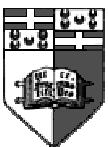




Revision Control Systems

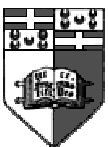
- Part of the software development is writing a piece of code and then adding things to it or fixing any bugs.
- Many times different people might be working on the same source code, for example one might be using it and another might be improving on it.
- Also it would be nice to maintain several version of a file's life history so that we can revert back to any version when something goes wrong.
- Using different filenames or a plethora of directories does not make managing working files any easier.
- We need a system that can keep track of revision automatically:**rcs**





RCS

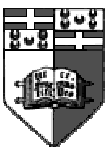
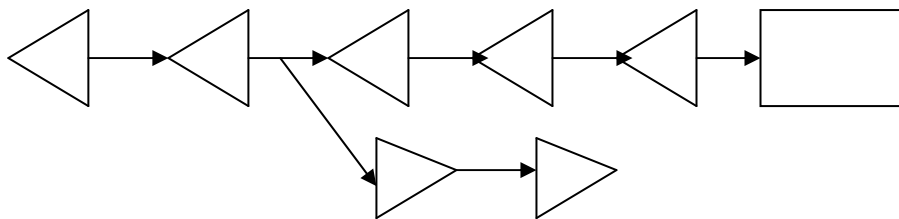
- RCS keeps a complete history of changes you make to a file.
- It can extract any version of the file upon request by keeping a complete history
- It can also manage access to a file from multiple users.
- Any file can be used with RCS including your C files, your latex file and any other file that might need changing from time to time.
- RCS does not store a copy of each revision. It takes very little space to store the changes.
- Every revision must be accompanied with a description by the writer about the changes.





Deltas

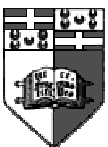
- To save space, RCS uses the fact that more of the text normally remains unchanged between one revision to the next.
- Using a technique similar to *diff*, it performs line comparison and marks revision changes by setting whole lines for deletion or addition.
- A whole set of operations to change one version into the next is called a delta.
- RCS maintains a copy of the latest revision and then keeps *reverse* deltas to backtrack to previous versions.
- In places where there are branches, *forward* deltas are used.





Checking In

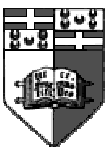
- First thing to do is create a directory called RCS.
- This directory could be a symbolic link to a common shared repository.
- After having a version of a file you want to store:
 - ci file
This checks in ‘file’ in the repository.
- A new file called *file,v* will be created in the RCS directory. The original file will also be removed.
- A description of the file will be requested from the user. This description is terminated with ^D or a period on an empty line.
- The first file checked in will be given revision number 1.1. Subsequent check ins will be assigned incrementing numbers, say 1.2, etc





Checking Out

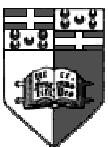
- When needing to access a file to view, one uses the `co` command.
 - `co file`
- This is used for compilation or viewing since it's a read only file.
- If one needs to edit the file, one has to obtain a lock on it using
 - `co -l file`
- Only one user can hold the lock on a file. And only the person holding the lock can check in the file again, thus releasing the lock again.
- Locks guarantee that no more than one user can be editing the program at any one time.
- If another user tries to break the lock (needed sometimes), the lock holder will be notified by e-mail.





Creating Specific Revisions

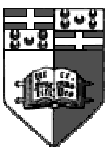
- One can start a new branch using the `ci` command.
 - `ci -r<no> file`
- These start a new revision numbering and any further check ins will follow these numbers.
- One can create new branches in the middle of the revision sequence. Ex:
 - `ci -r1.1.1 file`
Any new additions will follow this numbering: 1.1.2, 1.1.3, etc.
- New branches are needed for :
 - Temporary fixes
 - Distributed Development and Customer Modification
 - Parallel Development
 - Conflicting Updates





States and Names

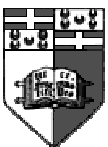
- Each revision has a specific state under which it is stored. The default state is Experimental (Exp).
- States are user defined but the most common ones are:
 - Exp :- experimental
 - Stab :- Stable
 - Rel :- Released
- One can specify the state of a specific revision using the rcs command:
 - rcs -sstate[:rel] file
- RCS allows you to group several revisions under a symbolic name using the -n option. Ex:
 - ci -n "TestName" file





Selective Checkout

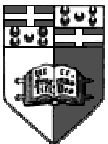
- When checking out a file, one can specify which revision one wishes to check out.
- A specific revision can be checked out using the `-r` option:
 - `co -r<no> file`
- One can `co` a file checked in by a specific user using the `-w` option:
 - `co -w<user> file`
- A cut-off date can be specified where a file closest and prior to this date will be checked out.
 - `co -d“January 13,12:00 PM” file`
- A symbolic name check out can be done using the `-r` option
 - `co -r“TestName”file`
- A filename can contain asterisks to check out (even check in) multiple files.





RCS Variables

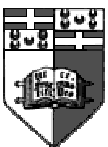
- RCS allows you to include special variables in your source code which is then automatically substituted on check out.
- These variables are:
 - `Id` : Full identification string
 - `$Revision$` : Revision No
 - `$Date$` : Date of revision
 - `Log` : Log entries of all revisions
- The `Log` variable when included in your C file can present a run down of what changes were done at what time.
- Expanded variables are not stored internally but only on checkout. The `Log` variable expands back to the revision where this variable started being included.





Extra RCS tools

- *ident* : If variables are included in a given file, this will extract them.
- *rcs* : This is used to set states, change revisions, steal locks, etc.
- *rcsclean* : Removes files that were checked out but have not changed. -u releases any locked files.
- *rcsdiff*: Compares different revisions
- *rcsfreeze*: Assigns a symbolic name to multiple files under the same revision number.
- *rcsmerge*: Merges two revisions together when possible. If uses a comparison of two in relation to a common ancestor.
- *rlog*: Prints all log messages in an rcs file





Exercises

- Read the supplied paper (Walter F. Tichy) especially issues on deltas(3) and locking(4).
- Look at your ,v file and try to figure out where the deltas are and what they do.
- Try checking in several revisions of a file and create branches of the revision tree.
- Try out some of the extra tools supplied with RCS.

