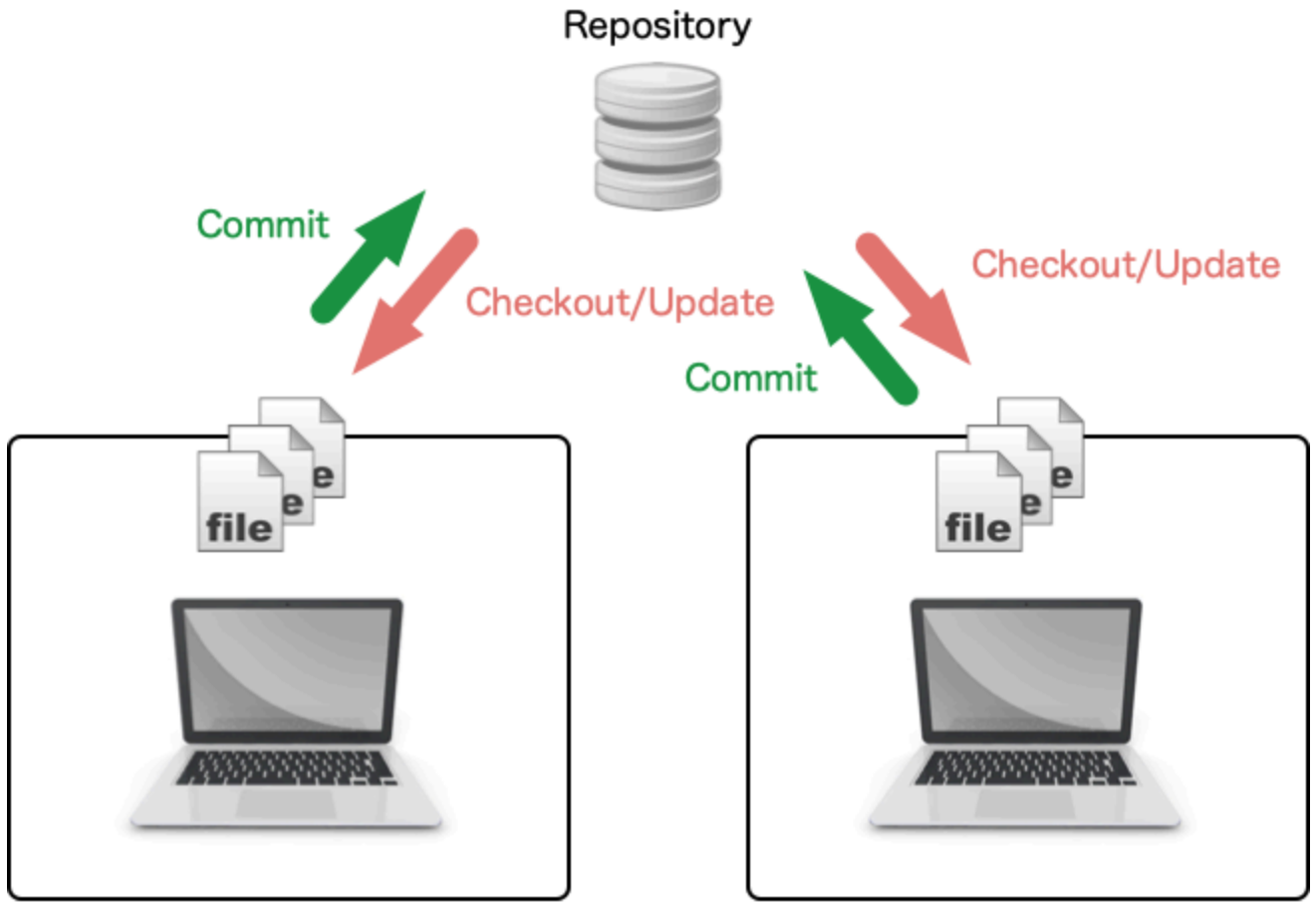
Git

Version Control System (VCS)

Basic Concepts

- Repository: A database to store artifacts of development
  - Data: Source codes, specifications (documents), test codes, etc.
  - History of changes
- Checkout: Load a specific version in the repository to local files
- Commit: Save local files to the repository (the change history is saved)


Repository
Commit
Checkout/Update
Checkout/Update
Commit
file
file

Advanced Concepts

- Conflict
- Branch
- Merge

Conflict

- Data integrity
  - All the changes are properly applied
  - File lock is the simplest way to gurantee the data integrity
- Conflict
  - The same position have been changed by two or more users.
  - The history of changes is undetermined.
  - The conflict should be resolved **by hands**.

Branch and Merge

What is branch

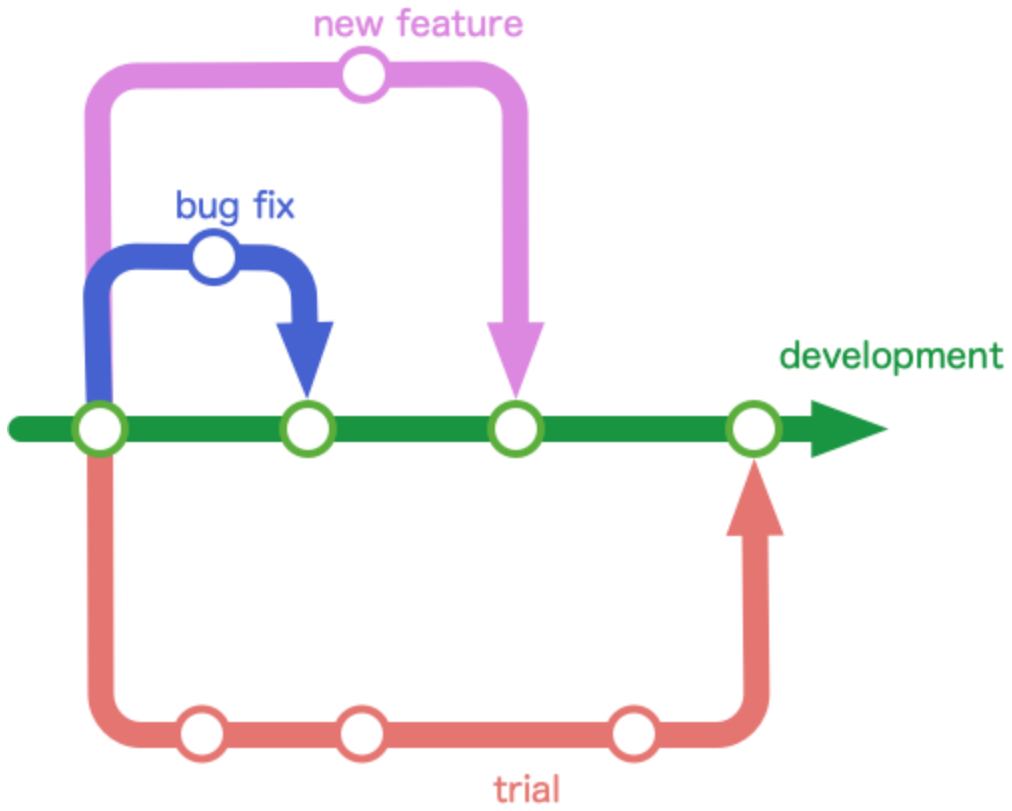
- A branch is a stream of history of changes, the management unit in the repository
  - a branch for development, a branch for releases, a branch for debugging, etc.

How to use

- Making a new branch from the present branch
- Merging changes on a branch to another branch

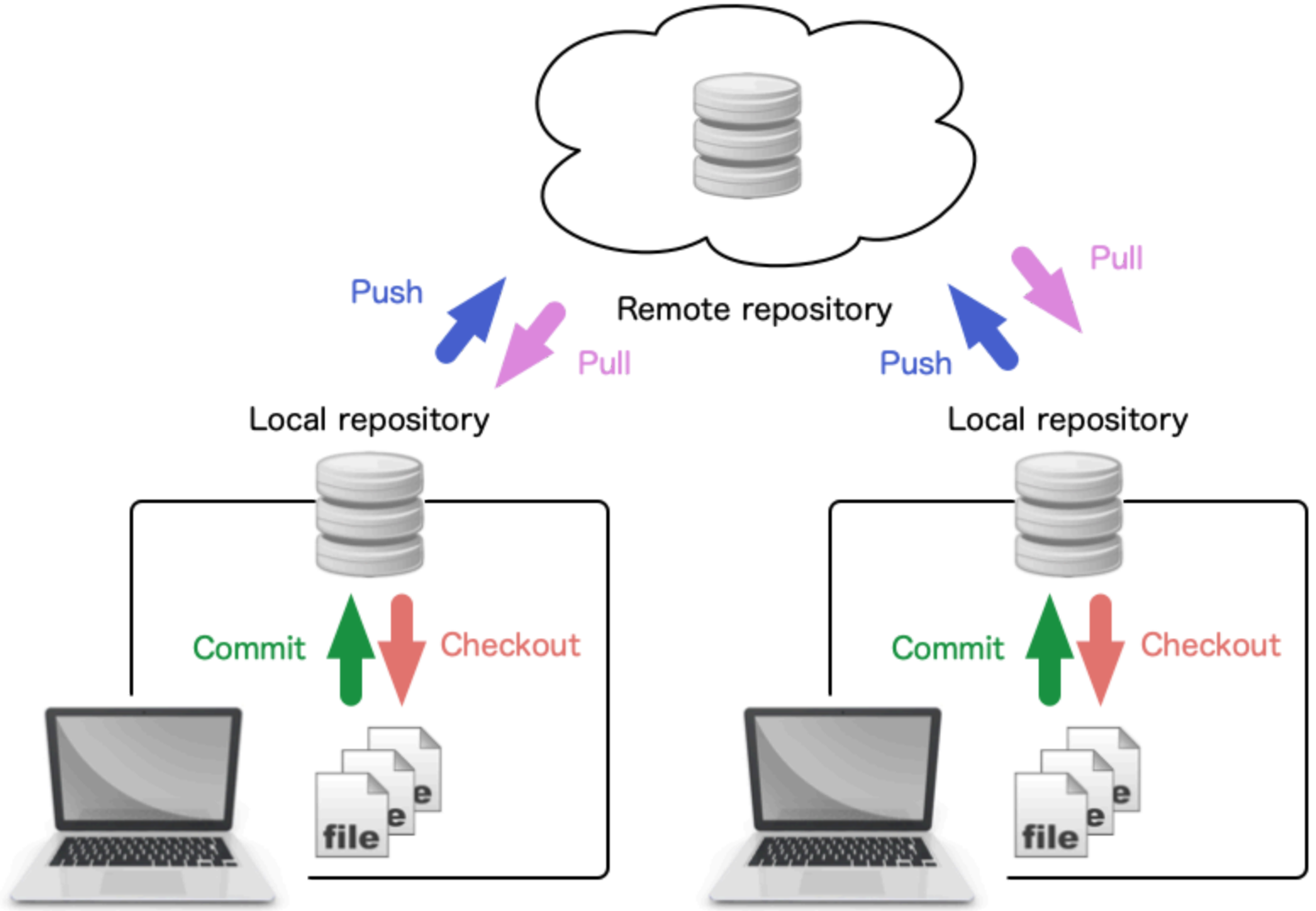
Why

- There is no conflict between different branches.
- The conflict is caused by merging branches.


new feature
bug fix
development
trial

Git

- A distributed version control system
  - This is originally developed on SCM for Linux.
- Two repositories
  - Remote repository located in a server
  - Local repository located in a client (local PC)
- Basic Operations
  - clone: copy a remote repository to a local repository
  - commit: save history of changes to the local repository
  - push: update the remote repository by the local repository
  - pull: update the local repository by the remote repository
- Default branch name: `main` (`master`)


Remote repository
Push
Pull
Pull
Push
Local repository
Local repository
Commit
Checkout
Commit
Checkout
file
file

Usage

- `git xxxx` where `xxxx` is a mode of git command

Basic commands

- `git clone` : Copy a remote repository to a local repository
- `git checkout` : Checkout a branch from the local repository (change a branch)
- `git add` : Add files to stage as files to be committed
- `git commit` : Save the history to the local repository
- `git status` : Display the status of the local repository
- `git branch` : Display branches of the local repository
- `git log` : Display the commit log
- `git push` : Update the remote repository by the local repository
- `git pull` : Update the local repository by the remote repository

Basic Flow

1. Create a local repository (only once at the beginning)
2. Update the local repository with the remote repository
3. Checkout the files from the local repository
4. Edit the files
5. Add the editted files to a stage
6. Commit the history of files that are added to the stage
7. (Repeat Step 4 through Step 6)
8. Update the remote repository with the local repository

Basic Flow

1. Create a local repository (only once at the beginning) **git clone**, **git init**
2. Update the local repository with the remote repository **git pull**
3. Checkout the files from the local repository **git checkout**, **git branch**
4. Edit the files
5. Add the editted files to a stage **git add**, **git status**
6. Commit the history of files that are added to the stage **git commit**, **git log**
7. (Repeat Step 4 through Step 6)
8. Update the remote repository with the local repository **git push**

Summary

- Version Control System (VCS)
  - Repository, Checkout, Commit
  - Conflict, Branch, Merge
- Git
  - Distributed VCS
  - Remote repository, Local repository
  - Basic commands: clone, checkout, add, commit, status, branch, log, push, pull
- Basic flow of Git
  - Create a local repository, Update the local repository, Checkout files, Edit files, Add files to stage, Commit history, Update the remote repository