ParMooN
(Parallel Mathematical Object Oriented Numerics)

Sashikumaar Ganesan

jointly with ...

Sangeeta Yadav,

Computational Mathematics Group
Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India

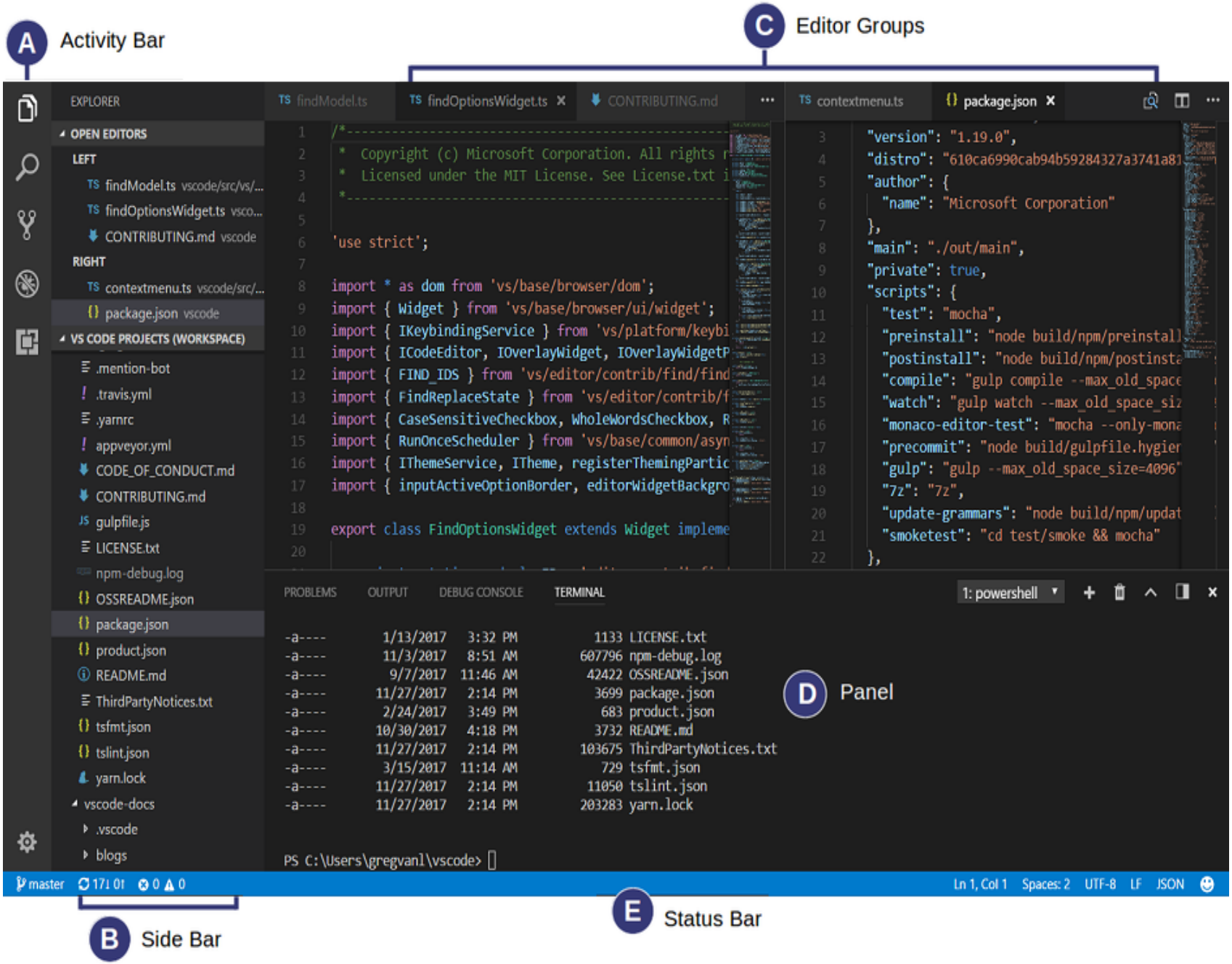
Version 1.0 (2019)

VS CODE

- Developed by Microsoft for Windows, Linux and macOS.
- Includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.
- It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.
- It is free and open source and released under the permissive MIT License.

Basic Layout

1. Editor - The main area to edit your files.
2. Side Bar - Contains different views like the Explorer to assist you while working on your project.
3. Status Bar - Information about the opened project and the files you edit.
4. Activity Bar - Located on the far left-hand side, this lets you switch between views and gives you additional context-specific indicators, like the number of outgoing changes when Git is enabled.
5. Panels - You can display different panels below the editor region for output or debug information, errors and warnings, or an integrated terminal. Panel can also be moved to the right for more vertical space.


A Activity Bar
C Editor Groups
B Side Bar
D Panel
E Status Bar
EXPLORER
OPEN EDITORS
LEFT
findModel.ts vscode/src/vs/...
findOptionsWidget.ts vsco...
CONTRIBUTING.md vscode
RIGHT
contextmenu.ts vscode/src/...
package.json vscode
VS CODE PROJECTS (WORKSPACE)
.mention-bot
.travis.yml
.yarnrc
appveyor.yml
CODE_OF_CONDUCT.md
CONTRIBUTING.md
gulpfile.js
LICENSE.txt
npm-debug.log
OSSREADME.json
package.json
product.json
README.md
ThirdPartyNotices.txt
tsfmt.json
tslint.json
yarn.lock
vscode-docs
.vscode
blogs
findModel.ts
findOptionsWidget.ts
CONTRIBUTING.md
contextmenu.ts
package.json
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
-a---- 1/13/2017 3:32 PM 1133 LICENSE.txt
-a---- 11/3/2017 8:51 AM 607796 npm-debug.log
-a---- 9/7/2017 11:46 AM 42422 OSSREADME.json
-a---- 11/27/2017 2:14 PM 3699 package.json
-a---- 2/24/2017 3:49 PM 683 product.json
-a---- 10/30/2017 4:18 PM 3732 README.md
-a---- 11/27/2017 2:14 PM 103675 ThirdPartyNotices.txt
-a---- 3/15/2017 11:14 AM 729 tsfmt.json
-a---- 11/27/2017 2:14 PM 11050 tslint.json
-a---- 11/27/2017 2:14 PM 203283 yarn.lock
PS C:\Users\gregvanl\vscode>
master
Ln 1, Col 1
Spaces: 2
UTF-8
LF
JSON

SIDE BY SIDE EDITING

1. Alt click on a file in the Explorer.
2. Ctrl+ to split the active editor into two.
3. Open to the Side (Ctrl+Enter) from the Explorer context menu on a file.
4. Click the Split Editor button in the upper right of an editor.
5. Drag and drop a file to any side of the editor region.
6. Ctrl+Enter (macOS: Cmd+Enter) in the Quick Open (Ctrl+P) file list.

ost.contribution.ts | extHostApiCommands.ts ×

```
25
26  class ExtHostApiCommands {
27
28      private _commands: ExtHostCommands;
29      private _disposables: IDisposable[] = [];
30
31      constructor(commands: ExtHostCommands) {
32          this._commands = commands;
33      }
34
35      registerCommands() {
36          this._register('vscode.executeWorkspa
37              description: 'Execute all workspa
38              args: [{ name: 'query', descripti
39              returns: 'A promise that resolves
40
41          });
42          this._register('vscode.executeDefinit
43              description: 'Execute all definit
44              args: [
45                  { name: 'uri', description: '
46                  { name: 'position', descripti
47              ],
48              returns: 'A promise that resolves
49          });
50          this._register('vscode.executeHoverPr
51              description: 'Execute all hover p
52              args: [
53                  { name: 'uri', description: '
54                  { name: 'position', descripti
55              ],
56              returns: 'A promise that resolves
57          });
58          this._register('vscode.executeDocumer
59              description: 'Execute document hi
60              args: [
```

buildfile.js × | workbench.main.css

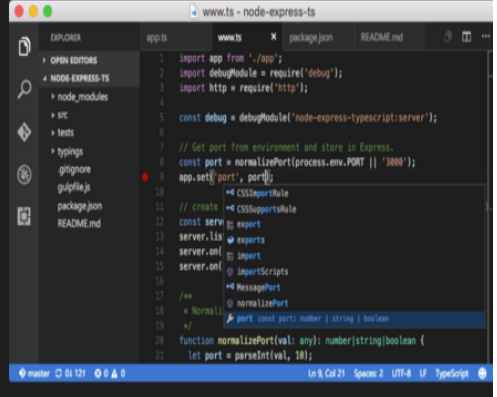
```
5   'use strict';
6
7   function createModuleDescription(name, exclu
8       var result = {};
9       var excludes = ['vs/css', 'vs/nls'];
10      result.name = name;
11      if (Array.isArray(exclude) && exclude.ler
12          excludes = excludes.concat(exclude);
13      }
14      result.exclude = excludes;
15      return result;
16  }
17
18  exports.collectModules = function(excludes) {
19      var languageMainExcludes = ['vs/editor/cc
20      var languageWorkerExcludes = ['vs/base/cc
21
22      var modules = [
23          createModuleDescription('vs/workbench
24          createModuleDescription('vs/workbench
25
26          createModuleDescription('vs/workbench
27          createModuleDescription('vs/workbench
28          createModuleDescription('vs/workbench
29
30          createModuleDescription('vs/workbench
31          createModuleDescription('vs/workbench
32          createModuleDescription('vs/workbench
33
34          createModuleDescription('vs/workbench
35          createModuleDescription('vs/workbench
36          createModuleDescription('vs/workbench
37
38          createModuleDescription('vs/workbench
39
40          createModuleDescription('vs/workbench
```

Preview 'README.md' × | issue_template.md

Visual Studio Code - Open Source

build passing | build passing

VS Code is a new type of tool that combines the simplicity of a code editor with what developers need for their core edit-build-debug cycle. Code provides comprehensive editing and debugging support, an extensibility model, and lightweight integration with existing tools.



The vscode repository is where we do development and there are many ways you can participate in the project, for example:

- Submit bugs and feature requests and help us verify as they are checked in
- Review source code changes
- Review the documentation and make pull requests for anything from typos to new content

master 1 0 Ln 33, Col 6 Tab Size: 4 UTF-8 CRLF TypeScript

LANGUAGE SPECIFIC EDITOR SETTINGS

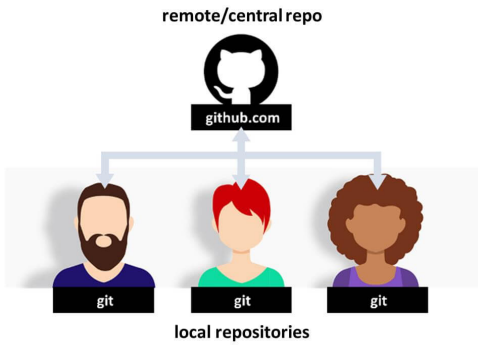
Customize the editor by language:

- Run the global command Preferences: Configure Language Specific Settings (command id: workbench.action.configureLanguageBasedSettings) from the Command Palette (Ctrl+Shift+P) which opens the language picker.
- Selecting the required language, opens the Settings editor with the language entry where you can add applicable settings.

VERSION CONTROL SYSTEM(GIT)

It is a software that helps software developers to work together and maintain a complete history of their work.The functions of a VCS are as follows:

1. Allows developers to work simultaneously.
2. Does not allow overwriting each others changes.
3. Maintains a history of every version.



Types of VCS :

1. Centralized version control system (CVCS).
2. Distributed/Decentralized version control system (DVCS).

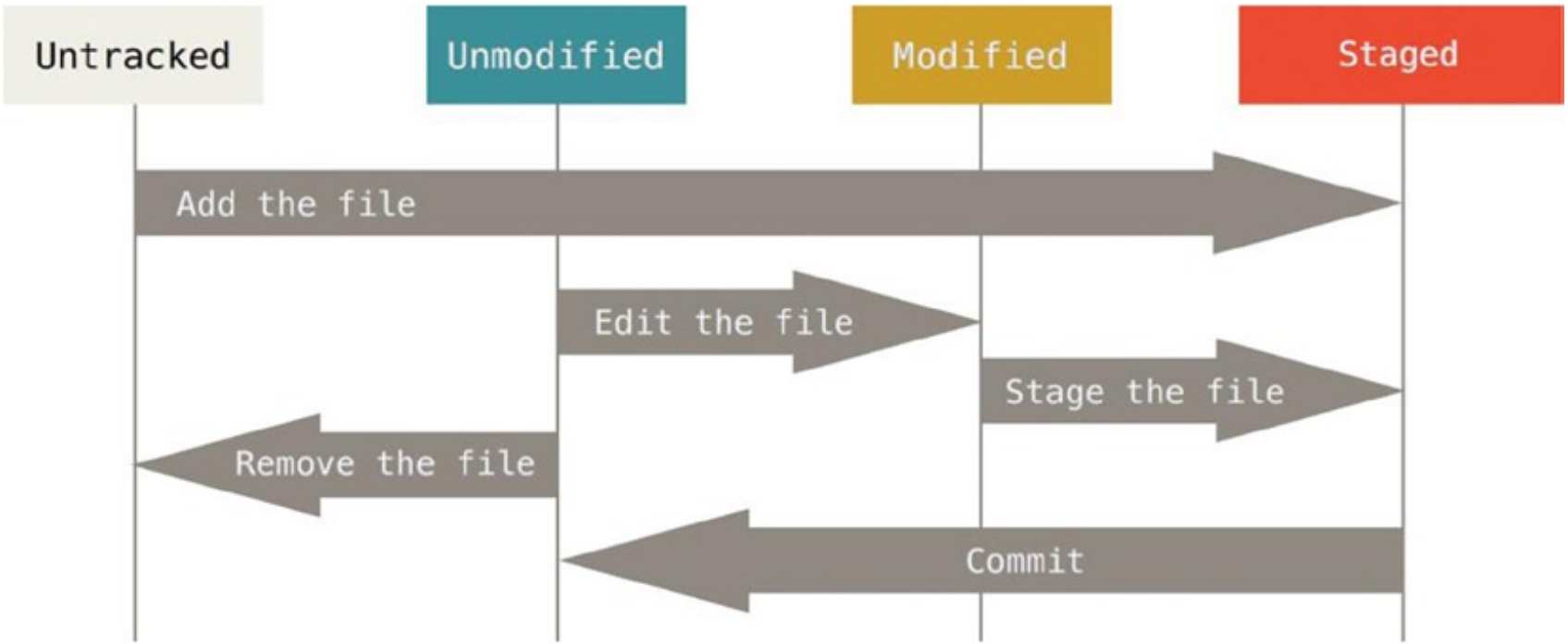
INSTALLATION

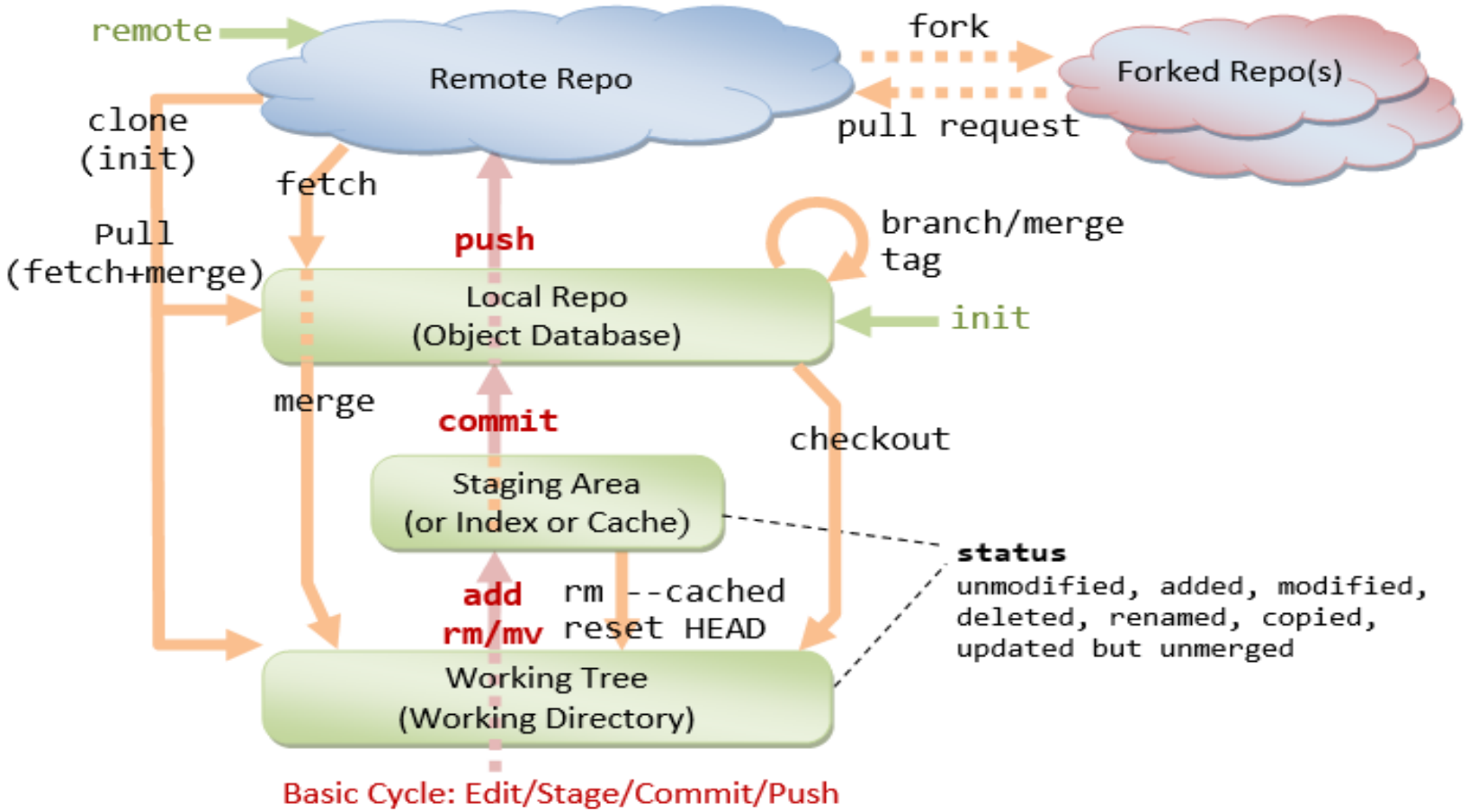
- **Fedora :** yum install git
- **Ubuntu :** sudo apt-get install git
- **Windows :** Just go to http://git-scm.com/download/win
- Set your username and password.
  - $ git config –global user.name ”John Doe”
  - $ git config –global user.email johndoe@example.com

COMMANDS

- Glossary
- Branch
- Checkout
- Cherry-picking
- Clone
- Fetch
- Fork
- HEAD
- Index
- Master
- Merge
- Origin
- Pull/Pull Request
- Push
- Remote
- Repository
- Stash

THE LIFECYCLE OF THE STATUS OF YOUR FILES




remote
Remote Repo
fork
Forked Repo(s)
pull request
clone
(init)
fetch
Pull
(fetch+merge)
push
branch/merge
tag
Local Repo
(Object Database)
init
merge
commit
checkout
Staging Area
(or Index or Cache)
status
unmodified, added, modified,
deleted, renamed, copied,
updated but unmerged
add
rm/mv
rm --cached
reset HEAD
Working Tree
(Working Directory)
Basic Cycle: Edit/Stage/Commit/Push

COMMANDS

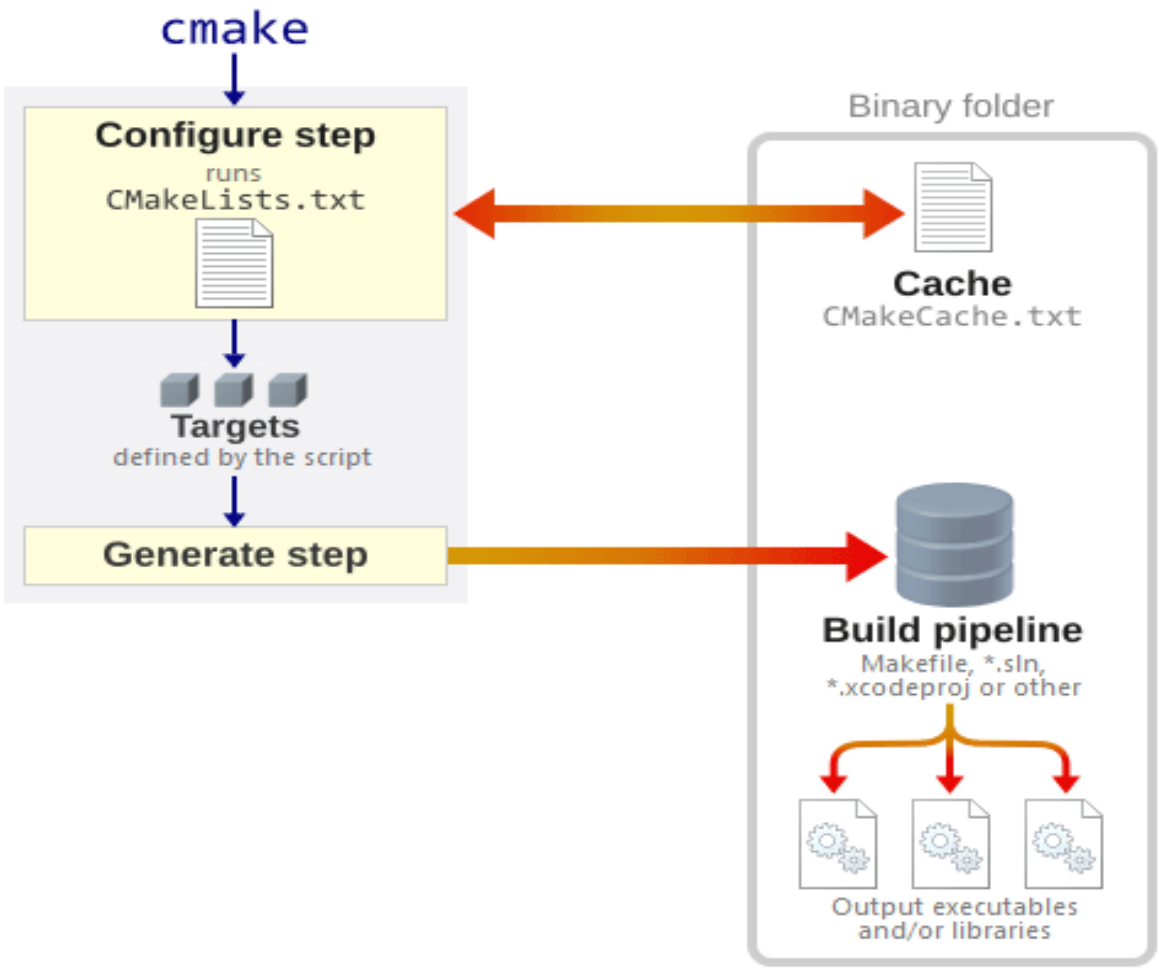
git fetch : The git fetch command communicates with a remote repository and fetches down all the information that is in that repository that is not in your current one and stores it in your local database.
git pull: The git pull command is basically a combination of the git fetch and git merge commands, where Git will fetch from the remote you specify and then immediately try to merge it into the branch you?re on.
git push: The git push command is used to communicate with another repository, calculate what your local database has that the remote one does not, and then pushes the difference into the other repository.

BUILD-SYSTEM GENERATOR

- Provides single-sourcing for build systems
- Knowledge of many platforms and tools
- Users configure builds through a GUI

Source and Build Trees

- The Source Tree contains:
  - CMake input files (CMakeLists.txt)
  - Program source files (hello.cxx)
- The Binary Tree (build tree) contains:
  - Native build system files (hello.dsp)
  - Program libraries and executables (hello.exe)
- Source and Binary trees may be:
  - In the same directory (in-source build)
  - In different directories (out-of-source build)


cmake
Configure step
runs
CMakeLists.txt
Targets
defined by the script
Generate step
Binary folder
Cache
CMakeCache.txt
Build pipeline
Makefile, *.sln,
*.xcodeproj or other
Output executables
and/or libraries

The CMake Cache

- Represents build configuration
- Populated by CMake code
- Stored in CMakeCache.txt at top of build
- Entries have a type to help the GUI
- Holds global information for CMake code
- Updated by CMake configuration phase

CMAKE FILES IN PARMOON

/ParMooN/ParMooN/CMakeLists.txt
/ParMooN/ParMooN/UserConfig.cmake

DIFFERENT USES OF CMAKE IN PARMOON

- Include main program
- Mention the output directory
- Select the architecture type
- Select the parallel type
- Select the Operating system