NWEN 241
Systems Programming

Week 1 Tutorial

Compilation Phases

Recall: C compilation undergoes 4 phases:

1) Preprocessing Phase
2) Compilation Phase
3) Assembly Phase
4) Linking Phase

- Usually, we use gcc to do all the phases and directly generate the binary executable
- We can also ask gcc to do certain phases

`gcc` Options

Phase	gcc Option	Result	Output File
Preprocessing	`-E`	Compilation unit	`.i` `.ii`
Compilation	`-S`	Assembler file	`.s`
Assembly	`-c`	Object file	`.o` `.obj`
Linking		Executable	Binary executable (`.exe` in Windows)

I/O Using Standard C Library

Recall: C provides a set of header files (standard C library) that you can use to write your code

C provides a standard library* which consists of the following headers:

`assert.h`	`float.h`	`math.h`	`stdarg.h`	`stdlib.h`
`ctype.h`	`limits.h`	`setjmp.h`	`stddef.h`	`string.h`
`errno.h`	`locale.h`	`signal.h`	**`stdio.h`**	`time.h`

- You don’t have to start from scratch!

I/O Streams

- C provides functions with input and output capability
- From the program's point of view, data input and data output are made possible through file streams
- Every C program has access to 3 such file streams: `stdin`, `stdout`, `stderr`

File	Description	Remarks
`stdin`	Standard input file	Connected to the keyboard
`stdout`	Standard output file	Connected to the screen
`stderr`	Standard error file	Connected to the screen

I/O Functions

- C input/output functions can be classified into 2 types:
  - Non-formatted input/output
    - `getchar`
    - `putchar`
    - `gets`
    - `puts`
  - Formatted input/output
    - `printf` and its variants
    - `scanf` and its variants

How To Use a Function

- Find its manual or documentation
  - In Linux terminal, use the **man** command
  - You can also search online
    - This website provides a pretty good documentation for the standard C library: https://www.tutorialspoint.com/c_standard_library/index.htm
- What to look for in the function manual?
  - What the function does
  - What header file(s) to include
  - What are the arguments to the function
  - What is the return type
  - What happens in case of errors

`printf()` and `scanf()`

- `printf()` writes a string to the standard output stream (stdout).
- The string is formatted using additional arguments that follow the initial string.

- `scanf()` accepts input from the standard input stream (stdin).
- The format of the expected items are specified and it returns the number of items successfully scanned