Protocol of an editing session with git

Step 1: prepare a local copy of the repository

```
git clone https://bitbucket.org/akohlmey/ljmd-c.git ljmd-c
cd ljmd-c

[akohlmey@zero ljmd-c]$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master

[akohlmey@zero ljmd-c]$ git remote -v
origin     https://bitbucket.org/akohlmey/ljmd-c.git (fetch)
origin     https://bitbucket.org/akohlmey/ljmd-c.git (push)
```

This sets up a local "master" branch. Note that there are two copies of the master branch in your local repository. "master" and "origin/master". "master" is the local version while "origin/master" is the status of the remote repository when it was last imported; more on this later. You can have multiple local clones of the remote repository on either the same or different machines. If you have write access to the remote repository, you can be synchronize them via **`git pull`** and **`git push`**. Without write access, you can only import from remote (pull).

Step 2: start branches for development

All work should be done on branches. Git can handle a large number of branches very efficiently and switching between them is fast and easy, too. It is highly recommended to have one branch for each subtask and perhaps a "subtask-master" to collect multiple related subtasks until they are ready to be merged into the master branch (often also called "trunk"). If you are following a read-only repository, it may be better to have master only be a mirror of the remote master and use a different local branch for development.

Create a branch called “add-version” with:

```
git branch add-version
```

and create a second branch called “add-timer” with:

```
git branch add-timer
```

We can see those branches now with (also try: git branch -v):

```
[akohlmey@zero ljmd-c]$ git branch
  add-timer
  add-version
* master
```

Now switch to the "add-version" branch and make the ljmd program output a version string.

```
[akohlmey@zero ljmd-c]$ git checkout add-version
Switched to branch 'add-version'
[akohlmey@zero ljmd-c]$ git branch
  add-timer
* add-version
  master
```

Now edit the source in src/ljmd.c. After the changes are made, you can check the status:

```
[akohlmey@zero ljmd-c]$ git status
On branch add-version
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working dir)

      modified:   src/ljmd.c

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

the changes can be viewed with git diff

```
[akohlmey@zero ljmd-c]$ git diff
diff --git a/src/ljmd.c b/src/ljmd.c
index 1e89d38..f0b21b1 100644
--- a/src/ljmd.c
+++ b/src/ljmd.c
@@ -11,6 +11,8 @@
 #include <stdlib.h>
 #include <math.h>

+#define LJMD_VERSION "ljmd-c AK v0.1"
+
 /* generic file- or pathname buffer length */
 #define BLEN 200

@@ -177,6 +179,8 @@ int main(int argc, char **argv)
     FILE *fp,*traj,*erg;
```

```
     mdsys_t sys;

+    puts(LJMD_VERSION);
+
     /* read input file */
     if(get_a_line(stdin,line)) return 1;
     sys.natoms=atoi(line);
```

now schedule the changes for committing them to the repository with:

```
git add src/ljmd.c

[akohlmey@zero ljmd-c]$ git status
# On branch add-version
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#    modified:   src/ljmd.c
#
```

at this point `git diff` will not show any changes anymore. Now test the change and if it is working correctly commit it to the branch. Otherwise make more changes, monitor with `git status`/`diff` and schedule for addition with `git add` until it works as expected.

When doing a commit, remember to formulate a meaningful commit message, so you can later more easily reconstruct what has been done without having to look at the detailed changes.

```
git commit

add output of version string when starting a run
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch add-version
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#    modified:   src/ljmd.c
#

[akohlmey@zero ljmd-c]$ git status
# On branch add-version
nothing to commit (working directory clean)

[akohlmey@zero ljmd-c]$ git log
[add-version d1d6766] add output of version string when starting a run
 1 file changed, 4 insertions(+)
```

```
[akohlmey@zero ljmd-c]$ git log
commit d1d67667d4517501e407a33354b49d07ca72020e
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:59:13 2016 -0500

    add output of version string when starting a run

commit 1923119a5d03fa527bb02a9ffe68de093877ffd8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:21:06 2016 -0500

    import unoptimized serial C version of LJMD code
```

Step 3: concurrent development

Before we merge this change to the master branch (trunk) let us practice working on a second branch first. We now switch to the “add-timer” branch.

```
[akohlmey@zero ljmd-c]$ git checkout add-timer
Switched to branch 'add-timer'
```

We can switch back and forth between branches as we like. git will automatically update the version of the sources in the checked out tree accordingly.
Now we add code to print out timer information for startup and simulation time. This adds changes to several places of the code:

```
[akohlmey@zero ljmd-c]$ git diff
diff --git a/src/ljmd.c b/src/ljmd.c
index 1e89d38..876e46a 100644
--- a/src/ljmd.c
+++ b/src/ljmd.c
@@ -10,6 +10,7 @@
 #include <ctype.h>
 #include <stdlib.h>
 #include <math.h>
+#include <sys/time.h>

 /* generic file- or pathname buffer length */
 #define BLEN 200
@@ -58,7 +59,16 @@ static int get_a_line(FILE *fp, char *buf)
     }
     return 0;
 }
-
+
```

```
+/* helper function: get current time in seconds since epoch */
+static double wallclock(void)
+{
+    struct timeval t;
+
+    gettimeofday(&t,0);
+    return ((double) t.tv_sec) + 1.0e-6*((double) t.tv_usec);
+}
+
 /* helper function: zero out an array */
 static void azzero(double *d, const int n)
 {
@@ -176,7 +186,9 @@ int main(int argc, char **argv)
     char restfile[BLEN], trajfile[BLEN], ergfile[BLEN], line[BLEN];
     FILE *fp,*traj,*erg;
     mdsys_t sys;
+    double t_start;

+    t_start = wallclock();
     /* read input file */
     if(get_a_line(stdin,line)) return 1;
     sys.natoms=atoi(line);
@@ -237,10 +249,14 @@ int main(int argc, char **argv)
     erg=fopen(ergfile,"w");
     traj=fopen(trajfile,"w");

+    printf("Startup time: %10.3fs\n", wallclock()-t_start);
     printf("Starting simulation with %d atoms for %d
steps.\n",sys.natoms, sys.nsteps);
     printf("     NFI            TEMP            EKIN
EPOT              ETOT\n");
     output(&sys, erg, traj);

+    /* reset timer */
+    t_start = wallclock();
+
     /**************************************************/
     /* main MD loop */
     for(sys.nfi=1; sys.nfi <= sys.nsteps; ++sys.nfi) {
@@ -256,7 +272,7 @@ int main(int argc, char **argv)
     /**************************************************/

     /* clean up: close files, free memory */
-    printf("Simulation Done.\n");
```

```
+    printf("Simulation Done. Run time: %10.3fs\n", wallclock()-
t_start);
     fclose(erg);
     fclose(traj);
```
after adding and commiting the change, we see in the log file that this branch does not contain the previous change.

```
[akohlmey@zero ljmd-c]$ git log
commit 847cd5021d7daa4a62e3c4e61726c60ecb8ccde8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:09:20 2016 -0500

    add helper function to measure startup and run time

commit 1923119a5d03fa527bb02a9ffe68de093877ffd8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:21:06 2016 -0500

    import unoptimized serial C version of LJMD code
```

Let us assume that both changes are successful.

Step 4: merging branches

Since both changes on the feature branches are complete, we now want to merge them into the trunk and then delete the feature branches. First we switch to the branch "master", which is the pristine copy of the repo.

```
[akohlmey@zero ljmd-c]$ git checkout master
Switched to branch 'master'

[akohlmey@zero ljmd-c]$ git log
commit 1923119a5d03fa527bb02a9ffe68de093877ffd8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:21:06 2016 -0500

    import unoptimized serial C version of LJMD code
```

With git merge we merge the first feature branch into the trunk.

```
[akohlmey@zero ljmd-c]$ git merge add-version
Updating 1923119..d1d6766
Fast-forward
 src/ljmd.c | 4 ++++
 1 file changed, 4 insertions(+)
```

```
[akohlmey@zero ljmd-c]$ git log
commit d1d67667d4517501e407a33354b49d07ca72020e
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:59:13 2016 -0500

    add output of version string when starting a run

commit 1923119a5d03fa527bb02a9ffe68de093877ffd8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:21:06 2016 -0500

    import unoptimized serial C version of LJMD code
```

So far, so good. The change set from the feature branch is now included in our local copy of the trunk. Now we can merge in the second feature branch “add-timer”:

```
[akohlmey@zero ljmd-c]$ git merge add-timer
Auto-merging src/ljmd.c
CONFLICT (content): Merge conflict in src/ljmd.c
Automatic merge failed; fix conflicts and then commit the result.
```

This time there is a problem, that the git software cannot resolve automatically. There is an overlap of the changes between the two branches (both modified the same regsion) and we need to decide what the final code will have to look like. To ease the process, git has already included both changes into the source and and encoded where they collide it in a special format:

```
<<<<<<< HEAD
     puts(LJMD_VERSION);

=======
     t_start = wallclock();
>>>>>>> add-timer
```

the resolution in this case is to keep both changes and remove the markers:

```
     t_start = wallclock();
     puts(LJMD_VERSION);
```

to indicate that we resolved the conflict, we use "**`git add`**" to confirm the change we did manually, add it to the list of changes to be committed from the merge and then run "**`git commit`**". The automatically generated commit message already contains some information about the conflict.

```
Merge branch 'add-timer'
```

```
Conflicts:
      src/ljmd.c
```

edit as needed and complete the commit. The final log message will now be something like:

```
commit 9925f2248b593a8b1275f0990e2fbf90c1c94f9b
Merge: d1d6766 847cd50
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:14:38 2016 -0500

    Merge branch 'add-timer'
    Resolved Conflicts:
      src/ljmd.c

commit 847cd5021d7daa4a62e3c4e61726c60ecb8ccde8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:09:20 2016 -0500

    add helper function to measure startup and run time

commit d1d67667d4517501e407a33354b49d07ca72020e
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:59:13 2016 -0500

    add output of version string when starting a run

commit 1923119a5d03fa527bb02a9ffe68de093877ffd8
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 15:21:06 2016 -0500

    import unoptimized serial C version of LJMD code
```

So now we have the commit messages (and their corresponding changes) included from both feature branches and an additional commit noting that we had resolved a conflict in the merge of the add-timer branch and were required to do some additional changes.

Both feature branches are no longer needed and can be deleted.

```
[akohlmey@zero ljmd-c]$ git branch -d add-timer
Deleted branch add-timer (was 847cd50).
[akohlmey@zero ljmd-c]$ git branch -d add-version
Deleted branch add-version (was d1d6766).
[akohlmey@zero ljmd-c]$ git branch
* master
```

Step 5: merging a pull request

Now we receive an e-mail that a remote developer has implemented an improvement and that the changes are available as branch in a remote repository. This is generally referred to as a "pull request". Web frontends like Github and Bitbucket allow to merge those mostly automatically, but we are going to do it manually. The same procedure can be applied to merge with developers in a team. As usually we first create a local feature branch for those changes.

```
git checkout -b merge-pull
```

The comand above includes the branch creation into the checkout due to the -b flag.

Now we need to pull the remote branch into our local branch. Since the remote branch has a different parent than our local branch (we did some additional development in between), this is not going to be a simple "fast forward", but the changes need to be extracted relative to the remote branch parent and merged into the local branch. Thus we get prompted for a commit message.

```
[akohlmey@zero ljmd-c]$ git pull \
  https://bitbucket.org/akohlmey/ljmd-c-pull.git improve-make
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From https://bitbucket.org/akohlmey/ljmd-c-pull
 * branch            improve-make -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 Makefile | 1 +
 1 file changed, 1 insertion(+)
```

Now you can test how those changes work with your local version. If there are problems, you can ask the remote developer to make additional changes and update his remote branch so that you can pull those changes into your merge-branch. If you are satisfied with the changes, you switch to your master branch and merge the local copy of the pull request. At this point it is desirable to have a commit message indicating the date of the merge and what the pull request was about, so we tell git to not do a "fast forward" (trivial) merge. Notice how the commit log contains the local commit messages and the commit history from the remote repository.

```
[akohlmey@zero ljmd-c]$ git log
commit 91ac4decf811cf07d227dee6e60cf8507222e704
Merge: 9925f22 d72c241
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:51:37 2016 -0500

    Merge branch 'merge-pull' which contains makefile updates.
```

```
commit d72c2414fd422b6800aa1c494d0b8d72dbd00090
Merge: 9925f22 791fc5d
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:44:49 2016 -0500

    Merge branch 'improve-make' of
https://bitbucket.org/akohlmey/ljmd-c-pull into merge-pull

commit 791fc5d76f4e152c7b0306de761d29f6962ac8b7
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:35:21 2016 -0500

    make clean also deletes trajectory and simulation data files

commit 9925f2248b593a8b1275f0990e2fbf90c1c94f9b
Merge: d1d6766 847cd50
Author: Axel Kohlmeyer <akohlmey@gmail.com>
Date:   Sun Jan 31 16:14:38 2016 -0500

    Merge branch 'add-timer'

    Resolved Conflicts:
        src/ljmd.c
```

Notice how git status keeps you informed about how much your local branch has progressed since it was cloned from the master repository.

```
[akohlmey@zero ljmd-c]$ git status
On branch master
Your branch is ahead of 'origin/master' by 6 commits.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
```

Step 6: GUI time

various of the steps can also be handled by a GUI.

```
gitk
```

allows to browse the git repository and view the history and relation of change sets

```
git gui
```

is a frontend for many standard tasks like branching, merging, commits, push fetch and so on.