



Assignment 3

Requirements

This exercise is designed to give you practice using the `GtkTextView` widget, among other things. Your program will open text files that contain special tags for formatting the text. It will find these tags and use them to display the text with the proper formatting. Some of the tags will be bookmarks. The program will display the list of bookmarked text items in a left side pane, and the text itself in the right side pane. Both panes will be scrollable.

The tags are bracketed, just like html, and will be one of:

Bold ` ... `

Italic `<i> ... </i>`

Monospace `<tt> ... </tt>`

Bookmark `<h> ... </h>`

The tags in the program will not overlap, to make your task easier. Thus, there cannot be a sequence such as `text<tt>texttext</tt>`. The formatting tags, ``, `<tt>`, and `<i>`, may nest, such as in `this is bold<i> and this is bold italic </i> but this is back to bold <tt> and now it is bold Monospace</tt>`. When the program reads the input file, it should find all bookmarked text and create a list of the bookmarks and display them in the left pane. For example, if the text contains

```
<h>Chapter 1</h> This is the start of it all. and so on
<h>Chapter 2</h> and the saga continued, much to the chagrin of ...
```

then the program will store the text “Chapter 1” and “Chapter 2” as separate lines in the left pane. The text itself will be displayed in the right pane, without the tags. In other words, the tags will be invisible text, or perhaps deleted text, depending upon how you choose to implement this. In place of the tags, the text will have the formatting applied to it in accordance with the meaning of the tags.

The program should have the following buttons:

Open opens a file chooser dialog to choose text files to open as input.

Close closes the currently open file, displaying just blank panes upon closing it.

Quit quits the application.

The buttons should be at the top and should be stock items, nonresizable. The main window title should change when a file is loaded, to the name “**Browser:**” followed by the absolute or relative pathname of the loaded file.

Extra Credit (30%)

Design the program so that the bookmarks in the left pane, when clicked, cause the text in the right pane to be scrolled so that the bookmarked text is at the top of the pane.



Hints and Suggestions

The program should make use of the Glib Library's Regex utilities. There are functions to find occurrences of regular expressions in input files, using the regular expression syntax of the UNIX utilities `grep`, `vi`, and such as well as Perl. It supports back-referencing and UTF-8 strings, so that it will count characters correctly when given UTF-8 input. This will be the easiest way to find the tags. Look at the demo programs in the `demos/glib_demos` directory on `eniac`, and read the GLib API documentation.

You will have two choices of implementation – either loading the entire file into the `GtkTextBuffer` and hiding tags and searching for the tags after it is loaded, or storing it in memory first, parsing it for tags, and then loading it into the buffer. Think about which choice will be easier for you and then plan it all out.

Submission

If the program consists of more than one source code file, create a directory named `hwk3_username`, with permission 700, and put all of your source code files and a makefile into it (and nothing else!) If it is a single source code file, name it `hwk3_username.c` or `hwk3_username.cc`, where `username` is the username you have on our UNIX system, and the extensions are for C and C++ respectively, with permission 600. Put the directory or the program into the directory

```
/data/yoda/b/student.accounts/cs493.73/projects/project3
```

on our file system.