Systems@ETH zürich

Exercise Session 2

Systems Programming and Computer Architecture

Fall Semester 2025

Agenda

- More on C-programming…
- **.c** and **.h** files
- **Make** and **makefiles**
- **gcc** flags

Setting up your Environment

Any remaining issues setting up the working environment?

Bit Lab

Deadline for Assignment 1 is next week.

Questions?

C Programming Whirlwind Tour

Touching on this week’s lectures

Example Structure of a C file

```c
#include <stdio.h>

int i = 79;

static void print_name(void)
{
  const char s[] = "Mothy";
  printf("My name is %s and I work in CAB F %d\n", s, i);
}

int main(int argc, char *argv[])
{
  print_name();
  return 0;
}
```

- You have function definitions and declarations and calls.
- You have variable declarations

How about calling `print_name()` from another source file?

Or

How does the other_program.c know about the location / signature of `print_name()` ?

Solution: Header Files and Modules

- There is a difference between **declaration** and **definition**
  - Declaration gives the signature of the function / variable
  - Definitions gives the code / storage space for variables

- Put declarations in header files

```
void print_name(void);
```

```
void print_name(void)
{
  const char s[] = "Mothy";
  printf("My name is %s and I work
          in CAB F %d\n", s, i);
}
```

http://en.wikipedia.org/wiki/Header_file

Outsourced print_name()

```c
/* print_name.h */
void print_name(void);



/* print_name.c */

#include <stdio.h>

int i = 79;

void print_name(void)
{
  const char s[] = "Mothy";
  printf("My name is %s and I work in CAB F %d\n", s, i);
}
```

New Structure of Main

```
#include “print_name.h”

int main(int argc, char *argv[])
{
  print_name();
  return 0;
}
```

Note: You do not need to include stdio.h anymore, since you do not make use of printf here. print_name makes use of printf and stdio.h is included in print_name.h

`#include “print_name.h”` → Your header files (same directory)

`#include “../print_name.h”` (in the parent directory)

`#include “folder/print_name.h”` (in the subdirectory)

`#include <stdio.h>` → Header file of the system (libc)

Some C standard library headers: `<stdlib.h>, <math.h> …`

Different file types

Header Files (*.h)

- Forward declarations (function prototypes, …)
- Globally usable definitions, typedefs, structs, …
- [Macro definitions]

Source Files (*.c)

- Function definitions (source code)
- Variable storage
- Local (static) function declarations & definitions

Note: Everything that is declared in a header file which can be included is considered to be globally accessible. Only put there what's necessary i.e. the public interface

Header Files

- Header files are included by text injection (copy-paste) by macro pre-processor:

```
#include "header1.h"
#include <system-file>
```

- Include Header Guards to make sure that a header file is only included once in a compilation unit (roughly a C file):

```
#ifndef HEADER_FILE
#define HEADER_FILE

// the entire header file

#endif // HEADER_FILE
```

Compiling The Program

- Just executing gcc with your program.c does not work anymore
- You have to specify every source file you used:

```
gcc –o program program.c print_name.c
```

-o is used to name the **output**, if -o is not specified the output will be named **a.out** for historic reasons.

- You do not have to list the header files
  - gcc looks for header files in the current directory
  - gcc also looks for header files in the system include directories

make ?

- GNU make:
  - *“In software development, **Make** is a utility that automatically builds executable programs and libraries from source code by reading files called **makefiles** which specify how to derive the target program.” - https://en.wikipedia.org/wiki/Make_(software)*
  - *Only builds the parts if they are modified and necessary w.r.t. the makefiles.*
  - *https://makefiletutorial.com/*

Example Makefile (from assignment 1)

```
CC = gcc
CFLAGS = -O -Wall

btest: btest.c bits.c decl.c tests.c btest.h bits.h
       $(CC) $(CFLAGS) -o btest bits.c btest.c decl.c tests.c

clean:
       rm -f *.o btest
```

Usage:

`make` or `make btest`: runs the compilation but only if the files are modified

`make clean`: removes your generated binary file

Some hints

- Function Pointers
  http://www.cprogramming.com/tutorial/function-pointers.html

- Pointer Tutorial
  http://www.cplusplus.com/doc/tutorial/pointers/

- More on modules and header files
  – http://www.tutorialspoint.com/cprogramming/c_header_files.htm
- Make files (important for later…)
  – http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/
- More on this in the lecture next week… ☺

Demo

The compiler is your friend!

GCC Flags for better coding style

- -Werror
  - Make all warnings into errors.
- -Wpedantic
  - Issue all the warnings demanded by strict ISO C and ISO C++; reject all programs that use forbidden extensions
- -Wall
  - Enables a number of warnings about questionable code
- -Wextra
  - This enables some extra warning flags that are not enabled by -Wall (such as -Wuninitialized)

https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html

GCC Flags for catching errors at runtime

- -fsanitize=address
  - Instrument code to detect memory errors
- -fsanitize=undefined
  - Instrument code to detect undefined behavior at runtime
- -fstack-protector-all
  - Instruments code to detect buffer overflows on the stack

- **Comes with a runtime cost!**

https://gcc.gnu.org/onlinedocs/gcc/Instrumentation-Options.html

Exercise

Let’s match some C expressions.

Exercise: Matching expressions

a

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two's complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., `W == 31`).

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
e. ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

a

Answers

a. ~(~a | (b ^ (MIN_INT + MAX_INT)))

[✓ b.] ((a ^ b) & ~b) | (~(a ^ b) & b)

c. 1 + (a « 3) + ~a

d. (a « 4) + (a « 2) + (a « 1)

e. ((a < 0) ? (a + 3) : a) » 2

f. a ^ (MIN_INT + MAX_INT)

g. ~((a | (~a + 1)) » W) & 1

h. ~((a » W) « 1)

i. a » 2

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two's complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
a * 7
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two's complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
e. ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
a * 7
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two’s complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Answers

a. ~(~a | (b ^ (MIN_INT + MAX_INT)))

b. ((a ^ b) & ~b) | (~(a ^ b) & b)

[✓ c.] 1 + (a « 3) + ~a

d. (a « 4) + (a « 2) + (a « 1)

e. ((a < 0) ? (a + 3) : a) » 2

f. a ^ (MIN_INT + MAX_INT)

g. ~((a | (~a + 1)) » W) & 1

h. ~((a » W) « 1)

i. a » 2

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

One’s complement of a

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two’s complement format for signed ints.
- `MAX_INT` and `MIN_INT` are the maximum and minimum representable signed integer values, respectively.
- `W` is one less than the number of bits needed to represent an int (i.e., `W == 31`).

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
e. ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

One’s complement of a

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two’s complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Answers

a. ~(~a | (b ^ (MIN_INT + MAX_INT)))

b. ((a ^ b) & ~b) | (~(a ^ b) & b)

c. 1 + (a « 3) + ~a

d. (a « 4) + (a « 2) + (a « 1)

e. ((a < 0) ? (a + 3) : a) » 2

[✓ f.] a ^ (MIN_INT + MAX_INT)

g. ~((a | (~a + 1)) » W) & 1

h. ~((a » W) « 1)

i. a » 2

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
a / 4
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two’s complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
e. ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
a / 4
```

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
[✓ e.] ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two's complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
a & b
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two’s complement format for signed ints.
- `MAX_INT` and `MIN_INT` are the maximum and minimum representable signed integer values, respectively.
- `W` is one less than the number of bits needed to represent an int (i.e., `W == 31`).

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
e. ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

a & b

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two's complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Answers

[✓ a.] ~(~a | (b ^ (MIN_INT + MAX_INT)))

b. ((a ^ b) & ~b) | (~(a ^ b) & b)

c. 1 + (a « 3) + ~a

d. (a « 4) + (a « 2) + (a « 1)

e. ((a < 0) ? (a + 3) : a) » 2

f. a ^ (MIN_INT + MAX_INT)

g. ~((a | (~a + 1)) » W) & 1

h. ~((a » W) « 1)

i. a » 2

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
(a < 0) ? 1 : -1
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two’s complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., W == 31).

Answers

```
a. ~(~a | (b ^ (MIN_INT + MAX_INT)))
b. ((a ^ b) & ~b) | (~(a ^ b) & b)
c. 1 + (a « 3) + ~a
d. (a « 4) + (a « 2) + (a « 1)
e. ((a < 0) ? (a + 3) : a) » 2
f. a ^ (MIN_INT + MAX_INT)
g. ~((a | (~a + 1)) » W) & 1
h. ~((a » W) « 1)
i. a » 2
```

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15

Exercise: Matching expressions

```
(a < 0) ? 1 : -1
```

Assumptions

- a and b are declared as int in C.
- The machine uses 32-bit two's complement format for signed ints.
- MAX_INT and MIN_INT are the maximum and minimum representable signed integer values, respectively.
- W is one less than the number of bits needed to represent an int (i.e., `W == 31`).

Answers

a. `~(~a | (b ^ (MIN_INT + MAX_INT)))`

b. `((a ^ b) & ~b) | (~(a ^ b) & b)`

c. `1 + (a « 3) + ~a`

d. `(a « 4) + (a « 2) + (a « 1)`

e. `((a < 0) ? (a + 3) : a) » 2`

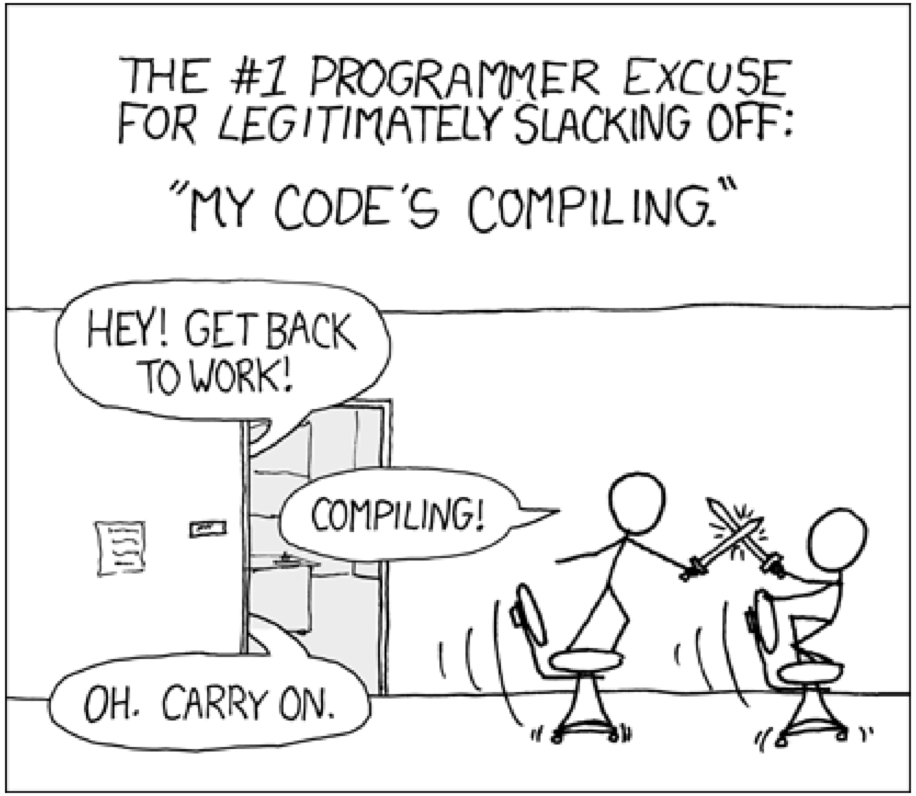
f. `a ^ (MIN_INT + MAX_INT)`

g. `~((a | (~a + 1)) » W) & 1`

[✓ h.] `~((a » W) « 1)`

i. `a » 2`

Systems Programming and Computer Architecture Exam, Autumn Term 2022, Question 15



Good luck and have fun!