Git Intro

Ml4econ | 26 05 24

What is git, Why using it

- A distributed version control system (VCS)
  - helps track changes in source code during software development
  - allows multiple developers to work on a project simultaneously

- Was created by Linus Torvalds on 2005,

  to develop Linux OS

GIT -Essentials

git

- VEERESH H R

About Git

Created by **Linus Torvalds**,creator of Linux, in 2005

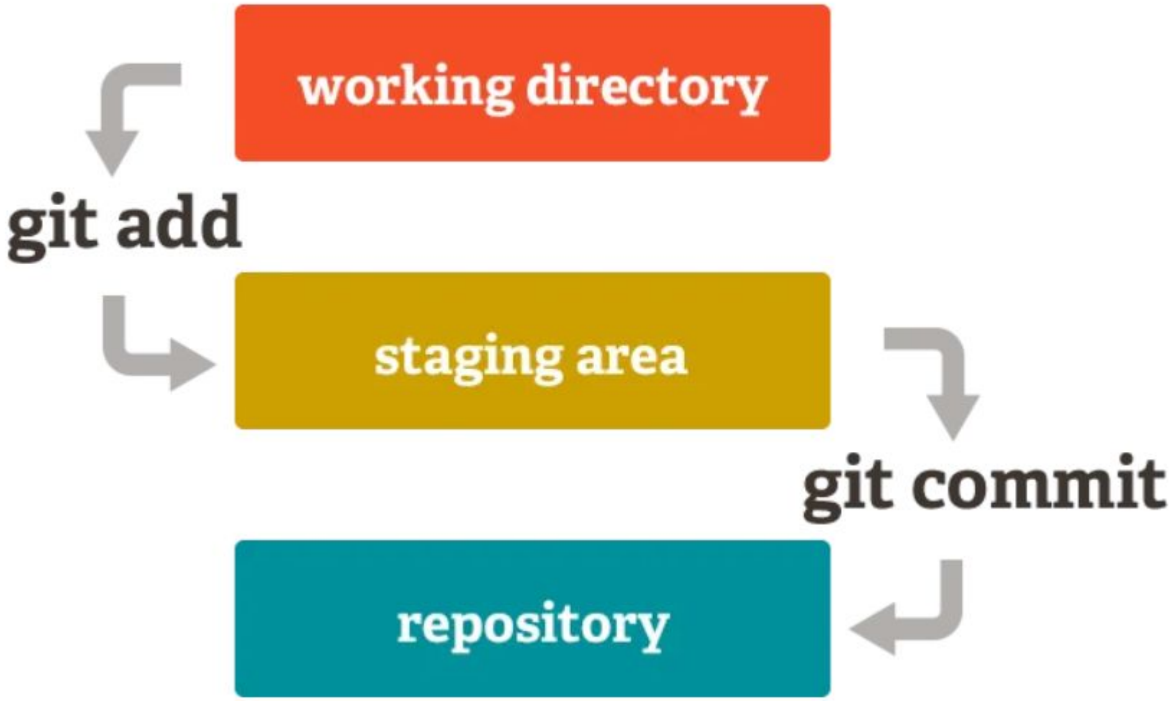
Linus Benedict Torvalds is a Finnish-American software engineer who is the creator and, historically, the principal developer of the Linux kernel, which is the kernel for Linux operating systems and other operating systems such as Android and Chrome OS.

- Came out of Linux development community
- Designed to do version control on Linux kernel.

Git is a free and open source distributed or decentralized version control system.

Reasons Everyone Use GIT

Git Spaces


working directory
git add
staging area
git commit
repository

Getting Started with Git

1. **Install Git:**

   Download from git-scm.com

2. **Configure Git:**

   `git config --global user.name <Your Name>`

   `git config --global user.email <your.email@example.com>`

3. **Clone a Repository (Repo):**

   `git clone <url>`

Branching in Git

1. **Create a new branch:**

   `git branch <branch-name>`

2. **Switch to a branch:**

   `git checkout <branch-name>`

3. **Create a branch and switch to:**

   `git checkout -b <branch-name>`

Basic Git Commands

1. **Add files to staging area:**

   `git add <file>` or `git add .`

   this command adds changes from working directory to staging area

2. **Commit changes to be ready to push:**

   `git commit -m "commit message"`

   this command adds staged changes to local repository

Basic Git Commands (cont.)

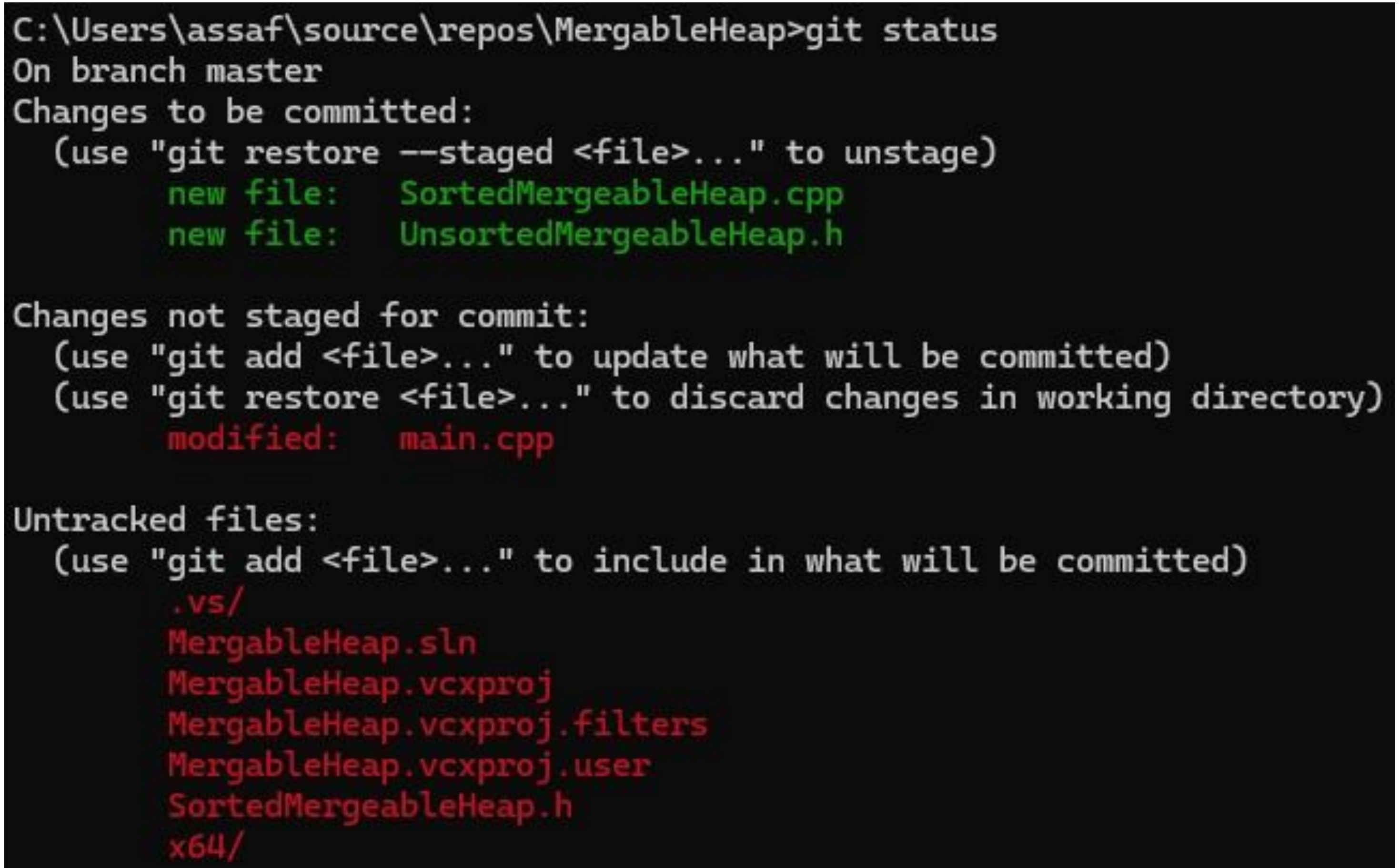
1. **Check status of all files that are in working directory & staging area:**

   `git status`

2. **List commits history of this branch, to see what's in local repository:**

   `git log`

`git status` example

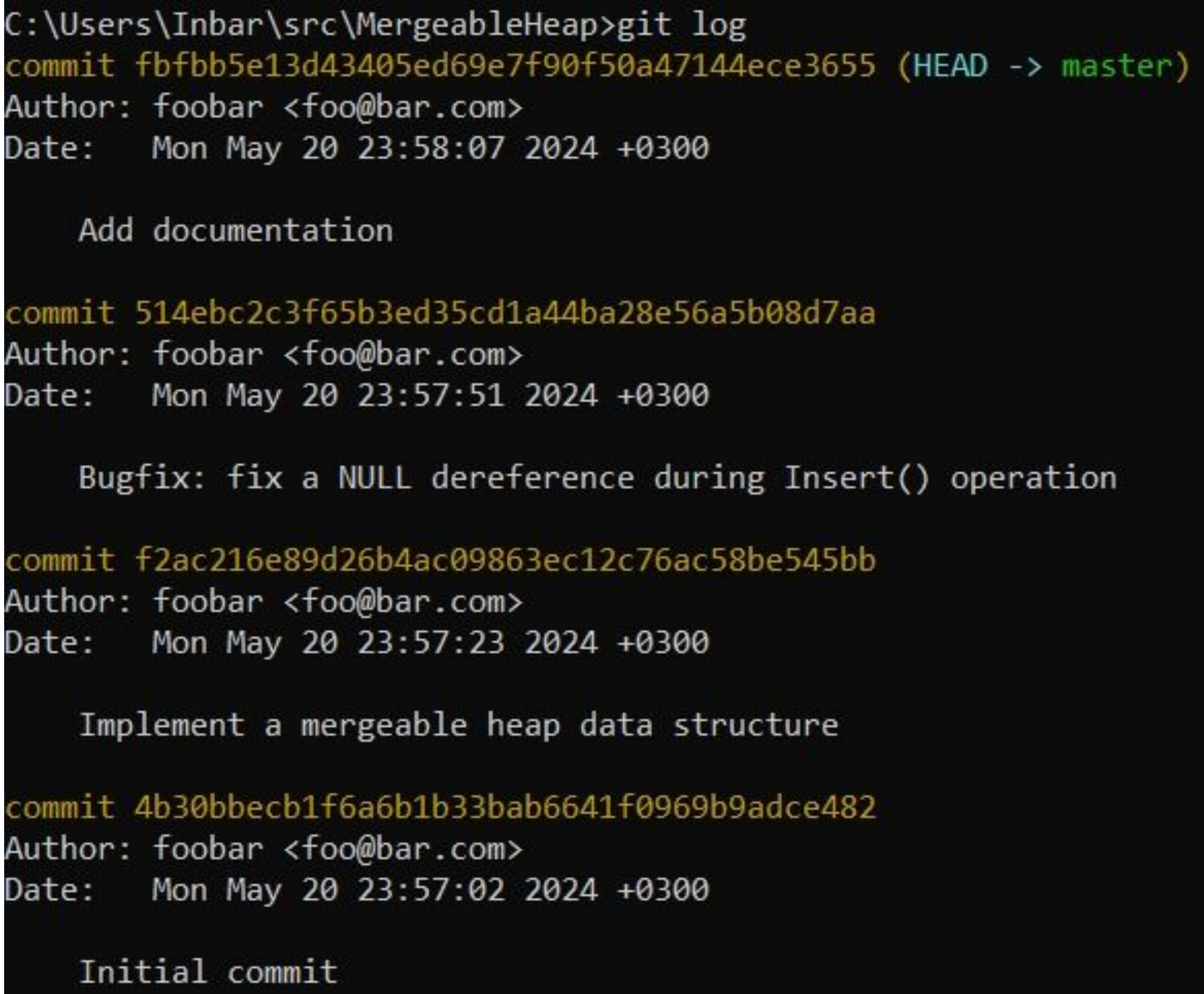


```
C:\Users\assaf\source\repos\MergableHeap>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   SortedMergeableHeap.cpp
        new file:   UnsortedMergeableHeap.h

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   main.cpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .vs/
        MergableHeap.sln
        MergableHeap.vcxproj
        MergableHeap.vcxproj.filters
        MergableHeap.vcxproj.user
        SortedMergeableHeap.h
        x64/
```

`*git log*` example



```
C:\Users\Inbar\src\MergeableHeap>git log
commit fbfbb5e13d43405ed69e7f90f50a47144ece3655 (HEAD -> master)
Author: foobar <foo@bar.com>
Date:   Mon May 20 23:58:07 2024 +0300

    Add documentation

commit 514ebc2c3f65b3ed35cd1a44ba28e56a5b08d7aa
Author: foobar <foo@bar.com>
Date:   Mon May 20 23:57:51 2024 +0300

    Bugfix: fix a NULL dereference during Insert() operation

commit f2ac216e89d26b4ac09863ec12c76ac58be545bb
Author: foobar <foo@bar.com>
Date:   Mon May 20 23:57:23 2024 +0300

    Implement a mergeable heap data structure

commit 4b30bbecb1f6a6b1b33bab6641f0969b9adce482
Author: foobar <foo@bar.com>
Date:   Mon May 20 23:57:02 2024 +0300

    Initial commit
```

Basic Git Commands (cont.)

1. **Push changes:**

   `git push origin <branch-name>`

2. **Pull changes:**

   `git pull origin <branch-name>`

Git Workflow Example

1. **Clone repository:**

   `git clone <url>`

2. **Create a branch:**

   `git checkout -b new-feature`

3. **Make changes and commit:**

   `git add .`

   `git commit -m <”Added new feature">`

4. **Push changes:**

   `git push origin new-feature`

5. **Create a pull request on GitHub**

Some Best Practices

- **Commit often:** Save progress frequently
- **Write meaningful commit messages**
- **Use branches:** Isolate features and bug fixes
- **Pull before pushing:** Avoid conflicts by updating your branch

Resources

- Git Commands
- Git Terminology
- Pro Git Book
- A Very-Good Medium page about Git