Source Code Management With Subversion

Diego MENÉNDEZ
dmenendez@psu.edu

Graduate Student
World Campus – Software Engineering
Penn State University

Penn State MacAdmins Conference, 2012
Outline

1 Intro
   - Source Code Management Systems

2 Concepts
   - Revisions
   - Repository Design

3 Practice
   - Problem Statement
   - Svn Subcommands
Outline

1 Intro
   - Source Code Management Systems

2 Concepts
   - Revisions
   - Repository Design

3 Practice
   - Problem Statement
   - Svn Subcommands
Evolution of SCMs

- **Source Code Control System (SCCS)**
  - Marc J. Rochkind, Bell Labs, 1972.

- **Revision Control System (RCS)**
  - Walter F. Tichy, Purdue University, 1982.

- **Concurrent Versions System (CVS)**

- **Subversion**
  - B. Collins-Sussman, B.W. Fitzpatrick, C.M. Pilato

- **Git**
  - Linus B. Torvalds
Evolution of SCMs

- **Source Code Control System (SCCS)**
  - Marc J. Rochkind, Bell Labs, 1972.

- **Revision Control System (RCS)**
  - Walter F. Tichy, Purdue University, 1982.

- **Concurrent Versions System (CVS)**

- **Subversion**
  - B. Collins-Sussman, B.W. Fitzpatrick, C.M. Pilato

- **Git**
  - Linus B. Torvalds
Evolution of SCMs

- **Source Code Control System (SCCS)**
  - Marc J. Rochkind, Bell Labs, 1972.

- **Revision Control System (RCS)**
  - Walter F. Tichy, Purdue University, 1982.

- **Concurrent Versions System (CVS)**

- **Subversion**
  - B. Collins-Sussman, B.W. Fitzpatrick, C.M. Pilato

- **Git**
  - Linus B. Torvalds
Evolution of SCMs

- **Source Code Control System (SCCS)**
  - Marc J. Rochkind, Bell Labs, 1972.

- **Revision Control System (RCS)**
  - Walter F. Tichy, Purdue University, 1982.

- **Concurrent Versions System (CVS)**

- **Subversion**
  - B. Collins-Sussman, B.W. Fitzpatrick, C.M. Pilato

- **Git**
  - Linus B. Torvalds
Evolution of SCMs

- **Source Code Control System (SCCS)**
  - Marc J. Rochkind, Bell Labs, 1972.
- **Revision Control System (RCS)**
  - Walter F. Tichy, Purdue University, 1982.
- **Concurrent Versions System (CVS)**
- **Subversion**
  - B. Collins-Sussman, B.W. Fitzpatrick, C.M. Pilato
- **Git**
  - Linus B. Torvalds
Outline

1 Intro
   - Source Code Management Systems

2 Concepts
   - Revisions
   - Repository Design

3 Practice
   - Problem Statement
   - Svn Subcommands
Concepts

- Working Copy
- HEAD
- BASE
Concepts

- Working Copy
- HEAD
- BASE
Diego MENÉNDEZ dmenendez@psu.edu Subversion

---

Concepts

- Working Copy
- HEAD
- BASE
Outline

1. Intro
   - Source Code Management Systems

2. Concepts
   - Revisions
   - Repository Design

3. Practice
   - Problem Statement
   - Svn Subcommands
Top Level Directories

- Trunk
- Branches
- Tags
Top Level Directories

- Trunk
- Branches
- Tags
Top Level Directories

- Trunk
- Branches
- Tags
Other Concepts

- Properties
- Keyword Substitutions
Outline

1. Intro
   - Source Code Management Systems

2. Concepts
   - Revisions
   - Repository Design

3. Practice
   - Problem Statement
   - Svn Subcommands

Diego MENÉNDEZ dmenendez@psu.edu

Subversion
Data Transmission Project

- Description
  - UNIX Commands
  - Language + Libraries
  - Variables
Data Transmission Project

- Description
- UNIX Commands
  - Language + Libraries
  - Variables
Data Transmission Project

- Description
- UNIX Commands
- Language + Libraries
- Variables
Data Transmission Project

- Description
- UNIX Commands
- Language + Libraries
- Variables
Outline

1. Intro
   - Source Code Management Systems

2. Concepts
   - Revisions
   - Repository Design

3. Practice
   - Problem Statement
   - Svn Subcommands
We create the directory structure and edit the files.

$ mkdir sockets
$ cd sockets
$ mkdir trunk branches tags trunk/src
$ emacs trunk/src/server.c trunk/src/client.c

We do the initial import to the repository.

$ svn import sockets https://server.edu/sockets -m 'Initial import.'
**svn checkout (co)**

- We can check out the contents of the repository anywhere.

```
$ svn co https://server.edu:/sockets/
A  sockets/trunk
A  sockets/trunk/src
A  sockets/trunk/src/client.c
A  sockets/trunk/src/server.c
A  sockets/branches
A  sockets/tags
Checked out revision 1.
```
We add *sending* code to the client.
We add *receiving* code to the server.

$ cd sockets/trunk/src/
$ emacs client.c server.c

We confirm that the code was changed.

$ svn status
M     client.c
M     server.c
We want to see the changes.

```
$ svn diff client.c
Index: client.c
=================================================================
--- client.c (revision 1)
+++ client.c (working copy)
@@ -22,6 +22,12 @@
 freeaddrinfo(result);
 + // SEND
 + char *info = "65535";
 + printf("Sending...\n");
 + int bytes = write(sock_fd, info, strlen(info));
 + printf("Sent %d bytes: %s\n", bytes, info);
```
Other svn subcommands

- We can check a new version in.
  
  $ svn ci -m 'Added receiving and sending code.'

- See the history of changes.
  
  $ svn log

- Create a branch for maintenance.
  
  $ svn copy trunk/ branches/v1/
  $ svn ci -m 'Creating maintenance branch.'

- And tag the code for release.
  
  $ svn copy branches/v1/ tags/v1-0/
  $ svn ci -m 'Tagging version one for release.'
Thanks!

- import
- co
- status
- ci
- log
- diff
- copy
B. Collins-Sussman, B. Fitzpatrick, C. Pilato. 
*Version Control with Subversion.*
O’Reilly, 2008.

D. Child. 
*Subversion Cheat Sheet,*