McMaster University
Department of Computing and Software
Dr. W. Kahl

SFWR ENG 3BB4 — Software Design III — Concurrent System Design

11 January 2007

This assignment is due on Thursday, January 25, at 9:30

The example programs of the textbook **"Beginning Linux Programming**" include the "CD database application" shell script of Chapter 2 (pp. 80ff.) in the book, file chapter02/app/cd_db.

(a) Keep a **log** (for example as a text file) documenting your activities for this assignment and their timing.

Document in this log which documentation you are consulting, for which commands or features, and, if applicable, which features you spend longer time in identifying or finding the right documentation for.

Document also your interaction with peers — **this assignment is to be solved individually!**

Include in the log for your own reference the solutions to problems you encountered.

Estimates on spent would be welcome.

(b) Understand cd_db in detail, in particular understand all shell constructs, shell built-ins, system utilities, and their options used there.

(c) Add to the title database a new field to accomodate an **optional cover art file name**.

Adapt cd_db so that:

- Users have an opportunity to enter, modify, and delete cover art file names together with other title information
- At least the presence of a cover art file name is indicated in the CD listing.
- At least one menu offers an option that will start a graphics viewer (for example display from the imagemagick package) on the selected cover art file.

(d) Replace the current interaction with appropriate use of dialog (or, if you prefer, Xdialog from http://xdialog.dyns.net/).

(e) Using find, add a feature that allows users to search for cover art files matching some selected pattern in some selected directories. Provide options to add each of the files found by the find invocation either to an existing CD entry, or to a new entry.

(f) **Bonus:** Add similar features for attaching, locating, and potentially playing track files. You could also use, for example, the utility mp3info to extract textual information from track files, or even to add it to track files.

Deliverables:

- Your **log** on paper.
- Up to two modified versions of cd_db — you may decide to implement for example the dialog version without the cover art features. For each version, submit:
  - **User documentation** for your modifications.

    This should be precise documentation (**specification style**) which features exactly you decided to add to the system, what different menu options mean, etc.

    This should normally not exceed one printed page.
  - A **well-documented source code printout**.

    Minimally, this would be the well-commented shell script.

    Much preferably this would be the document output of some **literate programming** system like Funnelweb, or noweb.[1]
  - ***Electronically:*** The shell script, the literate programming source (if applicable), and sample database files.
- All your files should be bundled together into a single .tar.gz file containing one top-level directory, and there at least a REDME file explaining the other files in the package.

Further technicalities of the **submission procedure** will be posted next week on the course page.

[1] You could also use the program typesetting capabilities of the TeX-like document formatting system Lout (available at ftp://ftp.it.usyd.edu.au/jeff/lout/). This is the literate programming mechanism I use in my lecture slides and for exercises, passing first the option pipe {tee *outfile* } and from then on pipe {tee -a *outfile* } to the programming language keyword — for shell programming, @Perl might work best among those currently offered.