

# EECS 22L: Software Engineering Project in C Language

## Lecture 8

Rainer Dömer

doemer@uci.edu

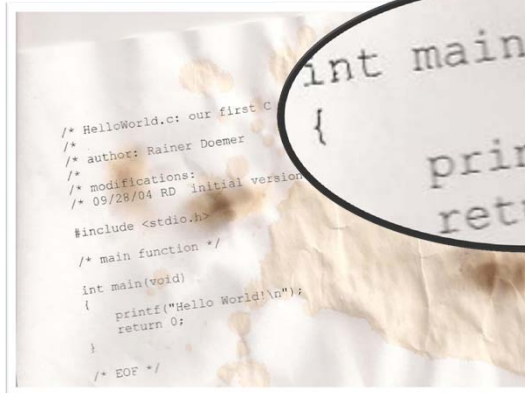
The Henry Samueli School of Engineering  
Electrical Engineering and Computer Science  
University of California, Irvine

## Lecture 8: Overview

- Project 2 Technical Discussion and Advise
  - Application specification
  - Application components, pipeline structure
  - Data structure organization
  - Approach and algorithm

## Project 2

- Introduction
  - OCR: Optical Character Recognition
  - Example:



EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

3

## Application Specification

- OCR Program Specification
  - Overall program flow
    1. Loading a scanned image into the program and displaying it
    2. Image preprocessing
    3. OCR
      - a) identifying and cropping single characters from the image
      - b) comparing cropped characters with reference images from a font data base
      - c) storing recognized characters in a text file data structure
    4. Text postprocessing
    5. Text file output

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

4

## Technical Advise

- Overview of an OCR Program
  - Major system components

EECS22L: Software Engineering Project in C, Lecture 8 (c) 2014 R. Doemer 5

## Technical Advise

- Overview of an OCR Program
  - Pipelining structure

EECS22L: Software Engineering Project in C, Lecture 8 (c) 2014 R. Doemer 6

## Technical Advise

- Overview of an OCR Program
  - Pipelining structure with loops

Image Data

Text Data

EECS22L: Software Engineering Project in C, Lecture 8 (c) 2014 R. Doemer 7

## Technical Advise

- Essential Data Structures
  - Image
    - For scanned text and single characters
    - Color image: 2D array of RGB pixels
    - Greyscale image: 2D array of intensities
    - Black and white image: 2D array of bits
    - Dynamic sizes (width, height)
  - Image operations
    - Color manipulation
    - Cropping
    - Resizing
    - Rotation
    - ...

EECS22L: Software Engineering Project in C, Lecture 8 (c) 2014 R. Doemer 8

## Technical Advise

- NetPBM Package
  - Graphics manipulation programs and library
    - Over 220 separate programs in the package
    - `man netpbm`
  - File formats are pbm(5), pgm(5), ppm(5), and pam(5)
    - Most commands have "pbm", "pgm", "ppm", or "pnm" in name
    - Examples: `jpegtopnm`, `pnmscale`, `pnmrotate`, ...
  - Manual pages and HTML documentation available
    - `man pnmrotate`
  - Library with ANSI C API
    - `man libnetpbm`
  - NetPBM commands are best used in shell pipelines
    - Example to convert PNG files to JPEG files:
    - `bash# for f in *.png; do pngtopnm $f | pnmtojpeg >`basename $f.png`.jpg; done`
    - `tcsh# foreach f (*.png); pngtopnm $f | pnmtojpeg >`basename $f.png`.jpg; end`

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

9

## Technical Advise

- System Calls
  - Linux provides a system call interface to execute a shell command from within a C program
    - `#include <stdlib.h>`
    - `int system(const char *command);`
    - `system(command)` executes a command specified in `command` by calling `/bin/sh -c command`, and returns after the command has been completed
    - The return value is -1 on error (something failed), or the return value of the command otherwise
  - Example:
 

```
if (0 != system("pnmtojpeg image.ppm >image.jpg"))
{ fprintf(stderr, "Conversion to JPG failed!\n");
  exit(10);
}
```
  - `libNetPBM` provides a similar API:
    - `#include <netpbm/pm_system.h>`
    - `pm_system(stdinFeeder,..., stdoutAcceptor,..., command);`

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

10

## Technical Advise

- Essential Data Structures
  - Text
    - For recognized and post-processed text and fragments
    - For dictionary
      - List of rows
      - Rows represented by strings
    - Dynamic sizes (rows, columns)
  - Text operations
    - List operations
    - String operations
    - ...
  - Take a 2<sup>nd</sup> look at `string.h` (EECS22, Lecture 18)

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

11

## Technical Advise

- Overall OCR program control flow
  - setup (read reference character library)
  - read the scanned image
  - display the scanned image
  - repeat
    - get image processing instruction from user
    - apply image processing to the image
    - display the processed image
  - repeat
    - locate and crop a character from the image
    - match the character with the reference characters
    - put recognized characters into the text data structure
  - keyword matching (optional)
  - output the recognized text to a file

EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

12

## Technical Advise

- **Provided Image Files**

- 01\_HelloWorld\_Clean300DPI.jpg - clean scan of HelloWorld.c
- 02\_HelloWorld\_Wrinkles300DPI.jpg - wrinkled version of the previous
- 03\_HelloWorld\_Coffee300DPI.jpg - coffee-stained version of the previous
- 04\_HelloWorld\_Burned300DPI.jpg - partially burned version of the previous
- 05\_HelloWorld\_Slanted300DPI.jpg - slanted scan of the previous
  
- CourierNew12.docx - original document with Courier New font, 12pt
- CourierNew12.pdf - PDF version of the previous
- CourierNew12\_300DPI.jpg - clean scan of Courier New font sample
- CourierNew12\_300DPI.ppm - portable pixel map (PPM) format of the prev.
- CourierNew12\_300DPI.pbm - black&white version of the previous
  
- CourierNew12\_300DPI/33.jpg - 30x46 pixel image of ASCII code 33 (!)
- CourierNew12\_300DPI/....jpg ...
- CourierNew12\_300DPI/126.jpg - 30x46 pixel image of ASCII code 126 (~)

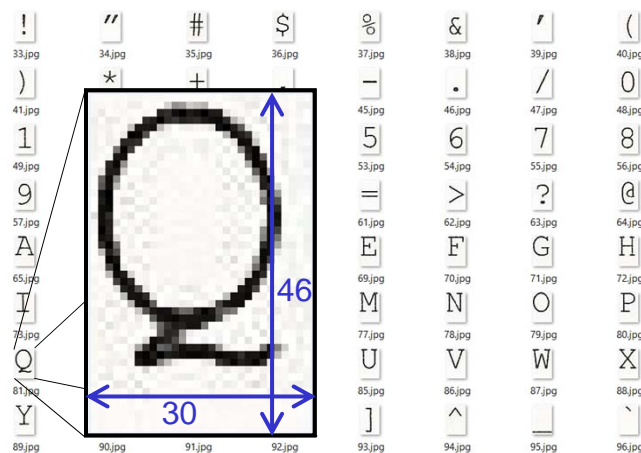
EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

13

## Technical Advise

- **Provided Font Database**



EECS22L: Software Engineering Project in C, Lecture 8

(c) 2014 R. Doemer

14