EECS 22: Advanced C Programming
Week 5

Mina Moghadam
mchmghdm@uci.edu

10/27/2017

Agenda

- General Information
- Makefile
- Advanced DIP operations
- Submission

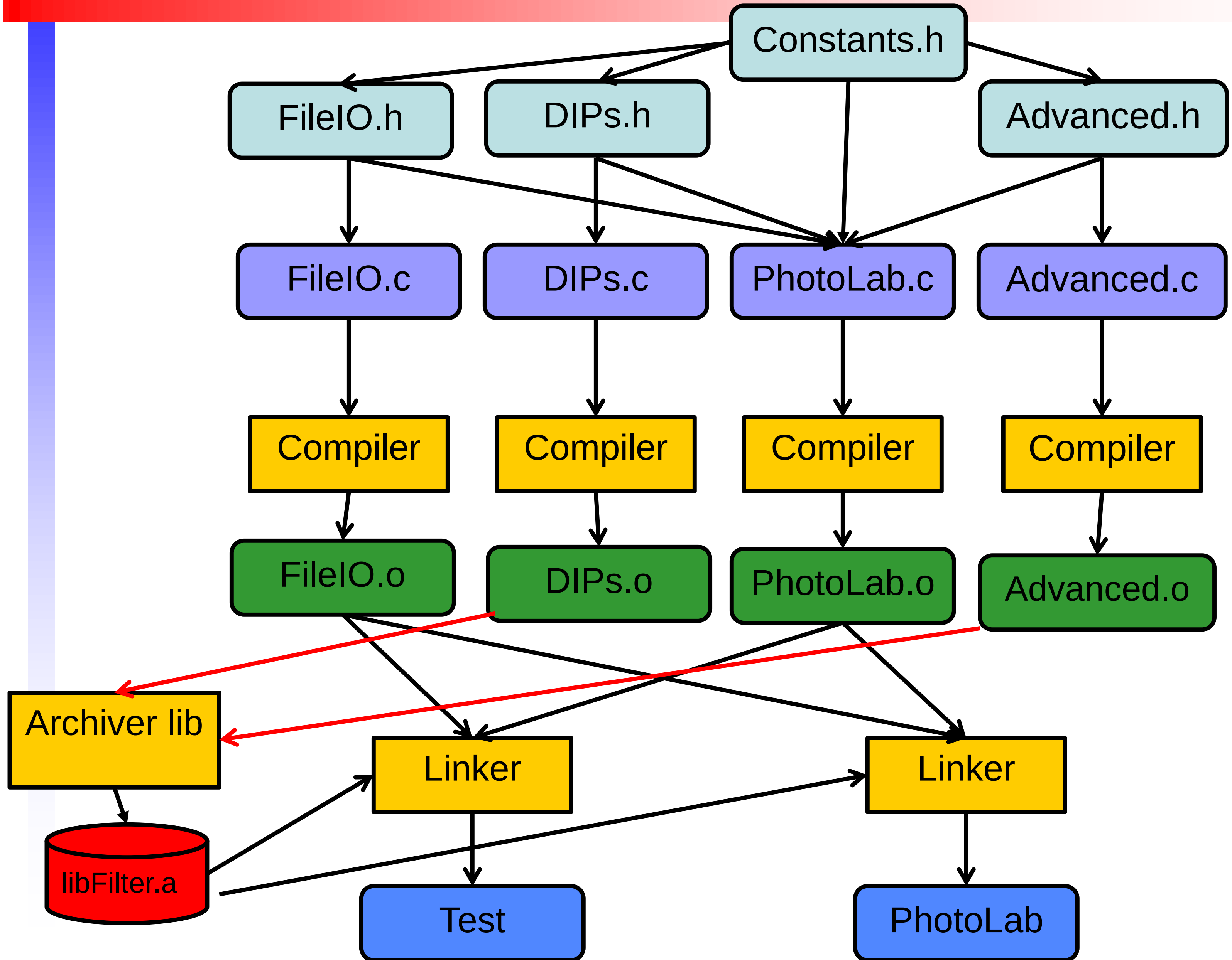
Assignment 3

- A menu driven digital image processing program [100 pts]
- Deadline: 2017/11/08, Wednesday, 6:00 pm
- Goal
  - Decomposing the PhotoLab in multiple source and header files
  - Adding new DIP operations
    - Add Noise to the image
    - Sharpen the image
    - Posterize the image
    - Bonus point: add ‘Motion Blur’ to the image

Decompose into Modules

File	Content
Photolab.c	Main(), PrintMenu(), AutoTest()
Constants.h	Constants like WIDTH and HEIGHT
FileIO.h FileIO.c	Declaration and definition of LoadImage() and SaveImage()
DIPs.h DIPs.c	Declaration and definition of BlackNWhite(), Negative(), ColorFilter(), Edge(), Shuffle(), VFlip(), VMirror(), AddBorder()
Advanced.h Advanced.c	Noise(), Sharpen(), Posterize(), MotionBlur()

Decompose into Modules



Compilation Commands

- Generate the object files for each module

```
% gcc -c FileIO.c -o FileIO.o -ansi -std=c99 -Wall
% gcc -c DIPs.c -o DIPs.o -ansi -std=c99 -Wall
% gcc -c Advanced.c -o Advanced.o -ansi -std=c99 -Wall
% gcc -c PhotoLab.c -o PhotoLab.o -ansi -std=c99 -Wall
```

- Create libraries

```
% ar rc libFilter.a DIPs.o Advanced.o
% ranlib libFilter.a
```

- Linking with the library

```
% gcc -ld PhotoLab.o FileIO.o –L. –lFilter –o PhotoLab
```

- Execute the program

```
% ./PhotoLab
```

Make and Makefile

Problem:

- When building a large program with multiple source files:
  - If small changes are made to one file, the entire program needs to be recompiled

Solution:

- many systems provides special utilities that recompile the only modified program file
  - In Linux systems the utility is called **make**
  - The utility make reads a file called **makefile**
  - **Makefile** contains instruction for compiling and linking the program

Make and Makefile

- Moreover!
  - we can put all the commands for compilation into a Makefile and use the make utility to automatically build the executable program from source code!

Makefile

Your Makefile has the following targets:

- **All:** the target to generate the executable programs
- **Clean:** the target to clean all the intermediate files, e.g. object files and the executable programs
- **PhotoLabTest:** the target to create and run PhotoLabTest
- **PhotoLab:** the target to generate the executable program PhotoLab

Requirement:

- There must be a rule for each object file depending on the corresponding .c file and any other needed dependency. Dependencies which are not needed will reduce the points.

Note:

we expect minimal build commands executing when only one file is touched

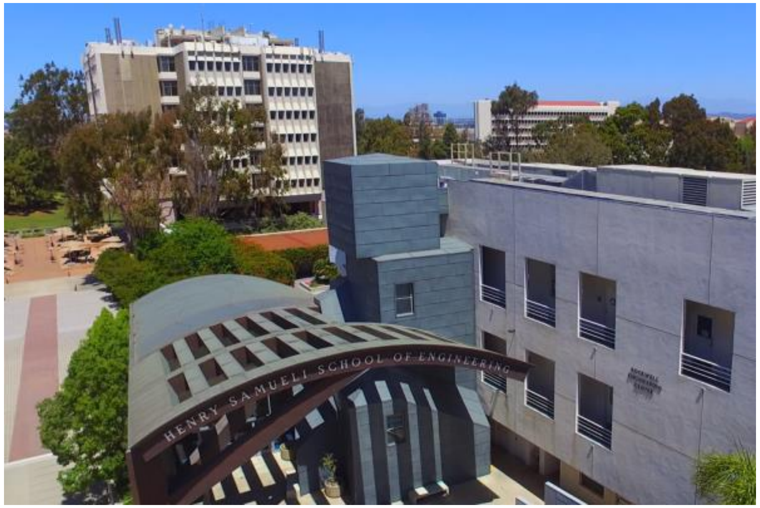
Advanced DIP Operations

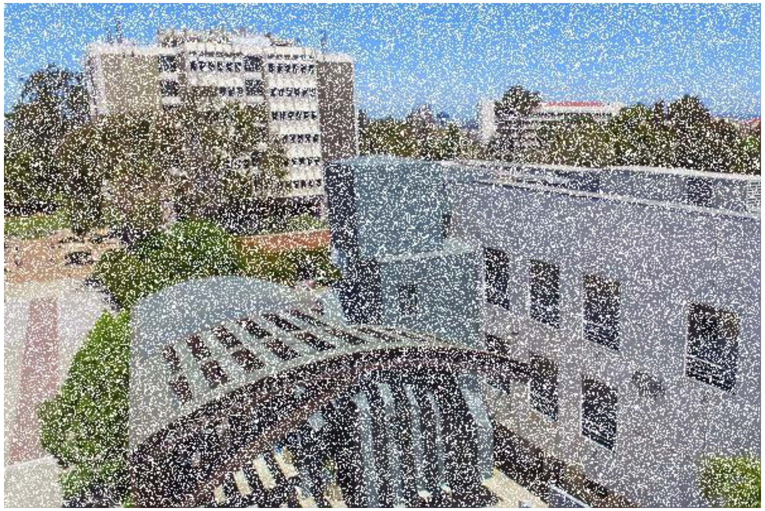
- New Operations
  - Add noise to the image
  - Sharpen the image
  - Posterize the image
  - Motion Blur (Bonus)

- The menu looks like:

```
--------------------------------
1: Load a PPM image
2: Save an image in PPM and JPEG format
3: Change a color image to black and white
4: Make a negative of an image
5: Color filter an image
6: Sketch the edge of an image
7: Shuffle an image
8: Flip an image vertically
9: Mirror an image vertically
10: Add border to the image
11: Add noise to an image
12: Sharpen an image
13: Posterize an image
14: Motion Blur
15: Test all functions
16: Exit
please make your choice:
```

Noise





```
void AddNoise(
          int n,
          unsigned char R[WIDTH][HEIGHT],
          unsigned char G[WIDTH][HEIGHT],
          unsigned char B[WIDTH][HEIGHT]);
```

- n: defines the percentage of white pixels
- Functions rand() and srand() should be used to generate random white spots
- Each execution creates a different result

Submission

The submission should include these files

- PhotoLab.c
- Constants.h
- FileIO.c
- FileIO.h
- DIPs.c
- DIPs.h
- Advanced.c
- Advanced.h

- Makefile

- PhotoLab.script
- PhotoLab.txt

Submission

In the Photolab.script, we expect following content:

1. Start the script by typing the command: script
2. Compile and run PhotoLab by using your Makefile:
   - type “make clean”, then “make”, then “./PhotoLab
3. Choose ’Test all functions’
   - The file names must be ’BlackNWhite’, ’Negative’, ’ColorFilter’, ’Edge’, ’Shuffle’, ’Vflip’, ’Vmirror’, ’Noise’, ’Sharpen’, ’Posterize’, and ’MotionBlur’
4. Exit the PhotoLab
5. Compile and run PhotoLabTest
   - type ’make PhotoLabTest’
6. Test the dependencies in your Makefile
   - type ’touch Advanced.c’, then ’make PhotoLab’
7. Stop the script by typing the command: exit
8. Rename the script file to PhotoLab.script

Time Management

- You have two weeks to complete this assignment

- We recommend the following schedule
- Week1
  - Decompose the program
    - Before decomposing, make sure your program compiles and works
  - Create the Makefile
    - Test the Makefile, Back up your work
  - Implement the first DIP function

- Week 2
  - Implement the remained advanced DIP functions
  - Enable/disable the DEBUG mode
  - Prepare your submission