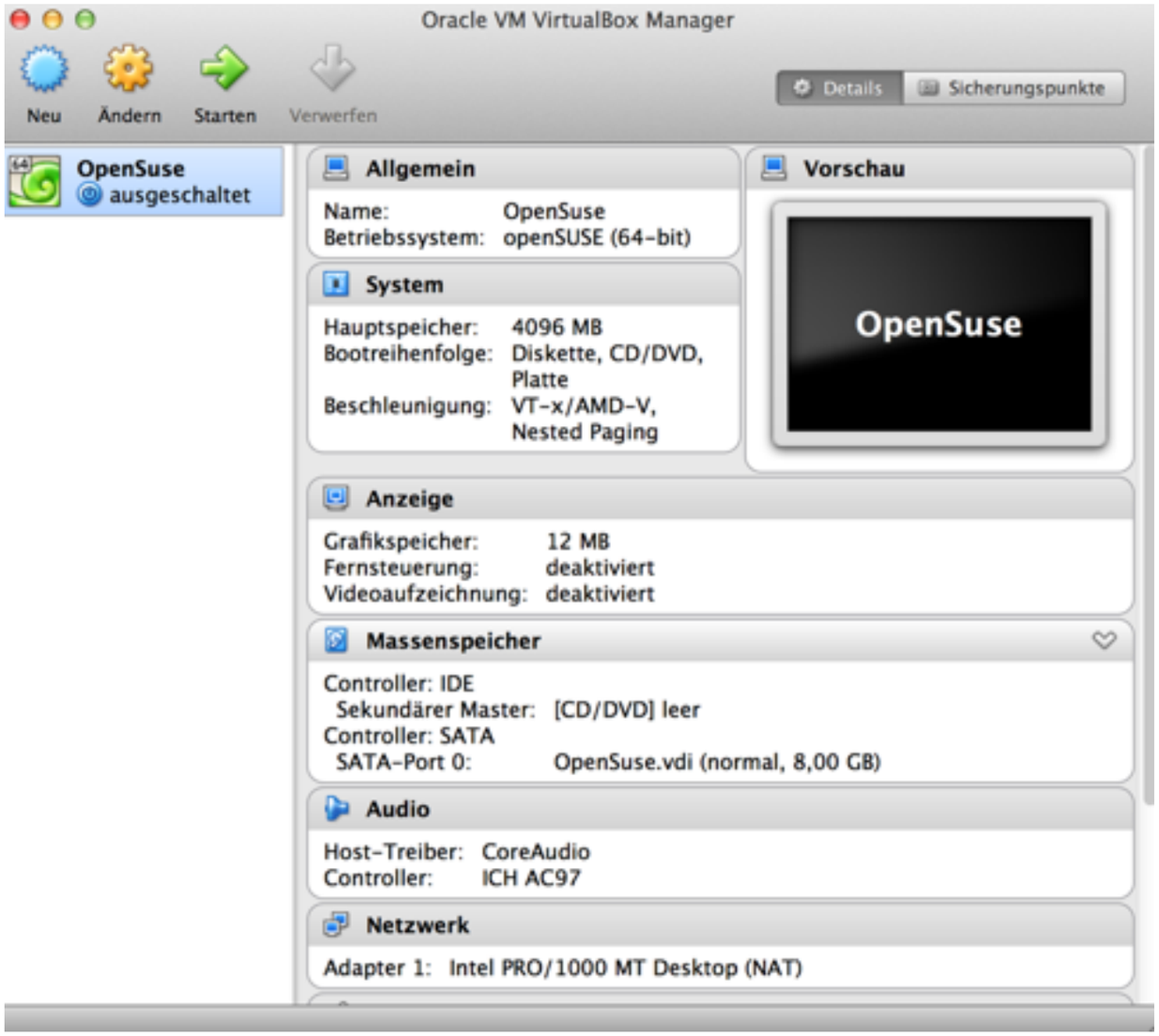
- Specify how much memory should be available for the virtual machine. Go for 4 GB if you can, but maybe not for more than half of what your computer has in total.

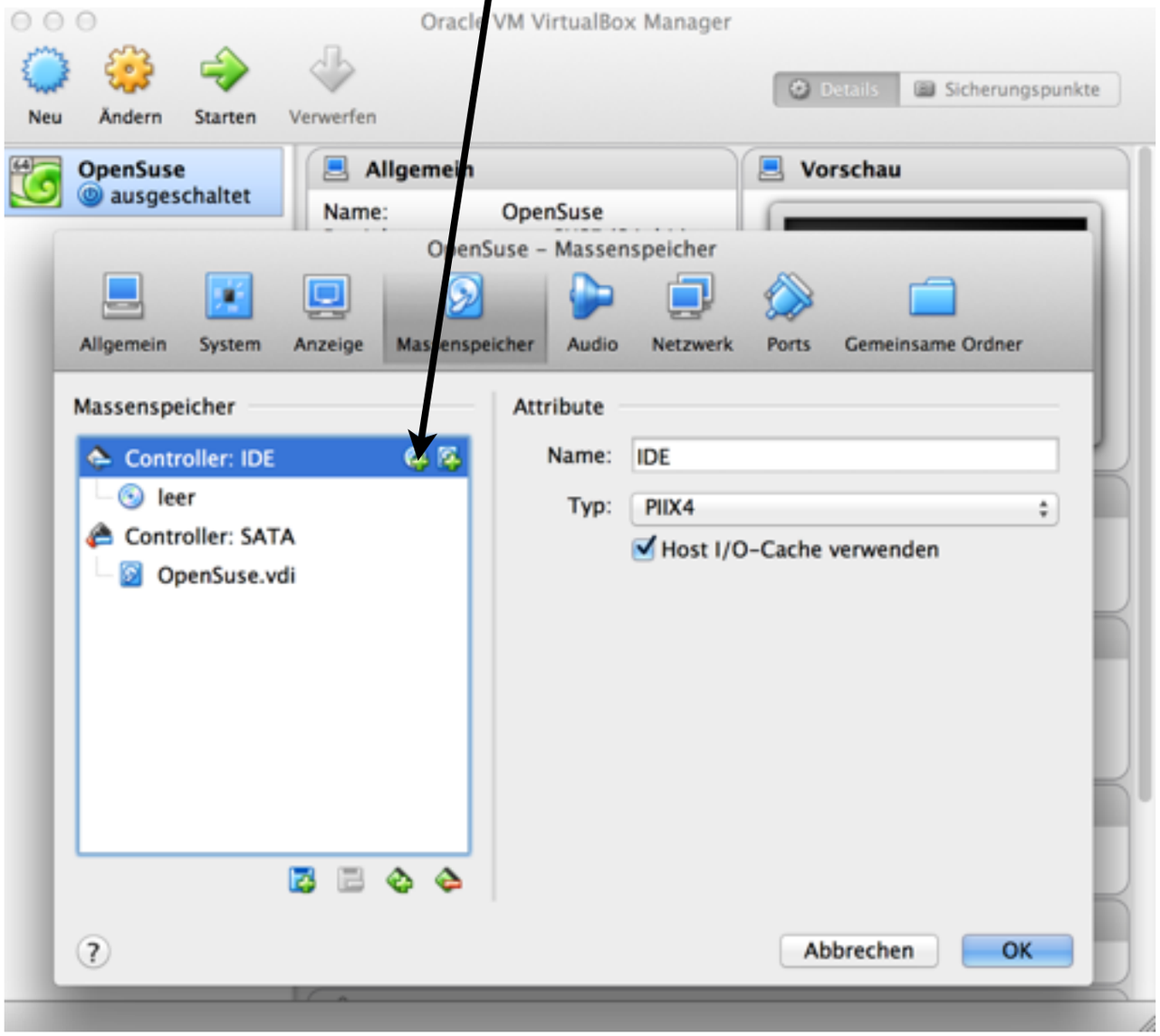
- Leave the recommended disk size as is. 8GB should be fine, significantly less could become problematic.
- Choose „VDI“ & „fixed size“ in the two next dialoges

Installing Linux in VirtualBox

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Click here and choose previously downloaded iso disk image of OpenSuse





....start the virtual machine to initiate the installation of OpenSUSE into the VirtualBox

Complete Git Workflow

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Necessary steps to work on lab classes & homework

- Fork the respective repository from the TP1-HHU account on GitHub into your private account

- Clone the GitHub (remote) repository from your private account into your work local directory (i.e. create a local repository)

- Work on the local files

- add & commit the files to your local repository

- push the changes in the local repository to the remote repository

- create a pull request on GitHub to tell us that we can take a look at your changes

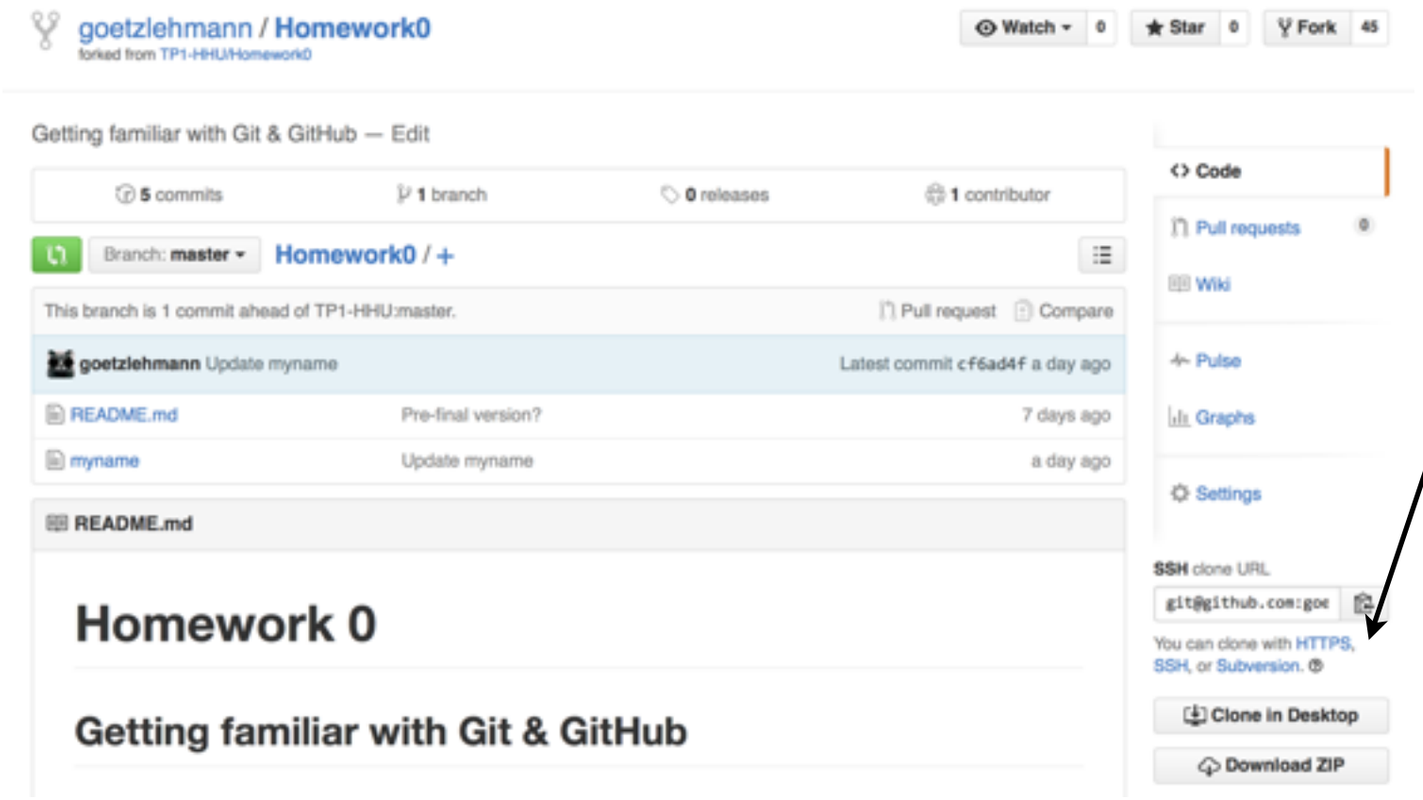
Complete Git Workflow

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Cloning your remote repository

Terminal commands:

```
git clone /path/to/repository   ← for a local repository
git clone username@host:/path/to/repository ← for a remote repository
```



1. Press HTTPS here
2. Copy HTTPS URL
3. Clone:

```
git clone https://github.com/
  goetzlehmann/Homework0.git
```

Complete Git Workflow

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Add & Commit

- Your local repository consists out of three things:
  - The working copy of your files
  - The index and the head (which are invisible to you if you don‘t look closely)
- `git add` and `git commit` are the most frequent commands that you will use

- `git add <filename>`
  Stages the changes made to the file for a subsequent commit to the repository. The idea is to first gather all the changes to different files which should be submitted to the repository upon the next commit.

- `git commit`
  This actually sends all changes which have been staged to the local repository. Upon execution of this command a text editor will open up, where you should enter a commit comment. This editor might be VI, tricky because you need to know the keyboard shortcuts to find your way out! Better: `git commit -m „Changed variable names for parameters“`

Complete Git Workflow

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Pushing to remote repository

- After commiting the files we can push the changes in the local repository to the remote repository residing on GitHub by doing a push

- `git push origin master`
  Submits the changes from local to remote repository. This will by done via the HTTPS protocol since we used this protocol when we cloned the repository. This will ask for username and password on GitHub to verify that you are allowed to submit to the repository.

- `git remote show origin  / git remote -v`
  Displays from where the repository was cloned and to which remote repository changes will be pushed.

Complete Git Workflow

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Complete Example

```
DeepThought2s-MacBook-Pro: ~/work/test $git clone https://github.com/goetzlehmann/Homework0.git
Cloning into 'Homework0'...
remote: Counting objects: 15, done.
remote: Total 15 (delta 0), reused 0 (delta 0), pack-reused 15
Unpacking objects: 100% (15/15), done.
Checking connectivity... done.
```

Clone

```
DeepThought2s-MacBook-Pro: ~/work/test $cd Homework0/
DeepThought2s-MacBook-Pro: ~/work/test/Homework0 $ll
total 16
-rw-r--r--  1 goetz  staff   2,3K 20 Okt 19:37 README.md
-rw-r--r--  1 goetz  staff    12B 20 Okt 19:37 myname
DeepThought2s-MacBook-Pro: ~/work/test/Homework0 $vi myname
```

Edit

```
DeepThought2s-MacBook-Pro: ~/work/test/Homework0 $git add myname
DeepThought2s-MacBook-Pro: ~/work/test/Homework0 $git commit -m "inserted my name"
[master 743d7cb] inserted my name
 Committer: Götz <goetz@DeepThought2s-MacBook-Pro.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
```

Add & Commit

```
DeepThought2s-MacBook-Pro: ~/work/test/Homework0 $git push origin master
Username for 'https://github.com': goetzlehmann
Password for 'https://goetzlehmann@github.com':
Counting objects: 5, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 307 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/goetzlehmann/Homework0.git
   cf6ad4f..743d7cb  master -> master
```

Push

Now everything is live and updated in your private GitHub repository. Send a pull request next.

Complete Git Workflow

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Branches

- Each repository may have several branches. The default branch is called `master`.

- If bigger changes are made to a project, first a new branch is created, e.g. `new_feature`. The new branch contains all files that where present at the time that branching was initiated.

- All changes which are related to the new feature are then committed to the new branch `new_feature`.

- Once the development in the branch is done, the changes are `merged` into the master branch and the new branch is deleted.

- For us the `master` branch is usually sufficient, you do not need to create branches. If something behaves weired in Git/GitHub, checkout if you did not accidentally create a new branch.