

EECS 10: COMP METHODS IN ECE

Discussion 2

Guantao Liu
guantaol@uci.edu

06/25/2015

Course Updates

- Everyone should be able to log into any of the HSSOE computer lab machines.
- The script to verify your submission
 - `/usr/ugrad/2004/fall/eecs10/bin/listfiles.py`
 - Permission has been adjusted, and it is now allowed to be executed by any user.

Hello World Example

- For a C program, you will have two different kinds of files.
 - Source code files: HelloWorld.c
 - Usually written in C programming languages
 - Plain text, readable to human beings
 - Executable files: HelloWorld (with no filename extension)
 - Usually written in machine languages
 - Binary digits (0/1), readable to CPU
- C compilers translate a source file to an executable file.
 - `gcc HelloWorld.c -o HelloWorld`

Hello World Example

- HelloWorld.c
 - `vi HelloWorld.c`
- HelloWorld
 - `vi HelloWorld`

The image shows two terminal windows side-by-side. The left window displays the source code of HelloWorld.c, which includes a header with author information and a main function that prints "Hello World!". The right window shows the output of the compiled binary, which is a large block of binary data (0s and 1s) representing the machine code.

```

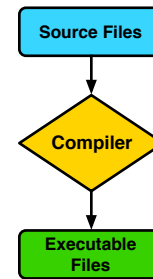
Terminal - ssh - 120x36
~/HelloWorld.c our first C program %
/* HelloWorld.c */
/* author: Guantao Liu */
/* modifications: */
/* 06/24/2015 G. Initial version */
#include <stdio.h>
/* main function */
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
~/HelloWorld.c 15L, 338C

Terminal - ssh - 120x36
HelloWorld [noel] 3L, 648C
1,1 Top
    
```

C Program Compilation

- C compilers translate source files to executable files.

```
vi SourceFile
gcc SourceFile -o ExecutableFile
./ExecutableFile
```
- To keep the source codes of a C program, `ExecutableFile` must be different from `SourceFile`.
- When you modify `SourceFile`, you need to recompile it to generate a new `ExecutableFile`.



EECS 10 Discussion 2, June, 2015

(c) 2015 Guantao Liu

5

Part 1: Exercise 2.25

- Print your initials in block letters. Construct each block letter out of the letter it represents.
 - Example: GL for “Guantao Liu”
 - Print character strings to the screen (`printf()`)
 - Hint: modified from the Hello World example
 - Text file: `initials.txt`
- Briefly describe what your program does and why you implemented it in this way

```

GGGGGGGGGG
G           G
G           G
G   G       G
GGGGGG     G
           G   G

LLLLLLLLLLLLL
L
L
L
L
  
```

EECS 10 Discussion 2, June, 2015

(c) 2015 Guantao Liu

6

Typescript

- A text file that captures an interactive sessions within the Linux shell.
- The typescript shows that your program runs correctly with the given input.
- Start the typescript with a command `script`, and stop it with `exit` or `ctrl-D`.
 - Don't start a text editor when recording.
 - Rename `typescript` after recording.

HelloWorld.script

```
zuma% script
Script started, file is typescript
zuma% ./HelloWorld
Hello World!
zuma% exit
exit
Script done, file is typescript
zuma% mv typescript HelloWorld.script
zuma% more HelloWorld.script
Script started on Wed Jun 24 14:52:39 2015
zuma% ./HelloWorld
Hello World!
zuma% exit
exit

Script done on Wed Jun 24 14:52:54 2015
```

Part 2: Deducting Two Timestamps

- Compute the difference of two timestamps
 - Each timestamp contains hours, minutes and seconds
 - Prompt for hours, minutes and seconds for each timestamp
 - Read two timestamps from stdin
 - Compute the difference and display the result
- Example:

Timestamp 1:	15 hours	21 minutes	37 seconds
Timestamp 2:	31 hours	57 minutes	27 seconds
Difference:	16 hours	35 minutes	50 seconds

EECS 10 Discussion 2, June, 2015

(c) 2015 Guantao Liu

9

Part 2: Deducting Two Timestamps

- Assumptions:
 - The second timestamp is greater than the first timestamp.
 - The input value for hours, minutes and seconds is always valid.
 - $0 \leq \text{hours}$
 - $0 \leq \text{minutes} \leq 59$
 - $0 \leq \text{seconds} \leq 59$
 - The input value for hours, minutes and seconds is integer.

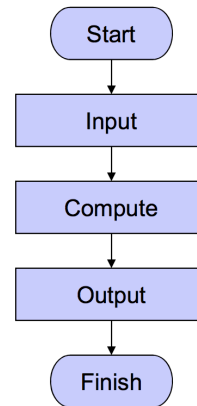
EECS 10 Discussion 2, June, 2015

(c) 2015 Guantao Liu

10

Part 2: Deducing Two Timestamps

- General Program Structure
 - Input
 - Computation
 - Output
- Input: read input data
 - `scanf()`
- Computation
- Output: write output data
 - `printf()`



EECS 10 Discussion 2, June, 2015

(c) 2015 Guantao Liu

11

Part 2: Deducing Two Timestamps

- Computation
 - Variable types
 - `int`, `float`, `double`, `char`, ...
 - Arithmetic operations
 - `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division), `%` (modulo), ...
 - Evaluation order: usually from left to right
 - Operators have different precedencies

EECS 10 Discussion 2, June, 2015

(c) 2015 Guantao Liu

12

subtraction.c

```

/* subtraction.c: deduct two integer numbers */
#include <stdio.h>

/* Main Function */
int main(void)
{
    int num1, num2, diff;

    printf("Please enter an integer: ");
    scanf("%d", &num1);
    printf("Please enter another integer: ");
    scanf("%d", &num2);

    diff = num2 - num1;

    printf("The difference of %d and %d is %d.\n", num1, num2,
        diff);

    return 0;
}

```

Comments

Preprocessor Directives

Main Function

Input

Computation

Output

EECS 10 Discussion 2, June, 2015 (c) 2015 Guantao Liu 13

Part 2: Deducting Two Timestamps

- Hint: Convert and store each timestamp triple as seconds before the subtraction
- Bonus: extend Part 2 to also handle days and weeks
 - 0 <= weeks
 - 0 <= days <= 6
 - 0 <= hours <= 23
 - 0 <= minutes <= 59
 - 0 <= seconds <= 59

EECS 10 Discussion 2, June, 2015 (c) 2015 Guantao Liu 14

Files for Assignment 1 Submission

- Part 1:
 - initials.c
 - initials.txt
 - initials.script
- Part 2:
 - timestamp.c
 - timestamp.txt
 - timestamp.script
- Due: Monday June 29, 2015 at 11:00pm