Dalhousie University

Topics in Program Comprehension

CSCI 6306

Architecture Extraction

Group : Something
Bhupendra Rajawat
Bryan Thomas D'silva
Saurabh Singh

June 12, 2017

Contents

Abstract

Your abstract.

1 Finding a Software System

We looked at the following software systems and games.

1. **Kodi**: Kodi is an open source software media player and entertainment hub. This is available on atleast seven platforms. This was one of the first softwares we looked at and kept it as a contender

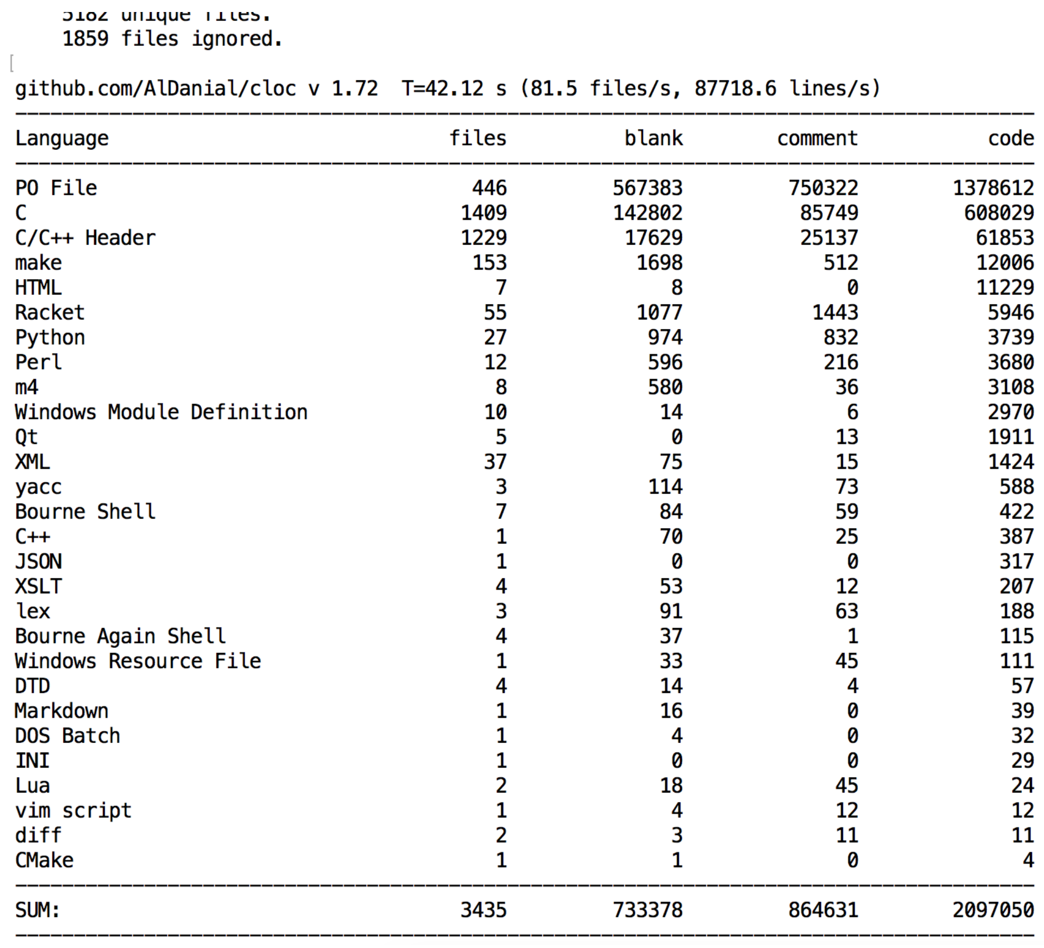
2. **Quake III**: The source code for Quake is challenging and interesting. However, there is almost no documentation for the same and hence we decided not to go ahead with this for now.

3. **GIMP**: GIMP is a cross platform image editor with an excellent community backing it up. We decided to extract the architecture for GIMP. We talk more about GIMP in the further sections.

We also looked at Doom3, SuperTuxKart, Wordpress, TuxRunner and Linux but found GIMP to be the most interesting on to use for this assignment

2 About GIMP

Gimp was designed initially as a General image manipulation program. This was later changed to GNU Image Manipulation Program. It is free and openly available to retouch and edit images, convert image formats and more specialized tasks targeted towards image manipulation. GIMP provides an interface to handle simple graphics needs without learning advanced methods for image manipulation. GIMP is more user friendly as compared to other photoshop techniques and can be modified to fit your needs. And the best part about it is that it is free.



```
    5182 unique files.
    1859 files ignored.

github.com/AlDanial/cloc v 1.72  T=42.12 s (81.5 files/s, 87718.6 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
PO File                        446         567383         750322        1378612
C                             1409         142802          85749         608029
C/C++ Header                  1229          17629          25137          61853
make                           153           1698            512          12006
HTML                             7              8              0          11229
Racket                          55           1077           1443           5946
Python                          27            974            832           3739
Perl                            12            596            216           3680
m4                               8            580             36           3108
Windows Module Definition       10             14              6           2970
Qt                               5              0             13           1911
XML                             37             75             15           1424
yacc                             3            114             73            588
Bourne Shell                     7             84             59            422
C++                              1             70             25            387
JSON                             1              0              0            317
XSLT                             4             53             12            207
lex                              3             91             63            188
Bourne Again Shell               4             37              1            115
Windows Resource File            1             33             45            111
DTD                              4             14              4             57
Markdown                         1             16              0             39
DOS Batch                        1              4              0             32
INI                              1              0              0             29
Lua                              2             18             45             24
vim script                       1              4             12             12
diff                             2              3             11             11
CMake                            1              1              0              4
-------------------------------------------------------------------------------
SUM:                          3435         733378         864631        2097050
-------------------------------------------------------------------------------
```

Figure 1: CLOC output for GIMP

3 Counting the lines of code

3.1 Count Lines of Code

Count Lines of Code, abbreviated as CLOC is a software used to count comment lines, blank lines and physical lines of code[?]. We used this software initially when we looked at what software system we need to pick. We can look at figure 1 for the output for GIMP.

3.2 LocMetrics

LocMetrics[**?**] is very similar to what CLOC does. We did find this software interesting because of how it visualized the output.

4 Architecture Extraction Tool

5 Directory Structure

The app directory is the main directory that we are concerned about. It contains the source code for GIMP without any external tools or plugins. Looking at the butterfly diagram below, we would ideally ignore the modules, plugins and libraries for now and focus on the app directory.

6 Entry Point

7 Build Process

Analysis of any application can not be completed without building it. Like every application provide documentation related to how one can build the application on any platform eg Linux, Mac etc. Building any application not only help to compile any application to create executable but end up to helping to figure out what are the entry point with in the code of application. Like in case of Gimp they have "MakeFile.am" which is a autogenerator file with in "master/app" directory to create "Makefile.in" . If you analyze Makefile.am you will get that "main.c" is the file from were compilation starts.
We tried to use Doxygen to generate call graphs for comparison and better understanding. However, the tool crashed on the first go. And on the second go generated quite a bunch of call graphs when run directly on the master/app directory so will leave this tool aside for now.
Talking about Dependencies required for the Gimp to get build you can check with in GIMP master folder you will find the INSTAL.in file which will tell how to build the application on particular platform with dependencies required. We are using Linux environment to build the application. To build this process we have two different type of build system provided one through a Tarball and other building from latest source (Git).We used the latter one

to build the application. Being a product of GNOME ,gimp use many of their pre existing libraries which they utilize for their other products as well like GLib , GObject ,GTK+, Cairo, Pango. The two major libraries which gimp uses are GEGL and babl which one need to install before building this application. And both can be installed either using tarball or git on system before building. Links for both the libraries are present on wiki page of the application.[3]
Once this is done we are require to just run autogen.sh which is a auto maker for configuration file. Once configuration file is generated we can run it to auto generate makefile. Also this is where you get to know if you have installed all libraries or not. This is the place where we got stuck as I already had a previous gimp installation and did not change the path for the new installation and end up having conflict between both of the installation which got resolved by just changing the default path. Once make file is generated you can run the "make" command and then "make install" to complete the build process. So if we see the major flow of the build after installing all libraries goes like as below:
autogen.sh—config—make—make install

8 Control Flow

9 Dependencies

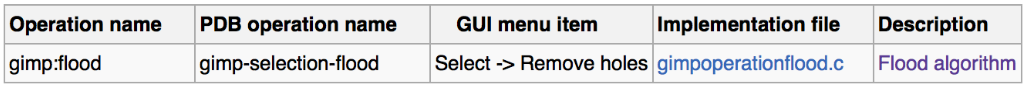
10 Functional Needs

11 Algorithms

To find the underlying algorithms behind the operations performed on gimp, the wiki for the same had a link to the algorithms used. We expected to find all the underlying algorithms here but encountered a problem. There was only one operation described here shown in figure 2.

Since this is implemented in the gimpoperationflood.c, the best way to find the algorithms for the other operations performed would be to have a look at where this file is located.

The app/operations folder is where this file located. Looking at the other



Operation name	PDB operation name	GUI menu item	Implementation file	Description
gimp:flood	gimp-selection-flood	Select -> Remove holes	gimpoperationflood.c	Flood algorithm

Figure 2: Documentation on operation algorithms

files in the folder, we got an idea of where to look incase we had to change or add any further operations. Since these operations are triggered using the GUI, changing something here would cause us to look at the UI elements to be updated as well.

12 Conclusion

12.1 Tables and Figures

Use the table and tabular commands for basic tables — see Table **??**, for example. You can upload a figure (JPEG, PNG or PDF) using the files menu. To include it in your document, use the includegraphics command as in the code for Figure **??** below.

1. Like this,
2. and like this.

. . . or bullet points . . .

- Like this,
- and like this.

We hope you find writeLaTeX useful, and please let us know if you have any feedback using the help menu above.

References

[1] http://www.locmetrics.com/

[2] https://github.com/AlDanial/cloc

[3] https://wiki.gimp.org/