

Source Control with RCS

David L. Levine
Christopher D. Gill
Department of Computer Science
Washington University, St. Louis
levine,cdgill@cs.wustl.edu

<http://classes.cec.wustl.edu/~cs342/>

Source Control

- Why use source control?
- **RCS** and its commands
- Example
- **RCS** with emacs
- For further information
- Summary

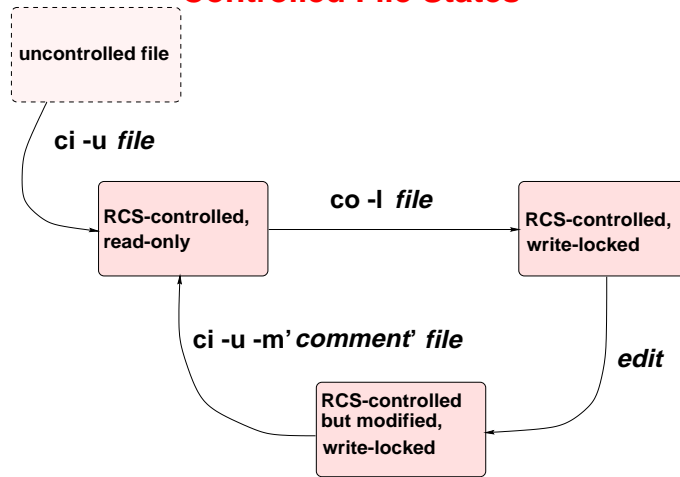
Why Use Source Control?

- Preserve as many revisions of a file as you like
 - With revision comments
- Backup
- Maintain multiple development branches
- Negligible overhead
 - There's no reason not to use source control!

RCS

- Revision Control System
- Developed by Walter Tichy in early 1980's
- Widely available
 - Source code is freely available

Controlled File States



RCS Commands

- Store a revision
 - `ci -u -m'comment' file [...]`
 - The same command (without the `-m'comment'` option) is used to initially put a file under RCS control
 - * NOTE: it's a good idea to create an RCS directory or symbolic link, before first putting a file under RCS control
- Retrieve latest revision
 - `co file [...]`
- Check out latest revision for modification (write)
 - `co -l file [...]`

RCS Commands (cont'd)

- View revision history
 - `rlog file [...]`
- Identify revision of a file
 - Keywords such as `Id`
- View differences between working revision and latest checked-in revision, or between any checked-in revisions
 - `rcsdiff [-rrevision] file [...]`
- Merge revisions
 - `rmerge`

RCS Example

```
$ mkdir RCS
```

```
$ ci -u rcs.tex
```

```
RCS/rcs.tex,v <-- rcs.tex
```

```
enter description, terminated with single '.' or end of fi
```

```
NOTE: This is NOT the log message!
```

```
>> RCS intro slides
```

```
>> .
```

```
initial revision: 1.1
```

```
done
```

```
$ ls -l rcs.tex
```

```
-r--r--r--  1 levine      4451 Feb 21 00:05 rcs.tex
```

RCS Example (cont'd)

```
$ co -l rcs.tex
RCS/rcs.tex,v --> rcs.tex
revision 1.1 (locked)
done

$ ls -l rcs.tex
-rw-r--r--  1 levine      4458 Feb 21 00:05 rcs.tex

$ head -1 rcs.tex
%%Id: rcs.tex,v 1.4 1999/10/04 16:55:17 levine Exp $
```

RCS with Emacs

- C-x v v
 - Or, Tools, VC, Check In/Out
- After initial check-in, emacs will automatically check in or out when read-only is toggled
 - C-x C-q

For Further Information

- man pages
 - General: `rcsintro`, `rcs`
 - Specific: `ci`, `co`, `rlog`, `rcsdiff`, `rcsmerge`

Summary

- **RCS** is invaluable for maintaining revisions.
- **RCS** is invaluable for backup.
- However, concurrent development is best done with CVS . . .
 - because there can only be one write lock of a file with RCS.