MSML 605 - Lecture 8

Version Control
Introduction

- New changes keep coming up
- We need to keep track of the changes
- There is some mechanism to see what changed overtime and undo certain actions
System Administration

- New software is updated, may be server updates
- If something breaks, without version control it would be a hassle.
- With version control you could rollback to a stable version.
Keeping track

- Making copies of the work

- Something that was removed only to be added later

- Keeping historical copies is elementary version control (primitive).

- Who did what and why is lost.
Comparing files

- Two copies of the same code from different times
- Eyeball them?
- `diff` command
- `meld, kdiff3, vimdiff`
- `patch`
Version Control

- Keeps track of all the versions
- Helps retrieve past versions and who changed the files and when
- Files are organized in repositories
- A repository can have thousands of contributors.
Git

- Creator - Linus Torvalds

- Open source - originally created to manage development of Linux kernel

- Distributed architecture - copies on the local as well as the remote machines.

- Git can be used as a server or a client or both on a machine.
Git

- Clients can communicate with the server over http, ssh or git own protocol.

- Multi-platform support

- Public servers GitHub, gitlab, etc.

- Official Git website, https://git-scm.com/ (scm - source control management)

- Other tools like Subversion (https://subversion.apache.org/), Mercurial (https://www.mercurial-scm.org/) etc.
Git - Installation

- `git --version` to check if it is already installed
- `conda install git`
  
  
  download page - [https://git-scm.com/downloads](https://git-scm.com/downloads)

- Windows comes with the environment *MinGW64* to operate like Linux

- There are some IDE’s available
Git commands

- To keep track of who made the changes
  
  `git config --global user.email “email”`
  
  `git config --global user.name “name”`
  
- User name and email are global for all git repositories

- Git init
  
  initializes a repository in the current directory
  
  `ls -la`

  `ls -l`
Git directory

- `git clone` creates a similar directory as `git init`

- Area outside the Git directory is the working directory - current version of the project

- Create a new file or copy an existing file.

- Add it to the staging area.
  
  `git add <filename>`
Staging area and commit

- Git directory, working tree and staging area

- *git status*

- *git commit*

- File Stages: modified, staged, committed
Tracking

- Check status
- Modify a file that is under version control
- Check status again
- Add the file to the staging area
- Commit the changes
- Check status
Git workflow

- `git log`
No staging area

- Any changes to tracked files

  `git commit -a -m "message"`

- HEAD alias represent the currently checked out snapshot of the repository

- HEAD can be a commit in a different branch of a project
Git log

- `git log -p`
  
  *p for patches*

- `git show`

- `git log --stat`
Git working area

- `git diff`
  difference before commit

- `git add -p`

- `git diff —staged`
Removing and renaming files

- `Git rm <filename>`
- `Git commit -m “message”`
- `Git mv <old file name> <new file name>`
- `Git commit -m “message”`
Ignore some files and remove changes

- Add them to .gitignore file
  echo .abc > .gitignore

  example: https://gist.github.com/octocat/9257657

- git checkout <filename>
  to undo changes made that you don’t want to add to the staging area.
remove changes in staged files

- `git restore --staged <filename>`
  to undo changes made and added to the staging area.

- `git restore <filename>`
Wrong commit message

- Create two files
- Add one file
- Commit one file with the message that two files were added
- Then add second file
- git commit —amend
to fix the message

Don’t use for public repositories
Wrong commit - undo last commit

- `git revert HEAD`
- `git log -p -2`
Wrong commit - by commit id

- `git log -1`

```bash
(msmlDockerApp) bash-3.2$ git log -1
commit 0107f7c9e12fc1c8670d490d408b77ac73b59fc3 (HEAD -> master)
```

- `git log`

- `git show`

```bash
git show 398f4175772bc2058a42db7500b355262d3b2009
```

- `git revert`

```bash
git revert 398f4175772bc2058a42db7500b355262d3b2009
```
Branches

- `git branch`
  command to list, create, delete and manipulate branches

- `git branch`

- `git branch new-development`

- `git checkout new-development`

- `git checkout -b new-branch`
Branches

- `git checkout -b new-branch`

- *Create a new file (list_numbers.py), add and commit it*

- `git log -1`

```
commit 0527648028ff19a907adc9187ba8e031f65fe653 (HEAD -> new-branch)
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date:   Tue Apr 28 16:55:32 2020 -0400

    Added list_number to the new-branch
```
Branches

- `git status`

  On branch new-branch
  nothing to commit, working tree clean

- `git checkout master`

  Switched to branch 'master'

- `git log -2`

- `ls -l`

  -rwxr-xr-x 1 nayeem staff 134 Apr 28 16:23 code.py
  -rw-r--r-- 1 nayeem staff 0 Apr 16 17:49 file1.py
  -rw-r--r-- 1 nayeem staff 0 Apr 16 17:49 file2.py
Branches

- `git branch`

- `git branch -d new-development`
  
  Deleted branch new-development (was daa034c).

- `git branch -d new-branch`
  
  error: The branch 'new-branch' is not fully merged. If you are sure you want to delete it, run 'git branch -D new-branch'.
Merging

- `git branch`
  
  ```
  * master
  new-branch
  ```

- `git merge new-branch`
  
  ```
  Updating daa034c..0527648
  Fast-forward
  list_number.py | 3 +++
  1 file changed, 3 insertions(+)
  create mode 100644 list_number.py
  ```

- `git merge`
  
  uses fast forward and three-way merge
Merge Conflicts

- `git branch`
  
  ```
  * master
  new-branch
  ```

- Modify a file and commit to the master branch
  
  ```
  git commit -a -m "print line added"
  ```

- Checkout a branch
  
  ```
  git checkout new-branch
  ```
Merge Conflicts

- Modify the same file in the new branch and commit to it

  ```
  git commit -a -m "Added print "Done" to list_number"
  ```

- Now Checkout the master branch

  ```
  git commit -a -m "Added print "Done" to list_number"
  ```

- Now merge the branch

  ```
  git merge new-branch
  ```

  Auto-merging list_number.py
  CONFLICT (content): Merge conflict in list_number.py
  Automatic merge failed; fix conflicts and then commit the result.
Merge Conflicts

- **Check the status.** - `git status`

On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
  both modified: list_number.py

no changes added to commit (use "git add" and/or "git commit -a")

- **See the file for the conflicts**

```bash
#!/anaconda3/bin/python
for i in range(10):
    print(i)

<<<<<< HEAD
print(i)
=======
print("Done")
>>>>>>>> new-branch
```
Merge Conflicts

- Keep both changes and add the file
  
  ```
  git add list_number.py
  ```

- Check the status

  ```
  git status
  ```
  
  On branch master
  All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

  Changes to be committed:
  
  modified:  list_number.py

- git commit
Merge Conflicts

- `git commit`

```bash
Merge branch 'new-branch'

# Conflicts:
#   list_number.py
#
# It looks like you may be committing a merge.
# If this is not correct, please remove the file
#   .git/MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# All conflicts fixed but you are still merging.
#
# Changes to be committed:
#   modified: list_number.py
```

- Add a comment like, Keeping changes from both branches
Merge Conflicts

- **Check the commit history**
  
  ```
  * d593708 (HEAD -> master) Merge branch 'new-branch'
  |  
  | * 5b822fd (new-branch) Added print Done to list_number
  | | a1262ab print line added
  |/
  * 0527648 Added list_number to the new-branch
  * daa034c Revert "code.py renamed to py_code.py"
  * 0107f7c Revert "Added nomsese" Roll back because of errors
  * a02b528 Added nomsese
  * 4d9883d Added two files file1 to verify rrsults file2 to display results
  * 5ble353 added .gitignore file
  * 398f417 code.py renamed to py_code.py
  * feldedf removed test.py since it was empty and useless
  * 2ede863 Added empty test.py file
  * 8b4ce55 Added done at the end
  * f9b2351 print pandas version
  * df367a2 imported pandas library
  * 74d16f6 first version of code.py file
  ```

- `git merge --abort` to reset to the previous commit to throw away the changes.
Remote repository

- Github
- Provides public and private repositories
- You will need an account. It is free.
Remote repository

Create a repository - test

Owner

Repository name *

nayeemmz /

Great repository names are short and memorable. Need inspiration? How about fuzzy-octo-engine?

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

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

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

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None
Add a license: None

Create repository
Remote repository

- New repository

https://github.com/nayeemzmz/test.git
Clone - Remote repository

- **Clone a repository**

```bash
git clone https://github.com/nayeemmz/test.git
```

Cloning into 'test'...
Username for 'https://github.com':
Password for 'https://nayeemmz@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

- A directory is created locally
Update local repository

- Go to local test repository and update README file
- Commit the changes
  
  ```
  git commit -m "Added two lines to README.md file"
  ```

- Push the changes to the remote repository
  
  ```
  git push
  ```
  
  Enumerating objects: 5, done.
  Counting objects: 100% (5/5), done.
  Delta compression using up to 4 threads
  Compressing objects: 100% (2/2), done.
  Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
  Total 3 (delta 0), reused 0 (delta 0)
  To https://github.com/nayeemmz/test.git
    2ecc9ac..e2da63b master -> master
Check the remote repository

- remote repository

<table>
<thead>
<tr>
<th>Branch: master</th>
<th>New pull request</th>
<th>Create new file</th>
<th>Upload files</th>
<th>Find file</th>
<th>Clone or download</th>
</tr>
</thead>
<tbody>
<tr>
<td>nayeemnzz Added two lines to README.md file</td>
<td></td>
<td></td>
<td></td>
<td>Latest commit e2da63b 4 minutes ago</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>README.md</th>
<th>Added two lines to README.md file</th>
<th>4 minutes ago</th>
</tr>
</thead>
</table>

- test

This repo is our first repository. We will populate it with fancy ML scripts.
Avoid Authentication

- Use ssh or

- Credential helper -

  ```
  git config --global credential.helper cache
  ```

- Pull the remote repository

  ```
  git pull
  Username for 'https://github.com':
  Password for 'https://nayeemxz@github.com':
  Already up to date.
  ```

- Pull the remote repository again

  ```
  git pull
  Already up to date.
  ```
Remote repository

- Use ssh or https
- Check the remote url
- Show origin on remote computer

```
$ git remote -v
origin https://github.com/nayeemzz/test.git (fetch)
origin https://github.com/nayeemzz/test.git (push)
```

```
$ git remote
* remote origin
  Fetch URL: https://github.com/nayeemzz/test.git
  Push URL: https://github.com/nayeemzz/test.git
  HEAD branch: master
  Remote branch:
    master tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```
Remote repository

- Remote branches

```bash
git branch -r
```

```
origin/HEAD -> origin/master
origin/master
```

- Check status

```bash
git status
```

```
On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean
```
Remote repository

- **Update an commit through Github**

  test

  This repo is our first repository. We will populate it with fancy ML scripts.

  Updating on Github

- **Check remote repository locally**

  ```
  * remote origin
    Fetch URL: https://github.com/nayeemz/test.git
    Push URL: https://github.com/nayeemz/test.git
    HEAD branch: master
    Remote branch:
      master tracked
    Local branch configured for 'git pull':
      master merges with remote master
    Local ref configured for 'git push':
      master pushes to master (local out of date)
  ```
Syncing

- **Sync remote and local repositories**

```plaintext
git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/nayeemmx/test
e2da63b..53ec7d8 master    -> origin/master
```
Remote log

- Check the log of the remote repository

```bash
$ git log origin/master
commit 53ec7d8746f15ac51c9388e9ae40d9477bfbf35f (origin/master, origin/HEAD)
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date:   Thu Apr 30 15:52:15 2020 -0400

Update README.md

commit e2da63b4f06afd7a40747b704a4088b5dc93bf62 (HEAD -> master)
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date:   Wed Apr 29 16:07:06 2020 -0400

Added two lines to README.md file

commit 2ecc9ac4606ace3e394bddd4ce0c8e0bf48f8e556
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date:   Wed Apr 29 15:37:15 2020 -0400

Initial commit
```
Missing commit

- **Check status**

  ```
  git status
  On branch master
  Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.  
  (use "git pull" to update your local branch)
  nothing to commit, working tree clean
  ```

- **Merge the changes**

  ```
  git merge origin/master
  Updating e2da63b..53ec7d8
  Fast-forward
  README.md | 2 ++
  1 file changed, 2 insertions(+)
  ```
Check the log again

- Check status

```bash
git log
```

```
commit 53ec7d8746f15ac51c9388e9ae40d9477bfbf35f (HEAD -> master, origin/master, origin/HEAD)
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date: Thu Apr 30 15:52:15 2020 -0400

  Update README.md

commit e2da63b4f06af7a40747b704a4088b5dc93bf62
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date: Wed Apr 29 16:07:06 2020 -0400

  Added two lines to README.md file

commit 2ecc9ac4606ace3e394bddd4ce0c8e0bf48f8e556
Author: Mohammad Nayeem Teli <mohammad.nayeem@gmail.com>
Date: Wed Apr 29 15:37:15 2020 -0400

  Initial commit
```
## Conflicts

- **Update a file locally**
- **Check the changes to accept**

```bash
git add -p
```

```diff
diff --git a/README.md b/README.md
index 6f48ba6..59d7539 100644
--- a/README.md
+++ b/README.md
@@ -4,3 +4,5 @@ This repo is our first repository.
    We will populate it with fancy ML scripts.

    Updating on Github
+
+New changes for conflicts
Stage this hunk [y,n,q,a,d,e,?]? y
```

- **Commit**

```bash
git commit -m "Added a line to test conflicts"
```
Conflicts

- Update the file in the remote repository on Github

This repo is our first repository. We will populate it with fancy ML scripts.

Updating on Github

Added text to test conflicts on Github

- Push the change you made locally

Username for 'https://github.com': nayeemmmz
Password for 'https://nayeemmmz@github.com':
To https://github.com/nayeemmmz/test.git
  ! [rejected] master -> master (fetch first)
error: failed to push some refs to 'https://github.com/nayeemmmz/test.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.
Conflicts

- **Merge first**
  ```
  git pull
  
  remote: Enumerating objects: 5, done.
  remote: Counting objects: 100% (5/5), done.
  remote: Compressing objects: 100% (2/2), done.
  remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
  Unpacking objects: 100% (3/3), done.
  From https://github.com/nayeemmm/test
    53ec7d8..e81cc08 master    --> origin/master
  Auto-merging README.md
  CONFLICT (content): Merge conflict in README.md
  Automatic merge failed; fix conflicts and then commit the result.
  ```

- **Check log**
  ```
  git log --graph --oneline --all
  
  * e81cc08 (origin/master, origin/HEAD) Update README.md
  | * bd79b98 (HEAD -> master) Added a line to test conflicts
  |
  * 53ec7d8 Update README.md
  * e2da63b Added two lines to README.md file
  * 2ecc9ac Initial commit
  ```
Conflicts

- Check the commits on origin/master

```
commit e81cc08dce09bb6df20086acfc9f4b327d2ff0d07 (origin/master, origin/HEAD)
Author: Mohammad Nayeen Teli <mohammad.nayeen@gmail.com>
Date: Thu Apr 30 16:40:31 2020 -0400

  Update README.md

diff --git a/README.md b/README.md
index 6f64ba6..d163ffec 100644
--- a/README.md
+++ b/README.md
@@ -4,3 +4,3 @@ This repo is our first repository.
        We will populate it with fancy ML scripts.

    Updating on Github
+    +Added text to test conflicts on Github

commit 53ec7d746f5ace51c9388e99e4d99477b9bf35f
Author: Mohammad Nayeen Teli <mohammad.nayeen@gmail.com>
Date: Thu Apr 30 15:52:15 2020 -0400

  Update README.md

diff --git a/README.md b/README.md
index 472187a..6f48ba6 100644
--- a/README.md
+++ b/README.md
@@ -2,3 +2,3 @@
    This repo is our first repository.
        We will populate it with fancy ML scripts.

    Updating on Github

commit e2da63bf406ad7a46747b704a4088b5d93bf662
Author: Mohammad Nayeen Teli <mohammad.nayeen@gmail.com>
Date: Wed Apr 29 16:07:06 2020 -0400

  Added two lines to README.md file

diff --git a/README.md b/README.md
index 008cbe6..472187a 100644
--- a/README.md
+++ b/README.md
@@ -1,4 +1,4 @@
```
Conflicts

- Correct or merge the multiple changes in the file
- Add the file
  
  ```
  vi README.md
  git add README.md
  ```
- Commit the changes
  
  ```
  git commit -m "Merged the changes from local and remote"
  ```
- Push the changes

  ```
  (base)Mohammds-MBP:test nayeem$ git push
  Username for 'https://github.com': nayeemzz
  Password for 'https://nayeemzz@github.com':
  Enumerating objects: 10, done.
  Counting objects: 100% (10/10), done.
  Delta compression using up to 4 threads
  Compressing objects: 100% (4/4), done.
  Writing objects: 100% (6/6), 594 bytes | 297.00 KiB/s, done.
  Total 6 (delta 2), reused 0 (delta 0)
  remote: Resolving deltas: 100% (2/2), completed with 1 local object.
  To https://github.com/nayeemzz/test.git
  e81cc08..71020b5 master --> master
  ```
Conflicts

- Check the log to verify

```
(base) Mohammads-MBP:test nayeem$ git log --onograph --oneline
fatal: unrecognized argument: --onograph
(base) Mohammads-MBP:test nayeem$ git log --graph --oneline
* 71020b5 (HEAD -> master, origin/master, origin/HEAD) Merged the changes from local and remote
|\  
| * e81cc08 Update README.md
* | bd79b98 Added a line to test conflicts
|/
* 53ec7d8 Update README.md
* e2da63b Added two lines to README.md file
* 2ecc9ac Initial commit
```