

Week 10 Lecture 1

Software Configuration Management

Software Configuration Management

Software Configuration Management

- SCM keeps versions of software
- Versions are important because people makes mistakes
 - You delete a file you didn't mean to
 - You change the program to where it doesn't work
 - ...
- It is helpful to go back to an earlier state

SCM in Production

- Production programs are written by teams
- SCM is even more important in team situations
- The person who broke the code can best fix it
 - But who broke the code
 - Only SCM knows for sure.
- SCM ensures that you always have working code.

Vocabulary

- *File*: object under control
- *Version*: a particular instance of a file that has been checked in
- *Repository*: a collection of files and their versions.
- *Check in*: putting an instance of a file in a repository.
- *Check out*: retrieve a version of a file
- *Branch*: a sequence of version

SCM Programs

- Capabilities
 - Add item to track
 - Commit item to a backup
 - Retrieve an item from backup
 - Create branch
 - Merge branches

SCM Programs

- RCS
 - Allows people to keep versions of files
- CVS, Subversion
 - Allows people to track versions of projects
 - CVS is the original and commonly used but is not currently being developed.
 - Subversion is currently most popular
 - Stores changes
- Git
 - Designed to support Linux development
 - Stores complete files
 - Allows multiple clones of repositories

Git

- We will use Git in this class
 - It is currently very popular
 - Its distributed capabilities are a little better than Subversion
 - Disk space is cheap
- There is a hosting site called `github.com`
 - Provide free hosting for open source
 - Source for much open source software

To use Git

- Set up your name and email in Git
 - Lets others know who you are

```
student> git config --global user.name "Student"  
student> git config --global user.email "student@mru.edu.in"
```

- The values are stored in .gitconfig

```
[user]  
    email = student@mru.edu.in  
    name = Student  
../.gitconfig (END)
```

Set up a Git repository

- Got to the directory and initialize the git repository

```
student> pwd
/home/student/project1
student> git init
Initialized empty Git repository in /home/student/project1/.git/
student> ls -la
total 20
drwxrwxr-x 3 student student 4096 Jul  8 16:36 .
drwxr-xr-x 6 student student 4096 Jul  8 16:29 ..
drwxrwxr-x 7 student student 4096 Jul  8 16:36 .git
-rw-rw-r-- 1 student student  94 Jul  8 16:16 hello.c
-rw-rw-r-- 1 student student 117 Jul  8 16:16 Makefile
```

Add the files and commit them

- Add the files to the repository using `git add`
 - This does not store anything; it tells git that you want to include these files when you do store.
- Commit the files using `git commit`
 - This creates a new version and stores the files in it.

```
student> git add hello.c
student> git add Makefile
student> git commit -m "Initial commit"
[master (root-commit) f0962cb] Initial commit
2 files changed, 16 insertions(+)
create mode 100644 Makefile
create mode 100644 hello.c
```

Commit Message

- It is hard to remember what each version contains
 - When you commit, an editor opens up so you can enter a message to help you find the right version
 - The more detailed your message, the easier it is to find the version if you need it

Retrieving a file

- You can get a list of the version using `git log`
- To get a file back, use `git checkout`

```
student> ls
hello.c  Makefile
student> rm hello.c
student> git log
commit bd1bb485ca54dfb9cbd23f97d040b9e29e0aa55b
Author: Student <student@mru.edu.in>
Date:   Wed Jul 8 16:38:18 2015 -0400

    Initial Commit
    - This directory contains the files for hello world
student> ls
Makefile
student> git checkout -- hello.c
student> ls
hello.c  Makefile
```

List Revisions: `git log`

- List revisions from most recent to oldest.
- Page though like **less**
 - `<enter>`: down one line
 - `<space>`: down one page
 - `'b'`: back one page
 - `'q'`: quit
- Refer to revisions by SHA-1
 - Long number following “commit”
 - Need only first four or so characters
- Easier to find stuff: `git log --oneline`

Example: git log

```
student@wren: ~/sp.20150718.experimental
commit 82de3d2fffcf9157b299ee6565472b274571e009
Author: Student <student@mru.edu.in>
Date:   Wed Jul 8 18:35:39 2015 -0400

    updated early projects to add a new week.

commit 897f4521b8d46cc2771daccf1e0acdd6a0e803bc
Author: Nathaniel Martin (Student) <ngm1955@gmail.com>
Date:   Mon Jul 6 08:51:52 2015 -0400

    Finished Week 6 lectures started Week 7

commit 0a5e2a4969bc9b243cece71e7e4f1087373a4764
Author: Nathaniel Martin (Student) <ngm1955@gmail.com>
Date:   Fri Jul 3 08:06:36 2015 -0400

    Finished Week 6 Lesson 1

commit 15a16b55c59d897a6fad2713ff7e9068a0dcb326
Author: Nathaniel Martin (Student) <ngm1955@gmail.com>
Date:   Wed Jul 1 11:22:11 2015 -0400

    Finished up lesson. Calc 6 enters an array and finds the median of it.
:
```

git log –oneline

```
student@wren: ~/sp.20150718.experimental
82de3d2 updated early projects to add a new week.
897f452 Finished Week 6 lectures started Week 7
0a5e2a4 Finished Week 6 Lesson 1
15a16b5 Finished up lesson. Calc 6 enters an array and finds the median of it.
e91311e Story 15: As a user, I want to be able to use the program calculate a square
      root, so I can use the calculator to calculate square roots.
d5e9b4e Storv 18: Enter 'sqrt' at the prompt and you are prompted for an integer
      . The calculator prints the square root of that integer.
931308d Finished Story 14: Prime test
1afd523 Reorganized to meet the backlog. Finished story 16: As a user, I want to
      be able to use the program to calculate the greatest common divisor (GCD) of
      two numbers, so I can use the calculator for GCD.
743f6e7 Finished story 15: get int. Mostly done in the previous iterations.
d7fbfee Finished Story14 display binary numbers.
cc20406 Completed Story 13: As a user, I want to be able to enter a int operand,
      so my calculator can do integer arithmetic.
171e702 Finished factorial, almost. The backlog is ambiguous.
2a1fd4e Finished Story 9: test for valid command.
615f9a9 Finished Story 8: division
7bb4078 Added Multiplication. Story 7
220f6b4 Finished up subtraction. Story 7
47d5e54 Finished Story 4: As a user, I want to be able to enter a floating point
      operand, so my calculator can do real arithmetic. Stored in cal
:
```


List Revision Contents: `git show`

- Shows revision message then differences between it and previous revision
- Page through as with `git log`.

Example: git show 82de

```
student@wren: ~/sp.20150718.experimental
commit 82de3d2fffcf9157b299ee6565472b274571e009
Author: Student <student@mru.edu.in>
Date:   Wed Jul 8 18:35:39 2015 -0400

    updated early projects to add a new week.

diff --git a/week2/project1/Makefile b/week2/project1/Makefile
index bc8d07a..a7b4a69 100644
--- a/week2/project1/Makefile
+++ b/week2/project1/Makefile
@@ -7,4 +7,4 @@ hello.o: hello.c
     gcc -c hello.c

clean:
-     rm hello.o hello
+     rm -f *~ hello.o hello
diff --git a/week2/project1/data_and_new_blocs.c b/week2/project1/data_and_new_blocs.c
new file mode 100644
index 0000000..9b0fc64
--- /dev/null
+++ b/week2/project1/data_and_new_blocs.c
@@ -0,0 +1,22 @@
:
```

Retrieving: git checkout

- To get work back use
 - `git checkout <revision>`
 - All of the files will be the like they were when you check in `<revision>`
 - Move the file you want to revert to a new location
- To return them to their current state:
 - `git checkout master`

More Information

- Best for starting out:
 - <https://www.atlassian.com/git/tutorials/setting-up-a-repository>
- The official manual:
 - <http://git-scm.com/book/en/v2>
- Reference
 - <http://git-scm.com/docs>