A Quick Introduction to RCS (the Revision Control System)

The Revision Control System (RCS) is a simple way to manage the storing, retrieval, logging, identification, and merging of different versions of files.

As an example, consider a program you are writing. At some point you "check in" a version of your program to RCS. Then you may continue working on your program, making many changes. Once you have a better version of your program you can "check in" that version, and RCS will store your new program, along with the differences between it and your old program. Suppose that later, in correcting an error, you accidentally made your program worse. You could "check out" one of your previous versions and start over on your revisions.

RCS can also be used for managing revisions of term papers, letters, or any kind of text or program file. When you check a file into RCS it is assigned a new version number, the date and time are recorded, and you are allowed to enter a short message describing your changes. It's easy to review this "revision log" to see what changes were made when, and by who. [It's possible to use RCS for programs or text files being worked on by more than one person, such as a programming team or research group.]

RCS stores the current version, along with the changes needed to reconstruct any previous version, in a "version" file. The name of the version file is the name of the working file with "`,v`" added to the end. Thus if you were working on the program in the file `mavassar13.f` the version file would have the name `mavassar13.f,v`. Normally RCS will keep the version file in the same directory as the working file, but to clean things up a little you can create a subdirectory called RCS, like so,

```
%  mkdir RCS
```

and RCS will automatically keep the version files in this subdirectory instead. (The "`%`" character represents the Unix command prompt, you do not type it.) You obviously only need to create this directory once.

Once you have a program or file you want to "check in" to RCS you need only give the "ci" command with the name of the file, like so:

```
%  ci mavassar13.f
```

RCS will tell you which version you are entering and ask you to enter your revision comments. These comments can go on for several lines; to end them press `^D` (Control-D). The first time you check a program in RCS will ask you to enter a short description of the file. This is not the revision comment, but it also shows up in the revision log. Here is an example:

```
% ci mavassar13.f
RCS/mavassar13.f,v  <--  mavassar13.f
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> Matthew Vassar's version of the heapsort program.
>> ⟨^D⟩
initial revision: 1.1
done
%
```

You will find that the working file, `mavassar13.f`, is no longer in your directory. You will have to "check out" the file to get a copy back in your directory. In order to make modifications to the file and check those back in you must also have a "lock" on the file. The command to check out a file, with a lock, is "`co -l`", as in this example:

```
% co -l mavassar13.f
RCS/mavassar13.f,v  -->  mavassar13.f
revision 1.1 (locked)
done
%
```

Now you will find the file in your directory. If you check the file out without the lock (by just saying "`co`") you will find that the file protection modes have been set to read-only. You can make the file writable with the Unix command '

```
%  chmod +w mavassar13.f
```

but you still won't be able to check the file back into RCS until you "lock" it with the command

```
%  rcs -l mavassar13.f
```

At any time while you are editing your file you can see how the current working file differs from what is kept in the RCS version file by using the "`rcsdiff`" command. For example:

```
% rcsdiff mavassar13.f
===================================================================
RCS file: RCS/mavassar13.f,v
retrieving revision 1.1
diff -r1.1 mavassar13.f
1c1
<         PROGRAM AVGVAR
```

```
---
>         PROGRAM MAVASSAR13
11c11
< C  Exercise 12 (Fortran), Spring 2003
---
> C  Exercise 13 (Fortran), Spring 2003
%
```

Only lines that differ between the two version are shown. The lines with the left arrow "<" are what is in revision 1.1 of the file, while the lines with the right arrow ">" are what is in the current working file. It is also possible to display the differences between two different versions that have already been checked into RCS.

After you have made your changes and want to save them you can check the file into RCS again. If you are going to check out a locked version afterwards anyway you can use the command "`ci -l`", which checks the file in, then checks it out again with a lock. For example:

```
% ci -l mavassar13.f
RCS/mavassar13.f,v  <--  mavassar13.f
new revision: 1.2; previous revision: 1.1
enter log message, terminated with single '.' or end of file:
>> Changed program name to MAVASSAR13.
>> Changed exercise number to 13.
>> ⟨^D⟩
done
%
```

If you try to check a file in but you don't have a lock on it the check-in will fail. You can get a lock on the file with the command "`rcs -l`", and then you will be able to check the file in.

RCS has a mechanism for incorporating the revision number, revision date, and other useful information, into comments in your program. In any character string of the form `$Revision: x.xx $` the "x.xx" is replaced by the version number when the file is checked out. Similarly `$Date: xxx $` is replaced by the date and time when the version was checked in. Thus it is a good idea to add a comment line containing these strings somewhere near the beginning of your program. For our example program we will add the line

```
C   $ Revision: 1.1 $  :  $ Date:  $ :  $ Author:  $
```

and then check this into RCS with "`ci -l`".

```
% ci -l mavassar13.f
RCS/mavassar13.f,v  <--  mavassar13.f
new revision: 1.3; previous revision: 1.2
enter log message, terminated with single '.' or end of file:
>> Added RCS revision number/date comment.
```

```
>> ⟨^D⟩
done
%
```

Now if we look at the beginning of the program we will see that the comment has been changed to reflect the new version number and date:

```
% more mavassar13.f
       PROGRAM MAVASSAR13
C=====================================================================*
C Exercise 13 -- sorting numbers with the Heapsort subroutine
C
C  Matthew Vassar   <mavassar@vassar.edu>
C  Department of Physics and Astronomy, Vassar College
C  Exercise 13 (Fortran Reading Course), Spring 2003
C  $ Revision: 1.3 $ : $ Date: 2005/04/16 19:57 $ : $ Author: mavassar $
C=====================================================================*
--More--(12%)
```

If you wish to review the revision log of a file you can simply use the "`rlog`" command. Saying "`rlog -h`" and the file name will show you only the header information from the version file. Saying simply "`rlog`" and the file name will show you the date and revision log entries for each revision. Since this can be a lot of information it is usually best to pipe this through the "`more`" command. For example:

```
% rlog mavassar13.f | more
RCS file: RCS/mavassar13.f,v
Working file: mavassar13.f
head: 1.3
branch:
locks: strict
       mavassar: 1.3
access list:
symbolic names:
comment leader: "c "
keyword substitution: kv
total revisions: 3;     selected revisions: 3
description:
Mathew Vassar's version of the heapsort program.
----------------------------
revision 1.3    locked by: mavassar;
date: 2003/04/16 19:57:20;  author: mavassar; state: Exp; lines: +7 -6
Added RCS revision number/ date comment.
----------------------------
revision 1.2
date: 2001/03/15 19:46:14;  author: mavassar; state: Exp; lines: +2 -2
Changed program name to MAVASSAR13.
Changed exercise number to 13.
----------------------------
```

```
revision 1.1
date: 2000/11/27 19:29:20;  author: mavassar;  state: Exp;
Initial revision
=============================================================================
%
```

For more information about RCS you can read the on-line man pages for "`rcsintro`", "`ci`", "`co`", "`rcsdiff`", "`rlog`" and "`rcs`". For more detailed information about RCS read see the section on RCS in "*Unix for Fortran Programmers*" by Mike Loukides (O'Reilly & Associates, Inc.), or read the original paper describing RCS, "A Revision Control System" by Walter F. Titchy.