

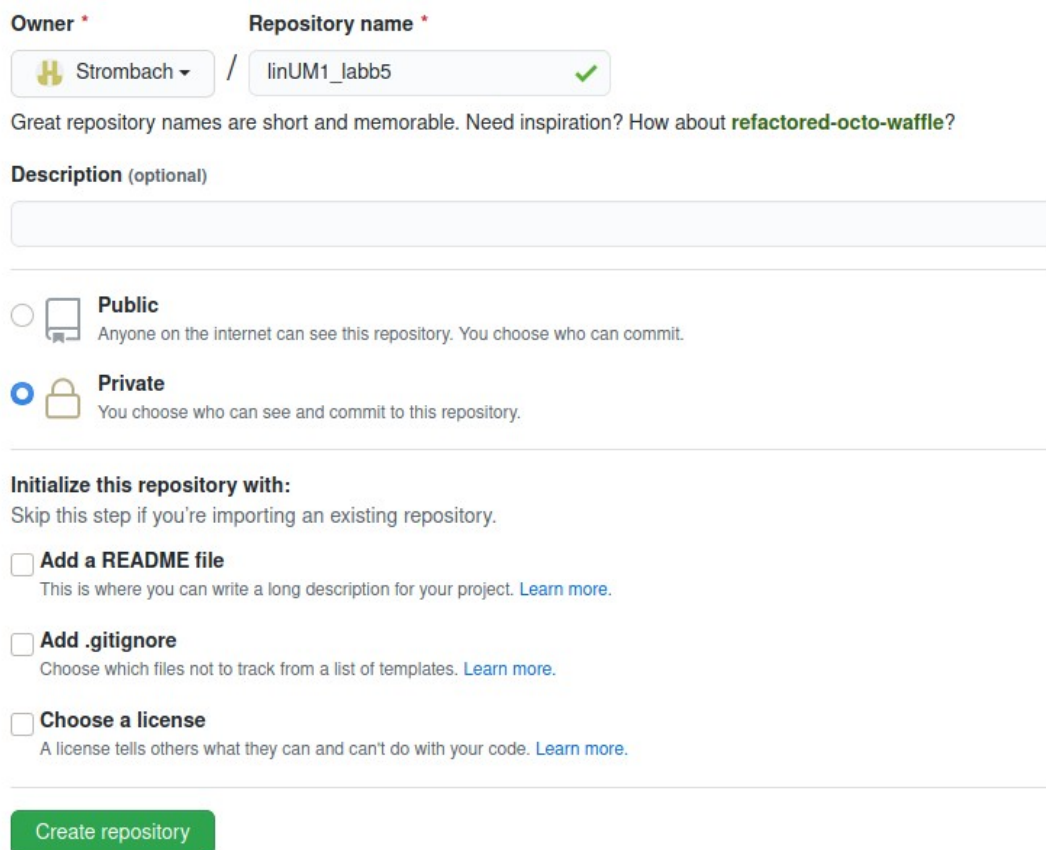
Uppgift 5, Git

Termin: HT20

Kurs: Linux som utvecklingsmiljö

Student: Fredrik Strömbäck, 19920823-1556

1. Hade redan ett konto på github så jag använder mig utav detta.
Fyller då i namnet och väljer att skapa privat repo.



The screenshot shows the GitHub 'Create repository' interface. At the top, there are two input fields: 'Owner' with a dropdown menu showing 'Strombach' and a 'Repository name' field containing 'linUM1_lab5' with a green checkmark. Below these is a text prompt: 'Great repository names are short and memorable. Need inspiration? How about [refactored-octo-waffle?](#)'. A 'Description (optional)' text area is below that. The 'Visibility' section has two radio buttons: 'Public' (unselected) and 'Private' (selected). The 'Initialize this repository with:' section has three checkboxes: 'Add a README file' (unselected), 'Add .gitignore' (unselected), and 'Choose a license' (unselected). At the bottom is a green 'Create repository' button.

Owner ^{*} Repository name ^{*}

Strombach / linUM1_lab5 ✓

Great repository names are short and memorable. Need inspiration? How about [refactored-octo-waffle?](#)

Description (optional)

☐ Public
Anyone on the internet can see this repository. You choose who can commit.

☒ Private
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file
This is where you can write a long description for your project. [Learn more.](#)

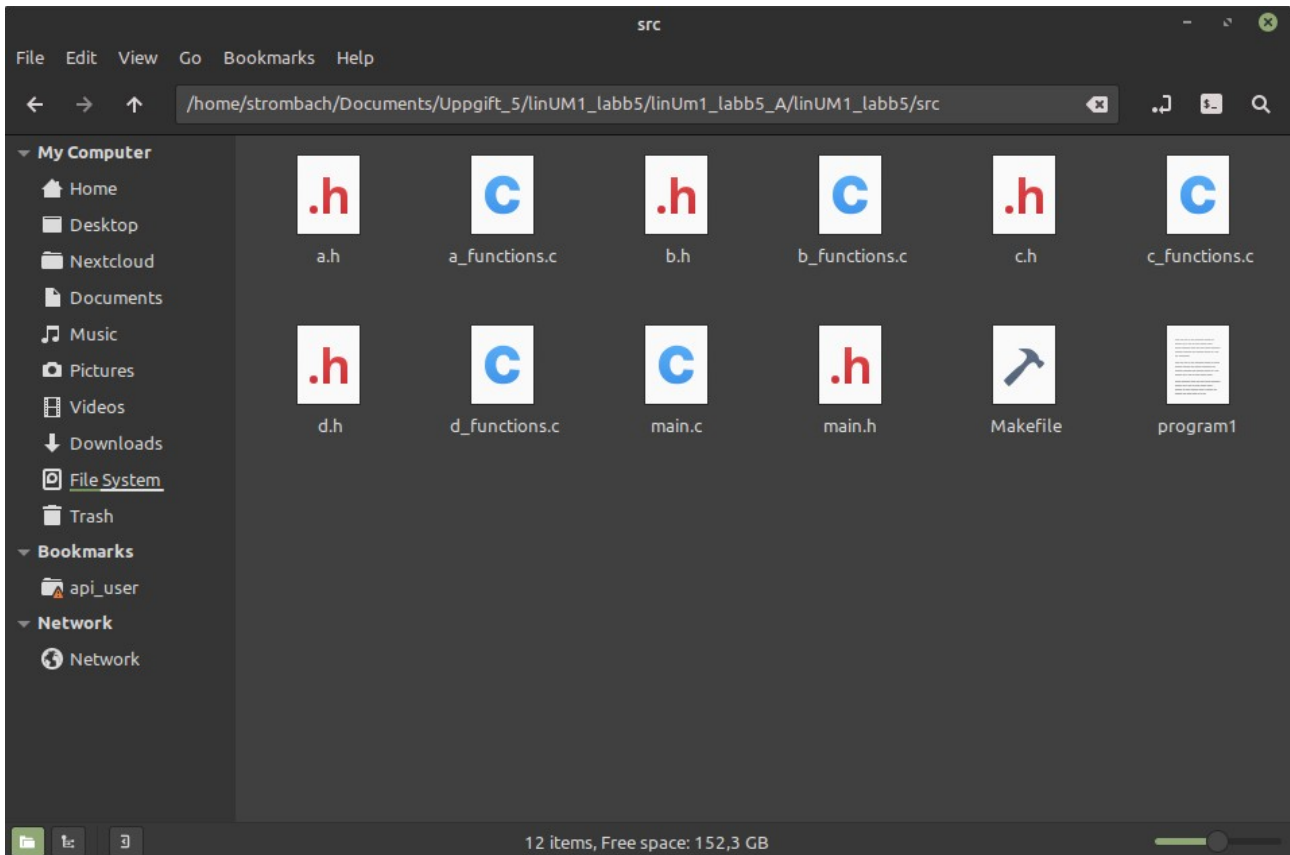
☐ Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

2. Skapar en katalog som heter linUM1_lab5 på plats
/home/strombach/Documents/Uppgift_5
3. Skapar undermappen linUM1_lab5_A med "mkdir".
4. Följer och kopierar de git-kommandon som finns i kursmodulen, för att koppla
samman repot med min lokala katalog.
5. Skapar src mappen i min lokala kopia.

6. Kopierar in alla filer från 3D i src.



7. Använder mig av "git add .", "git commit -m 'adds src-files'" och "git push" för att addera, commita och pusha upp de nya filerna till github.

8. Skapar den parallella katalogen "linUM1_lab5_B".

9. Klonar ner repot till "linUM1_lab5_B". Jag använder mig utav SSH istället för HTTPS.

```
strombach@ThinkPad-T480: ~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5...
File Edit View Search Terminal Help

strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B$ git clone git@github.com:Strombach/linUM1_lab5.git
Cloning into 'linUM1_lab5'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 18 (delta 0), reused 18 (delta 0), pack-reused 0
Receiving objects: 100% (18/18), 4.53 KiB | 2.27 MiB/s, done.
strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B$ ls
linUM1_lab5
strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B$ cd linUM1_lab5/
strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B/linUM1_lab5$ ls
README.md  src
strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B/linUM1_lab5$
```

10. Ändringarna gör jag på rad nio. Där "print_B" byts ut mot "print_A" och "print_D"

```
C main.c > main(int, char *[])
1  #include <stdio.h>
2
3  #include "main.h"
4
5  int main(int argc, char *argv[])
6  {
7      printf("The first four letters of the word are: ");
8      print_A();
9      print_A();
10     print_C();
11     print_D();
12     return (RETVALUE);
13 }
14
```

Jag pushar upp linUM1_lab5_A till github först.

När jag försöker pusha upp linUM1_lab5_B så får jag följande felmeddelande:

```
strombach@ThinkPad-T480: ~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5...
File Edit View Search Terminal Help
_lab5/src$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   main.c

strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B/linUM1_lab5/src$ git commit -m 'changed print_B to print_D'
[master 2d9048a] changed print_B to print_D
1 file changed, 1 insertion(+), 1 deletion(-)
strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B/linUM1_lab5/src$ git push
To github.com:Strombach/linUM1_lab5.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'github.com:Strombach/linUM1_lab5.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B/linUM1_lab5/src$
```

Jag använder mig av "git pull --rebase" för försöka kopiera ner senaste commiten från github och lägga min lokala commit i linUM1_lab5_B ovanpå. Detta lyckas inte och jag får ett felmeddelande om att merge konflikten måste lösas manuellt.

```

strombach@ThinkPad-T480:~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5_B/linUM1_lab5/src$ git pull --rebase
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 2), reused 4 (delta 2), pack-reused 0
Unpacking objects: 100% (4/4), 359 bytes | 359.00 KiB/s, done.
From github.com:Strombach/linUM1_lab5
   48b632e..06cbcl1  master    -> origin/master
Auto-merging src/main.c
CONFLICT (content): Merge conflict in src/main.c
error: could not apply 2d9048a... changed print_B to print_D
Resolve all conflicts manually, mark them as resolved with
"git add/rm <conflicted_files>", then run "git rebase --continue".
You can instead skip this commit: run "git rebase --skip".
To abort and get back to the state before "git rebase", run "git rebase --abort"
.
Could not apply 2d9048a... changed print_B to print_D

```

Jag öppnar upp main.c i nano där jag ser den "inkommande" commiten, alltså den som kopierades ner från github. Denna inkommande commiten är den som finns under "<<<<<< HEAD" och slutar vid "====="". Därefter ser jag den lokala commiten som sträcker sig fram till ">>>>>> 2d9048a... changed print_B to print_D".

```

strombach@ThinkPad-T480: ~/Documents/Uppgift_5/linUM1_lab5/linUM1_lab5...
File Edit View Search Terminal Help
GNU nano 4.8 main.c
#include <stdio.h>

#include "main.h"

int main(int argc, char *argv[])
{
    printf("The first four letters of the alphabet:\n");
    print_A();
<<<<<< HEAD
    print_A();
=====
    print_D();
>>>>>> 2d9048a... changed print_B to print_D
    print_C();
    print_D();
    return (RETVALUE);
}

[ Read 17 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line

```

För att lösa konflikten så tar jag bort hela den inkommande commiten, samt "<<<<<< HEAD", "=====" och ">>>>>> 2d9048a... changed print_B to print_D". Sedan kör jag en "git add ." för att markera konflikten som löst sedan kör jag "git rebase -continue" för att commita lösningen. Sedan "git push".