

EECS 10: Computational Methods in Electrical and Computer Engineering

Lecture 25

Rainer Dömer

doemer@uci.edu

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

Lecture 25: Overview

- Course Administration
 - Reminder: Final course evaluation
- Translation Units
 - Introduction
 - Compiler components
 - Modules
 - Program example **PhotoLab2**
 - Module **FileIO**
 - Module **Age**
 - Module **Main**

Course Administration

- **Final Course Evaluation**
 - Open **until Sunday night** (end of 10th week)
 - Nov. 15, 2010, through Dec. 5, 2010, 11:45pm
 - Online via EEE Evaluation application
- **Mandatory Evaluation of Course and Instructor**
 - Voluntary
 - Anonymous
 - Very valuable
 - Help to improve this class!
- **Please spend 5 minutes!**

EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

3

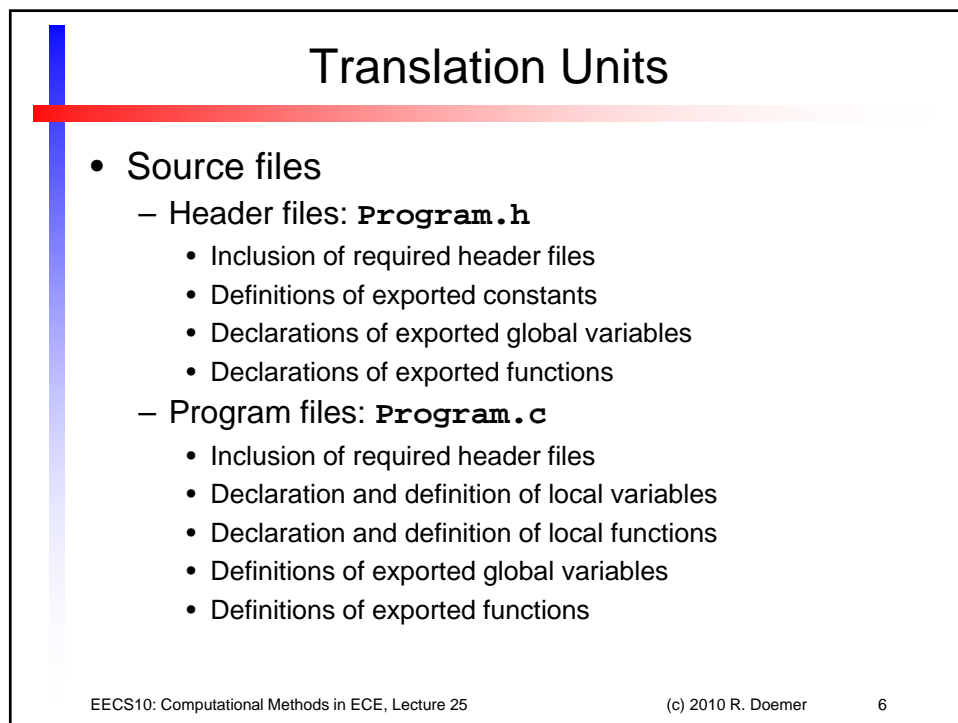
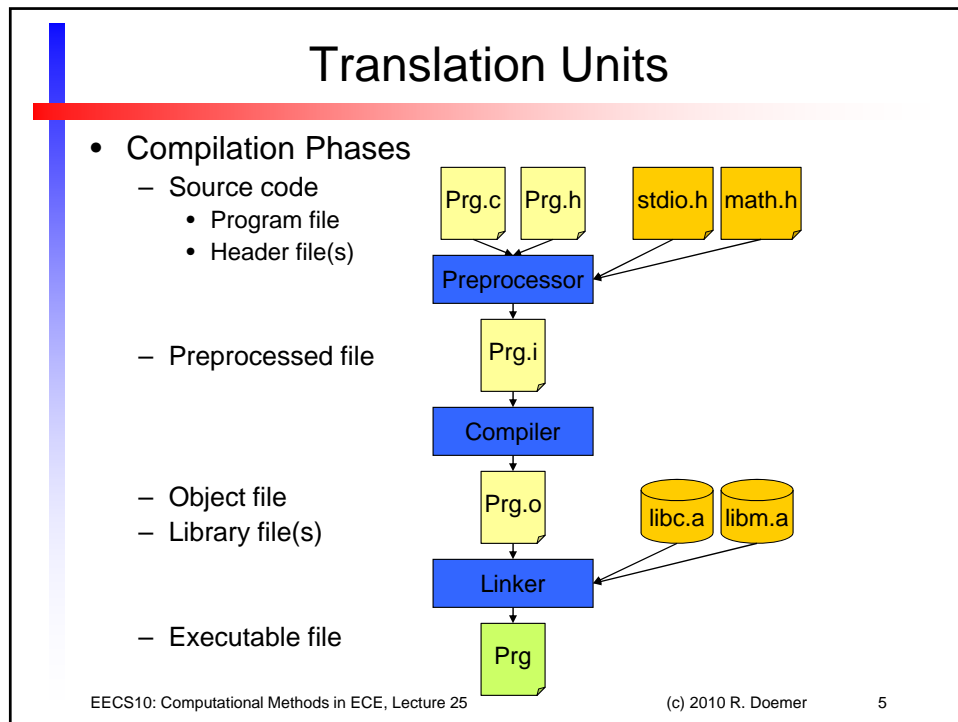
Translation Units

- **Introduction**
 - C compilation process is a sequence of phases
 - Preprocessing (handle # directives)
 - Scanning and parsing (generate internal data structure)
 - Instruction generation (emit stream of CPU instructions)
 - Assembly (generate binary object file)
 - Linking (combine objects into executable file)
 - C compiler consists of separate components
 - Preprocessor (processes # directives)
 - Compiler (compiles and assembles code)
 - Linker (processes object files and libraries)

EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

4



Translation Units

- C Preprocessor
 - preprocesses source files
 - handles # directives
- Preprocessing Directives
 - Constant definition
 - Macro definition
 - Header file inclusion
 - Conditional compilation

```
#define WIDTH 480
```

```
#define ABS(x) (x>0 ? x : -x)
```

```
#include <stdio.h>
```

```
#define DEBUG /* comment out to turn debugging off */
...
#ifdef DEBUG
printf("value of x is now %d\n", x);
#endif
```

EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

7

Translation Units

- Object files
 - **Program.o**
 - Compiled object code of source file **Program.c**
 - Use option **-c** in GNU compiler call to create object files
 gcc -c Program.c -o Program.o -Wall -ansi
 - **Library.a**
 - Archive of compiled object files
- Executable file
 - **Program**
 - Object files and libraries linked together into a complete file ready for execution
 - GNU compiler recognizes object files by **.o** suffix, so object files and libraries require no special option
 gcc Program.o -lc -lm -o Program

EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

8

Translation Units

- Multiple Translation Units
 - C programs can be partitioned into multiple translation units, aka. *modules*
 - Modules typically consist of
 - Module header file (file suffix `.h`)
 - Module program file (file suffix `.c`)
 - Module object file (file suffix `.o`)
 - Modules are *linked* together
 - Linker combines object files and required libraries into an executable file
 - `gcc Program.o Mod1.o Mod2.o -lc -lm -Wall -ansi -o Program`

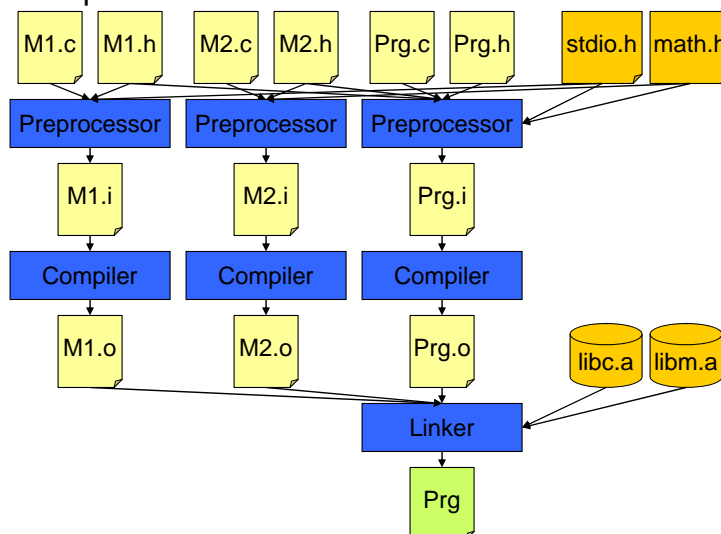
EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

9

Translation Units

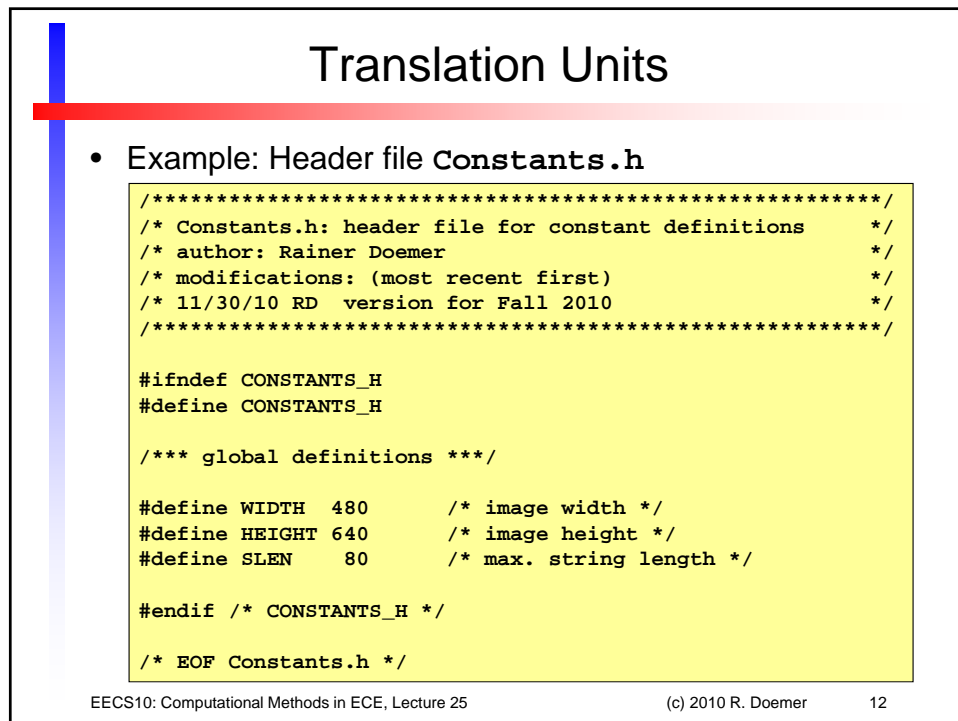
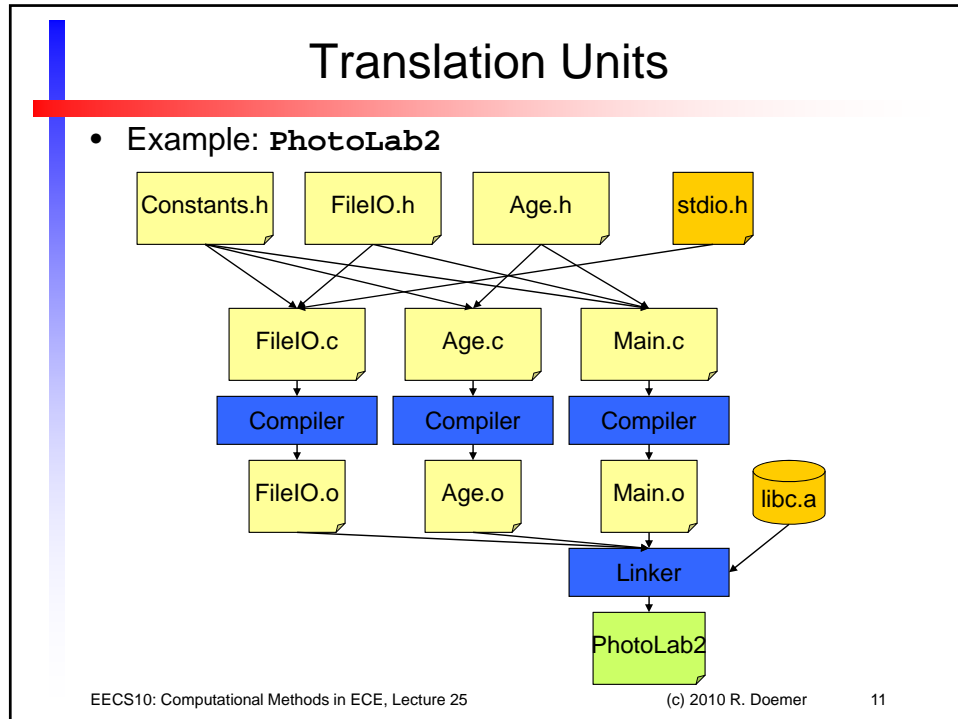
- Multiple Translation Units



EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

10



Translation Units

- Example: Header file `FileIO.h`

```

/*****
/* FileIO.h: header file for I/O module          */
/*****
#ifndef FILE_IO_H
#define FILE_IO_H

#include "Constants.h"

int ReadImage(      /* read image from file */
    char Filename[SLEN],
    unsigned char R[WIDTH][HEIGHT],
    unsigned char G[WIDTH][HEIGHT],
    unsigned char B[WIDTH][HEIGHT]);

int SaveImage(     /* write image to file */
    char Filename[SLEN],
    unsigned char R[WIDTH][HEIGHT],
    unsigned char G[WIDTH][HEIGHT],
    unsigned char B[WIDTH][HEIGHT]);

#endif /* FILE_IO_H */
/* EOF FileIO.h */

```

EECS

Translation Units

- Example: Program file `FileIO.c`

```

/*****
/* FileIO.c: program file for I/O module        */
/*****
#include <stdio.h>
#include "FileIO.h"

/**/ function definitions ***/

int ReadImage(char Filename[SLEN],
    unsigned char R[WIDTH][HEIGHT],
    unsigned char G[WIDTH][HEIGHT],
    unsigned char B[WIDTH][HEIGHT])
{
    /* ... function body ... */
}

int SaveImage(char Filename[SLEN],
    unsigned char R[WIDTH][HEIGHT],
    unsigned char G[WIDTH][HEIGHT],
    unsigned char B[WIDTH][HEIGHT])
{
    /* ... function body ... */
}

/* EOF FileIO.c */

```

EECS

Translation Units

- Example: Header file `Age.h`

```

/*****
/* Age.h: header file for aging operation */
/*****

#ifndef AGE_H
#define AGE_H

/** header files */
#include "Constants.h"

/** function declarations */

void Age( /* age the image */
         unsigned char R[WIDTH][HEIGHT],
         unsigned char G[WIDTH][HEIGHT],
         unsigned char B[WIDTH][HEIGHT]);

#endif /* AGE_H */

/* EOF Age.h */

```

Translation Units

- Example: Program file `Age.c`

```

/*****
/* Age.c: program file for aging operation */
/*****

#include "Age.h"

/** function definitions */

/* age the image so that it looks like an old photo */

void Age(
    unsigned char R[WIDTH][HEIGHT],
    unsigned char G[WIDTH][HEIGHT],
    unsigned char B[WIDTH][HEIGHT])
{
    /* ... function body ... */
}

/* EOF Age.c */

```


Translation Units

- Example: Program file **Main.c**

```

/*****
/* Main.c: main program file */
/*****
#include "Constants.h"
#include "FileIO.h"
#include "Age.h"

int main(void)
{
    unsigned char R[WIDTH][HEIGHT];
    unsigned char G[WIDTH][HEIGHT];
    unsigned char B[WIDTH][HEIGHT];

    if (ReadImage("swan.ppm", R, G, B) != 0)
    { exit(10); }
    Age(R, G, B);
    if (SaveImage("aged.ppm", R, G, B) != 0)
    { exit(10); }

    return 0;
} /* end of main */
/* EOF Main.c */

```

EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

17

Translation Units

- Example session:

```

% vi Constants.h
% vi FileIO.h
% vi FileIO.c
% vi Age.h
% vi Age.c
% vi Main.c
% gcc -c FileIO.c -o FileIO.o -Wall -ansi
% gcc -c Age.c -o Age.o -Wall -ansi
% gcc -c Main.c -o Main.o -Wall -ansi
% gcc FileIO.o Age.o Main.o -o PhotoLab2
%

```

swan.ppm



aged.ppm



EECS10: Computational Methods in ECE, Lecture 25

(c) 2010 R. Doemer

18