

Version Control Systems

The screenshot shows the SourceForge homepage. At the top, there's a navigation bar with 'sourceforge' logo, a search bar, and links for 'Browse', 'Enterprise', 'Blog', 'Help', 'Jobs', 'Log In', and 'Join'. Below this is a secondary bar with 'SOLUTION CENTERS', 'Go Parallel', 'Smarter IT', 'Resources', and 'Newsletters'. The main banner features the text 'Find, Create, and Publish Open Source software for free' and a search bar with the text 'Search from thousands of software titles'. Below the banner, there are statistics: 'TODAY: 4,343,789 DOWNLOADS', '34,360 CODE COMMITS', '3,579 FORUM POSTS', '2,736 BUGS TRACKED', and a link to 'MORE DETAILS'. On the left, there's a sidebar with categories: 'Audio & Video', 'Business & Enterprise', 'Communications', 'Development', 'Home & Education', 'Games', 'Graphics', 'Science & Engineering', and 'Security & Utilities'. The main content area is titled 'Projects Of The Month' and features two projects: 'Win32 Disk Imager' (Staff Choice) and 'Universal Media Server' (Community Choice), each with a 'Download' button and a brief description.

- Source Forge, Google Code (shelved), Github

The screenshot shows the GitHub repository page for 'openlink / virtuoso-opensource'. The repository is public and has 10,000+ commits, 7 branches, 38 releases, and 6 contributors. The main content area shows the commit history, with the latest commit being 'Merge branch 'develop6' into develop7' by 'VOS Maintainer' 2 days ago. The commit message is 'Merge branch 'develop6' into develop7'. The commit details show that the commit includes changes to 'appsrc', 'bin', and 'binsrc'. The right sidebar shows links to 'Code', 'Issues', 'Pull Requests', 'Wiki', 'Pulse', 'Graphs', and 'Network'.

Version Control Systems (cont.)

Version Control Systems

- A software tool that can store source code @ a central location
- Keeps a record of changes that have been made from time to time
- Keeps a record of who did what changes from time to time
- Can be used as a backup if something goes wrong in your code.

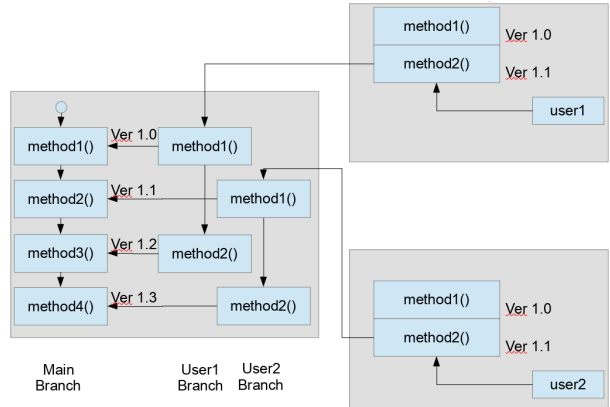
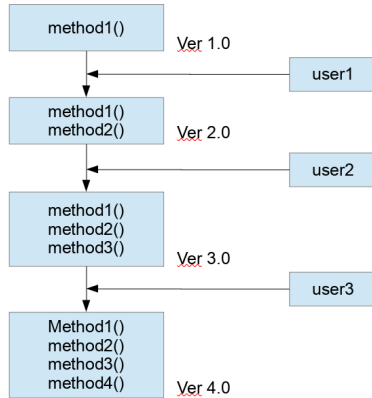
Individual Code Developer

- Can use version control as a backup tool
- Knows that the recent version is the live code
- Can maintain branches, e.g. developer branch, like branch, test branch, etc. etc. (Branches also known as Trunks)

Team Developers

- Each developer on a separate branch
- Developers can be in different parts of the world
- *upstream* can merge branches of two developers to do a version increment.

Centralized vs Distributed Version Control Systems



Centralized vs Distributed Version Control Systems (cont.)

Popular Version Control Systems

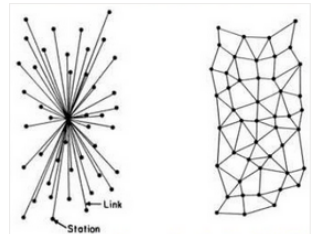
- SVN; Sub-version (Centralized)
- GIT (Distributed)
- Bazaar
- Mercurial
- ...

SVN (Centralized)

- Corruption risk. Will effect everybody
- Won't work on low network speeds
- Not scalable if users are too much
- Stores Complete Files

Git (Distributed)

- Corruption will only affect a local server
- Support for Load Balancing.
- Stores Partial Files.



Partial Storage

• Version 1.0

```
- public class test {
    public static void main(String[] args) {
    }
}
```

Stores This

• Version 2.0

```
- public class test {
    public test() {
    }
    public static void main(String[] args) {
        new test();
    }
}
```

Stores This

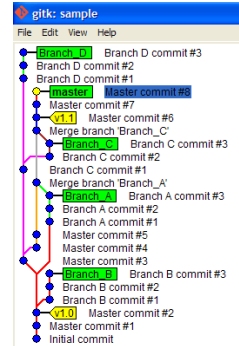
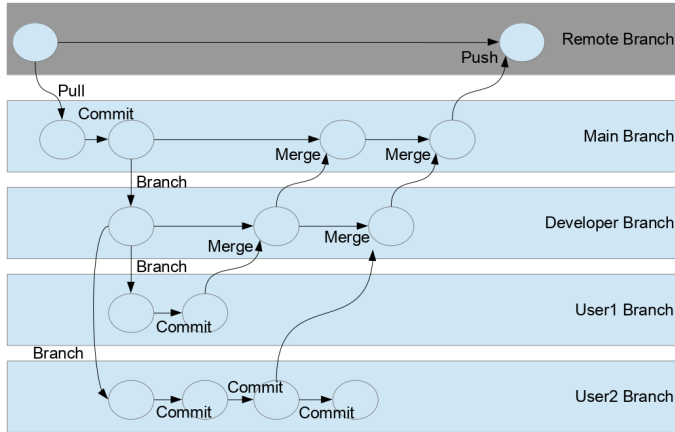
and This

Version 1.0

Version 2.0

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <assert.h>
5 #include <math.h>
6
7 #ifndef TOPEFFT_H
8 #define TOPEFFT_H
9 #include "topefft.h"
10 #endif
11
12 void tope3Dcheck( struct topeFFT *f, struct topePlan3D *t)
13 {
14     int type = 1; // Do not Change
15     f->error = c1SetKernelArg(t->kernelX, 7, sizeof(int), (void*)&type);
16     CHECKERROR
17     f->error = c1SetKernelArg(t->kernel_swap, 7, sizeof(int), (void*)&type);
18     CHECKERROR
19
20     /* Run Swapper */
21     t->globalSize[0] = t->nx;
22     t->globalSize[1] = t->ny;
23     t->globalSize[2] = t->nz;
24     t->localSize[0] = t->nx < 128 ? t->nx/2 : 128;
25     t->localSize[1] = 1;
26     t->localSize[2] = 1;
27     f->error = c1QueueNDRangeKernel(f->command_queue, t->kernel_swap, t->ndim, NULL,
28                                     t->globalSize, t->localSize, 0,
29                                     NULL, 64->event);
30     CHECKERROR
31     c1Finish(f->command_queue);
32     t->totalPTimeKernel += profileThis(f->event);
33 }
```

Git Workflow



Git Workflow (cont.)

How to Work with these Workflows?

- Built-into Eclipse (no need to install anything else)
- Built-into Netbeans (no need to install anything else)
- Via Commands on the Command Line
- Via Clients: GitHub (browser), Gitlab (Browser), SourceTree, Git-Cola, Git-Eye, SmartGit, GitG (Linux), Gigggle, ...
- Hosting your own Git: gitolite