System Programming – Project 2

Using FUSE, implement a file system that will display the text files in a source directory as images, especially for handling “ASCII art” or “ANSI art” files. Examples of these files can be found on the page https://cleaner.ansilove.org/gallery.html (the text source files are under the RELEASES page). You can create such text images from existing image files using the img2txt utility (part of the caca-utils package on Ubuntu).

For example, assume that the source directory has the path `/src` and contains the following hierarchy:

```
/src
|- file1.txt
|- file2.zip
|- file3.html
|- file4.ans
|- subdir/
|  |- file5.pd1
|  |- file6.c
```

And assume that the FUSE file system is mounted on the `/dst` path. It will contain only the text files from the source directory, with their extensions changed to “`.png`”:

```
/dst
|- file1.png
|- file3.png
|- file4.png
|- subdir/
|  |- file6.png
```

NOTE THAT THESE FILES ARE NOT STORED ON THE DISK, THEY WILL BE GENERATED WHEN THE FILE IS READ. For example, when the file `/dst/file4.png` is read, the resulting buffer will contain data that is automatically generated by converting the file `/src/file4.ans` to PNG format.

When a new text file is added to the source directory, the file will be available in the destination without having to remount. Similarly files deleted from the source directory will disappear from the destination directory.

To test whether a file is in a text format or not, you have to use the libmagic library. This package is available on major Linux distributions (`libmagic-dev` on Ubuntu). An example usage of this library can be found on the page https://gist.github.com/vivithemage/9489378

To convert a text file into a PNG image, you have to use the libansilove library. This package is available on major Linux distributions (`libansilove-dev` on Ubuntu). An example usage of this library can be found on the page https://github.com/ansilove/libansilove/blob/master/example/example.c

You are expected to submit a ZIP archive containing the code for the driver module through Ninova.

Important notes!

- Pay attention to the general guidelines for projects.
- You are required to submit the following files through the Ninova system as a zip file containing source codes of your test programs and information on how to use the program (if required).
- Make sure to return a meaningful error code when an error occurs.
- Don't blindly copy any code from other sources. (e.g. rofs)
- This is a **group project**. However, each member of the team **must make an individual submission**, even though the submitted files are the same for all members.
- Team members will be graded individually based on their performance in the lab/demo session as well as based on the project submitted. Students who are not present during the lab session will not receive a grade for the project, even though they may have made a submission through the Ninova system.

Any form of cheating or plagiarism will not be tolerated. The submitted work should be the work of the team; collaboration or code sharing between different teams will be regarded as cheating. Cheating also includes actions such as, but not limited to, submitting the work of others as one's own (even if in part and even with modifications) and copy/pasting from other resources, including Internet resources, (even when attributed). Serious offenses will be reported to the administration for disciplinary measures.