CS 262, Fall 2014, All sections
Programming assignment 3 (Project03) (100p)

Due date: Wednesday, Dec. 03 2014 before midnight

1 Description

In your third project you will write a pair of *steganography* programs to hide and extract information in/from images. In the first part you will write a program to hide a binary file in the *least significant bits (LSBs)* of a *PGM* (or *PPM*) image. We call the image in which the information will be hidden the *cover*, the file that will be hidden is called the *payload*, and the modified image containing the payload is called the *stego image*. *PGM* and *PPM* are the simplest uncompressed image formats for gray and color images. To view these images you can use several program. On a linux machine or a mac you can use *gimp* or *GraphicConverter*. On a Windows machine the best program for viewing all image formats is *IrfanView*. You will find it easily.

To get you started we have provided three files:

`http://cs.gmu.edu/~zduric/cs262/Homeworks/image.h`,
`http://cs.gmu.edu/~zduric/cs262/Homeworks/image.c`, and
`http://cs.gmu.edu/~zduric/cs262/Homeworks/Stego.c`.

image.c has several functions for reading and writing *PGM* and *PPM* images and binary files. Several macros and examples of their use have been provided to simplify image byte manipulation. You will need to use 8 bytes of an image to hide one byte of information (remember that only LSBs of the cover image can be modified). You will use the first 16 bytes of the cover image to embed 16 bits of $b.size$ (two least significant bytes of the size field for the binary data), next 32 bytes of the cover will be used to embed the file extension for the payload file, and after that you will use $8 * b.size$ bytes to embed the payload ($b.data$).

Please note that you can get and set image bytes and the binary data using the provided macros. Unless you become really ambitious and decide to play with color (*PPM*) images you only need to use *GetGray*, *SetGray*, and *GetByte* macros. The skeleton program can be used to understand the operation of the methods. To compile the programs you should do the following commands in your make file:

gcc image.c -c
gcc Stego.c -c
gcc Stego.o image.o -o Stego

Please note that *Stego* requires three parameters, a PGM image file, the name of the output (*PGM*) file, and finally the payload file.

When you complete your information hiding program you will write an information extraction program *StegoExtract.c*. *StegoExtract* will take a stego image and extract a binary file hidden in it using the inverse of the process used in embedding. You should test your program on a PGM image with several different payload files. If you feel like an experienced steganographer you can make your functions work with color image files (*PPM*) for 10% extra credit.

2 Example of operation

./Stego half.pgm half_stego.pgm Stego.c
... Reading input file 'half.pgm'
... Reading binary file 'Stego.c'
... hidden file type = 'txt'
... Writing file 'half_stego.pgm'

Note that if the file extension is shorter than 3 characters 'txt' is used.

3 Internals

You *should* use functions *setlsbs* and *getlsbs* from *Lab11* to embed a byte into 8 bytes of cover data and to extract a byte from 8 bytes of stego image. You should only modify *Stego.c* file.

4 Instructions for Submission

1. You should create a folder and put all your code, your Makefile, and input and output files in it. Your images should be small so that each file is less than 256K in size (512×512). Note that you only need a single input PGM image, but there may be several output images. Use script to show your session in which you compile and run your code for various combination of image and input files.
   Important note: You should follow all directions exactly so that we can test your code in combinations with our versions of *Stego* and *StegoExtract* programs.

2. Use *tar* command to create an archive of your script file, your fully commented source code, and your Makefile. You should follow these naming rules:

   (a) Name your c source file as
   <your email account>_<your class section #>_project3_<embed,extract>.c
   For example, *cliu6_section001_project3_embed.c*, *cliu6_section001_project3_extract.c*

   (b) Name your script as
   <your email account>_<your class section>_project3
   For example, *cliu6_section001_project3*
   If you used other name for the script, change it by using unix command *mv*.

   (c) Your Makefile should include *clean* as well. Makefile name is still *Makefile*.

   (d) Use *tar cvf* to compress all files including source code, Makefile and script to
   <your email account>_<your section>_project3.tar.
   NOTE: DO NOT compress the folder. PLEASE compress files directly.
   For example, if all your files are in directory *./Project3*. You do the following steps:

   > *cd ./Project3*

   > *tar cvf cliu6_section001_project3.tar <related file names>* or
   > *tar cvf cliu6_section001_project3.tar ** (* means all files in this directory)

   (e) Double check that you followed the above procedures before you submit the compressed tar file.

3. Submit your *tar* file on Blackboard.