COMPUTER SCIENCE

UNIT F WEEK 1, TUESDAY MAR 20TH + THURSDAY MAR 22ND

WWW.CITA.UTORONTO.CA/~WOODFORD

WOODFORD@CITA.UTORONTO.CA

THIS WEEK IN CS AND STEM

- Mapping the Worlds Oceans
  - https://futurism.com/underwater-robots-map-marine-microbes/

- Micro-needles that intentionally break off
  - https://futurism.com/microneedles-dissolve-skin/

FINAL PROJECTS

- 3 different project descriptions, you need to:
  - Complete the coding, using all of the skills we've learned
  - Write user documentation
  - Write a report
  - Give a 10 minute presentation showcasing your work (last day of classes, June 5th)
- Due dates:
  - Pick your project (1 of the 3 given): April 3rd – tell me in person or in email!
  - Submit Report, code, supporting documents: June 4th at midnight
  - Give presentation: June 5th in class

PROPER CODE DOCUMENTATION

- We've been talking about this throughout the whole course, but we should take some time to address proper documentation and what's expected in industry.

PROPER CODE DOCUMENTATION

- We've been talking about this throughout the whole course, but we should take some time to address proper documentation and what's expected in industry.
  - Docstrings
  - Block comments
  - Line comments
  - Help files
  - User manuals
  - Training documentation

PRACTISE!

- I have a sample code with NO documentation. We'll need to add:
  - Docstrings
  - Block comments
  - Line comments
- As well as brainstorm what would need to go in:
  - User manual
  - Training documentation
  - Help file(s)

INTRO TO VERSION CONTROL

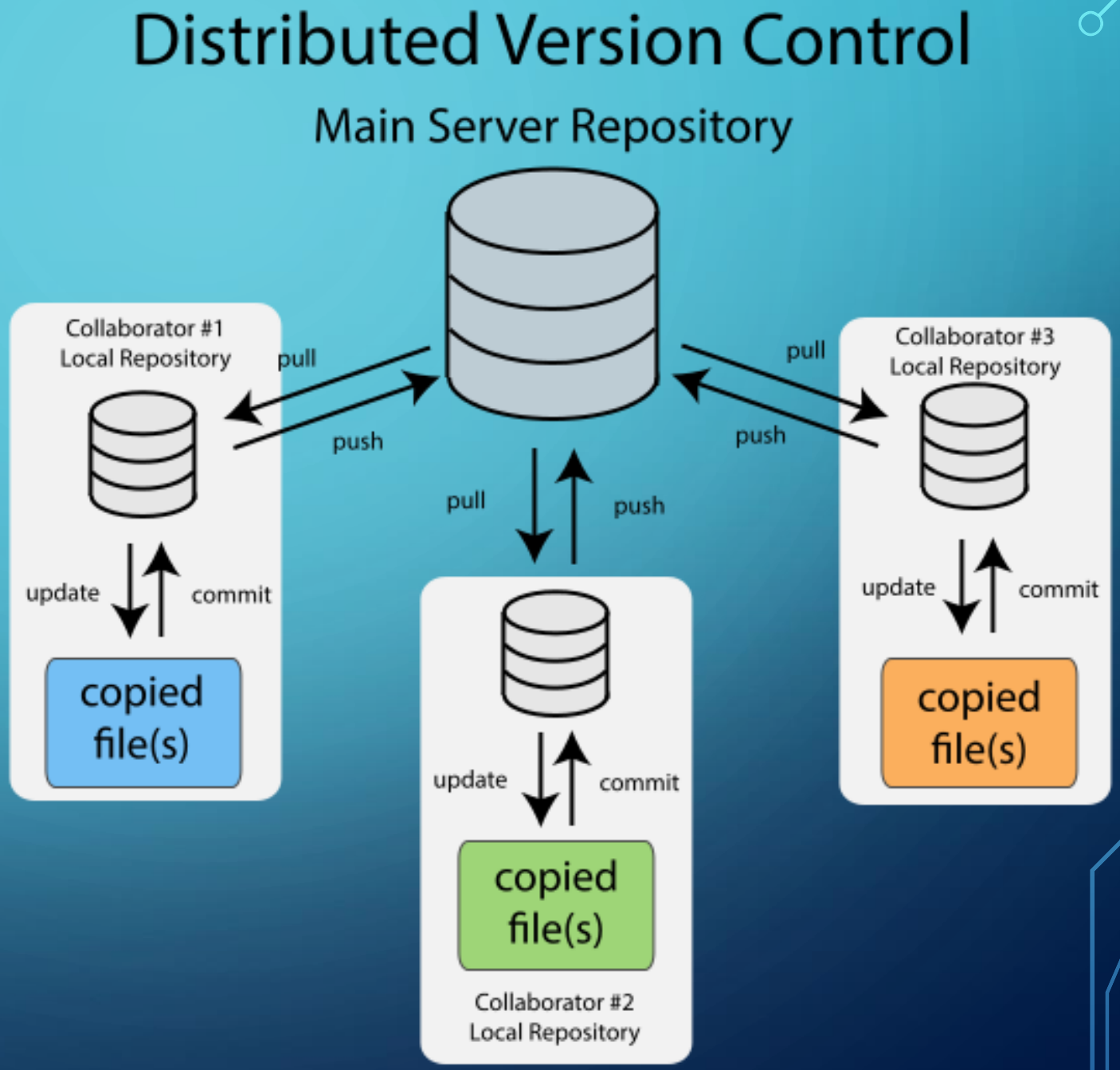
- What is version control? Why is it important?

INTRO TO VERSION CONTROL

- What is version control? Why is it important?
- We typically need to the following when working on shared or publically available codes:
  - Backup and Restore.
  - Synchronization.
  - Short-term undo.
  - Long-term undo.
  - Track Changes.
  - Track Ownership.
  - Branching and merging.

Key Terms:

- Repository
- Server
- Client
- Working Copy
- Main
- Check out/Pull
- Check in/Push
- Check in Message/ Commit
- Log/History
- Revert
- Branch
- Merge



GIT

- Most common version control system
- Constantly being upgraded and revised
- Friendly with Linux, Windows, macOS operating systems
- Use in terminal

GIT

- Developed in 2005 by Linus Torvalds to track files while developing the Linux Kernel.
- Used mostly for source code management in software development, but widely used in the academic community as well.
- Description from the source code:

“The name "git" was given by Linus Torvalds when he wrote the very first version. He described the tool as "the stupid content tracker" and the name as (depending on your way):

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "goddamn idiotic truckload of shit": when it breaks”

GIT : COMMANDS

- When you have a repository that uses git and have cloned it onto your machine, the most common commands are:
  - git push
  - git pull
  - git add [ ]
  - git commit –m [ ]
  - git log
  - git status

GIT CLONE

- First we need to get access to a repo.

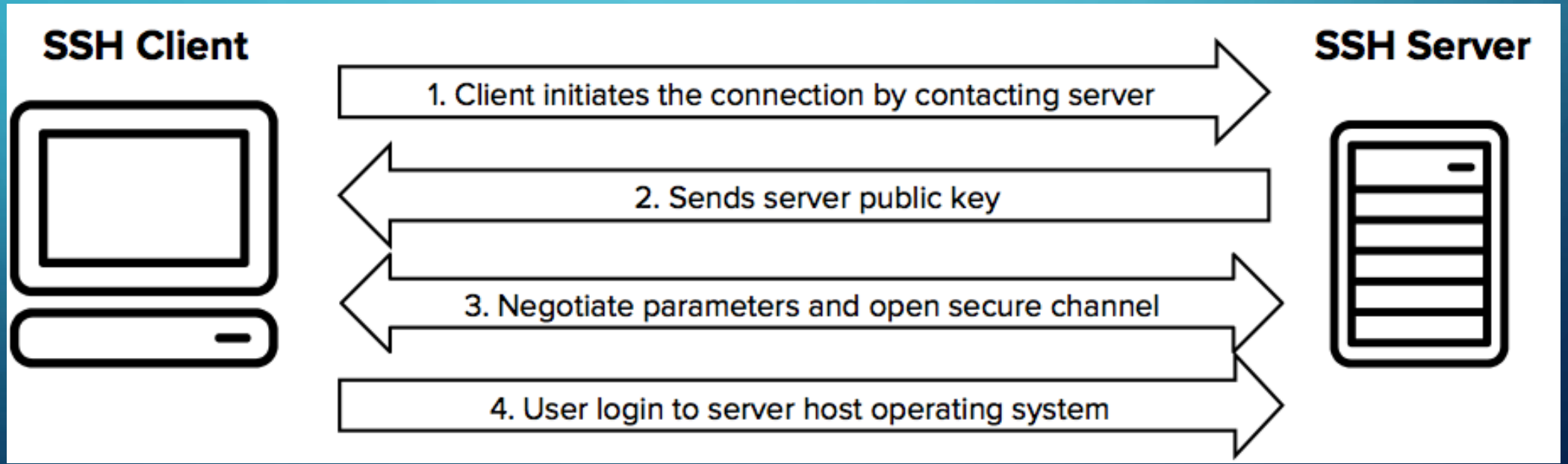
- https://github.com/cwoodford/Abelard_CS
  - Make your own github account – it will allow you access to public repos on github like the one I've initialized above
  - You won't be able to clone yet, but let's look at the repo and figure out what we need to do to access it!

SSH AND TUNNELING

- In order to access the repo from your local machine, you need to git clone. There two ways to do this:
  - SSH
  - HTTPS
- You NEVER want to use HTTPS. So, we need to lean how to SSH.

SSH AND TUNNELING

- SSH = Secure Shell, used for “tunneling” between servers, systems, and repositories. It uses encryption through key-pairs to ensure security.



BACK IN THE DAY…

- It's no surprise that communications between servers needs to be encrypted. Before 1995, there were several tools that assisted with keeping communication and information "secure", such as telnet, ftp, FTP/S, rlogin, rsh, and rcp.
- Unfortunately, these methods were weak to "password sniffers", as they relied on username and password combinations that can be cracked relatively easily. Tatu Ylonen developed SSH in 1995 after his company network was attacked and thousands of username/password pairs were leaked – meaning personal as well as company information was no longer secure.
- SSH uses the key pairs for automated authentication and randomly generated encryption keys for passing information – it's used to manage over ½ the world's web servers!

BACK TO OUR REPO…

- How to generate keys and put them into GitHub: https://help.github.com/enterprise/2.12/user/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/

- Let's try this together!

COMMON COMMANDS

- We've already discussed git commands that are used constantly, but there are a few more that will also come in handy:
  - git init (for a repo that isn't being tracked by git, you can start tracking it using this)
  - git rm [ ] (remove a file, files, or whole directories. This removes them from the local repo.)
  - git reset [--hard, --soft] (undo recent commits in the local repo, can be the most recent or specify up to a specific commit)
  - **git config –global core.editor ["emacs –nw", vim] (automatically open an editor to write your message when using "git commit")**
- Note that all of the git commands we've seen and used have many options and flags. Use the "-h" flag to see all the options and their meaning, or search on the git website for further descriptions and examples!

ASSIGNMENT 13

- Due Sunday April 1st at midnight
- Using whichever server you prefer (github, bitbucket, etc), make your own git repo to house material for your final project. Note that you can keep everything and anything in the repo: code, notebooks, pdf, images, txt, Microsoft/Libre office documents, etc. Name the repo appropriately! It should have a ReadMe and at least 1 other file (any kind of file). Subsequently, there should be at least 2 commits (initial commit and one for adding the file).
- Send me the url to your repo by email

REFERENCES

- https://betterexplained.com/articles/a-visual-guide-to-version-control/
- https://www.visualstudio.com/learn/what-is-version-control/
- https://git-scm.com/
- https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- https://www.ssh.com/ssh/protocol/
- https://sethrobertson.github.io/GitFixUm/fixup.html