

Source Code Control System in Unix

Prof. P.C.P. Bhatt



How Does Versioning Help?

- A facility which we need often is “undo” to rollback. *Rollback* may be needed and can be achieved at any stage of development.
- From the *version of file* one may fork out to a newer file which is a better version.
- A software can cater to users with various levels of abilities and with a range of workplace environments. This can be achieved with very *little overhead*.



The SCCS

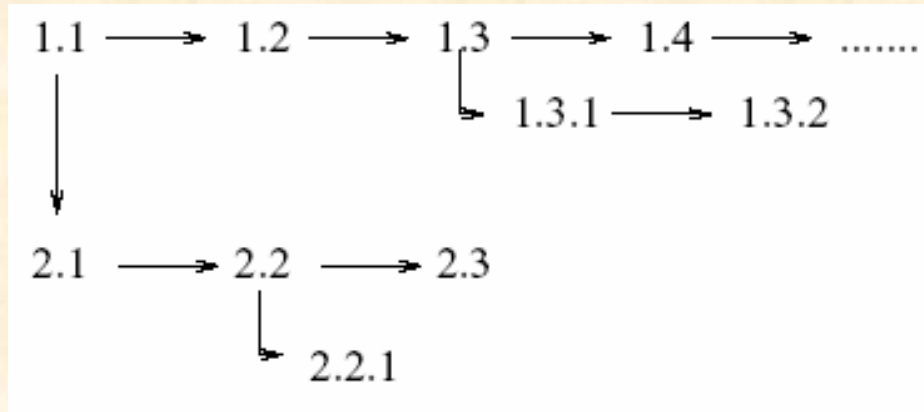
- During development period there will be modules in *various stages of progression*.
- Incremental development of system can be made using *stable* modules.
- Ability to *lock* each version for limited access can be provided.
- One should be able to chain various stages of locked versions in a hierarchy to reflect a particular *software evolution*.



How This is Achieved

- Version tree hierarchy is generated automatically and follows a *numbering scheme*.
- SCCS creates files in a *special format*.
- The *compilation* is handled in a special way.
- Edits to the files are handled in a special way.
- A suite of SCCS commands is provided to *manage versions* and create links in the version trees.

A Typical Version Tree under SCCS - 1



Unix obtains this capability by maintaining a version tree with the following support mechanisms.

- Version tree hierarchy is generated automatically and follows a numbering scheme as shown in above figure. The numbering helps users to identify the evolution of software from the version tree. Internally, it helps Unix in maintaining differences between related versions of files



A Typical Version Tree under SCCS - 2

- The SCCS creates files in a special format.
- The compilation is handled in a special way.
- Edits to the files are handled in a special way. In fact one can edit files at any level in the version tree.
- A suite of SCCS commands are provided to manage versions and create links in the version trees.



SCCS Facilities

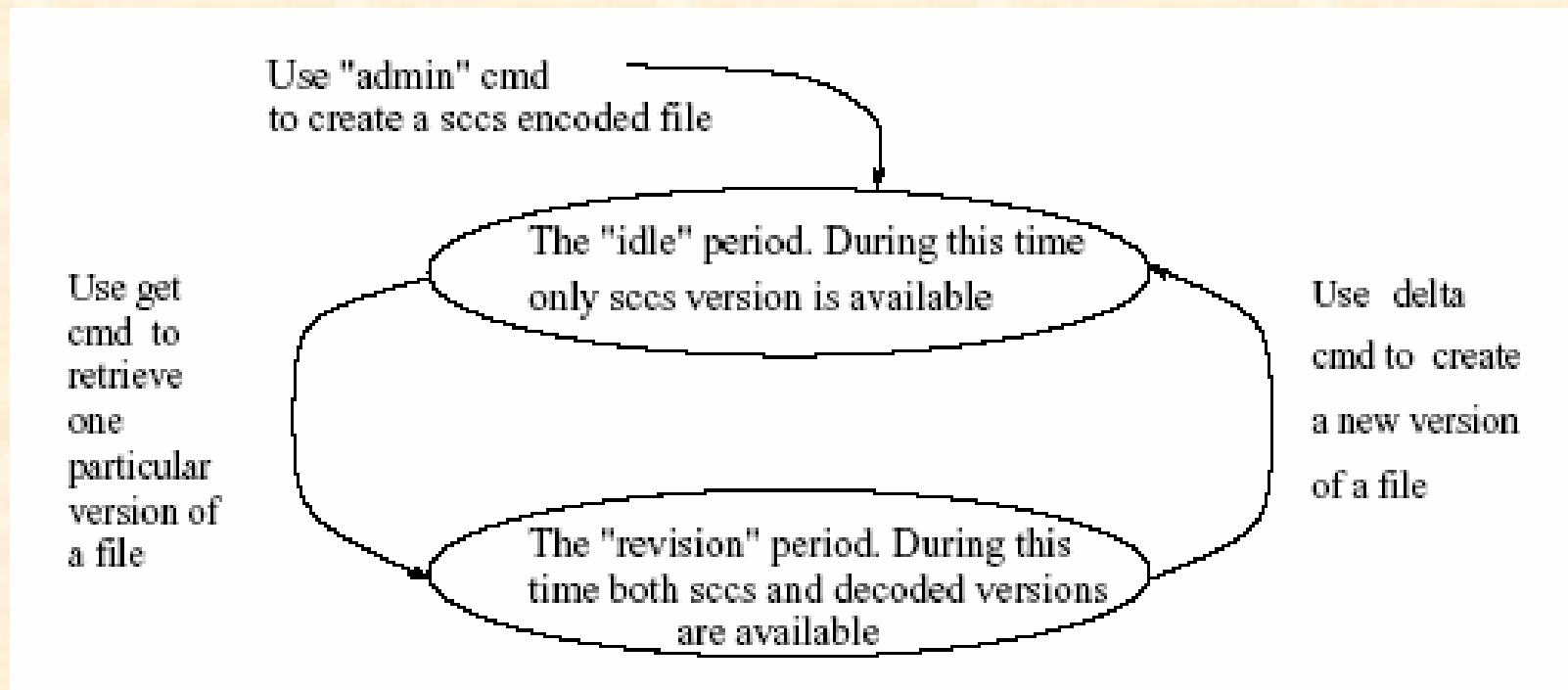
- Unix permits locking out of modules under development.
- For management of hierarchy, it is possible to create a tree structure amongst versions.
- Unix supports precedence amongst the versions of a module.



SCCS Command Structure

- Between two revisions a file under SCCS is considered to be *idle*.
- Between two adjacent versions, both idles, there is only a *small change* in the content.
- *Version numbers* can be generated automatically.
- We need to command use to indicate which file we retrieve and a command *delta* to indicate the change that needs to be incorporated.

The “idle” and “revision” Periods in SCCS





SCCS Command Options.

Option	Effect
-n	creates a new SCCS history file
-a	used to add users to a gid, this user can now check deltas
-e	used to erase (when used with admin) a user



An Example Session

get -e s.testfile

The `-e` option is to invoke “open for edit”. This command prints the version number and number of lines in `testfile`. If we perform a delta, we can create a newer version of `testfile`. This can be done simply as follows:

delta s.testfile

We may just add comments or actually make some changes. This can be now checked by using a visual editor again.

Unix provides a facility to view any file with a given version number. For example, to view and run a certain previous version 1.2, we may use the command shown below.

get -p -r1.2 s.testfile

The `-p` option is to invoke a path and `-r` is to invoke a run.