Jair Ulises Nino-Espino

09/17/2018

Computer Science 250

Homework 1 – From Binary to C

Q1. Representing Datatypes in Binary

a1) Convert +57₁₀ to binary

2 ⁰ = 1	57 = 32 + 25
$2^1 = 2$	57 = 32 + 16 + 9
2 ² = 4	57 = 32 + 16 + 8 + 1
$2^3 = 8$	
2 ⁴ = 16	
2 ⁵ = 32	
2 ⁶ = 64	Binary: 0011 1001

a2) Convert +57₁₀ to hexadecimal

$$57/16 = 3.5625$$
 $16 \times 0.5625 = 9 \rightarrow 3R9$
 $3/16 = 0.1875$
 $16 \times 0.1875 = 3 \rightarrow 0R3$
Hexadecimal: 39

b1) Convert -17₁₀ to binary

2 ⁰ = 1	17 = 16 + 1
$2^1 = 2$	Positive: 0001 0001
2 ² = 4	Subtract 1: 0001 0000
$2^3 = 8$	
2 ⁴ = 16	
2 ⁵ = 32	
2 ⁶ = 64	2s complement Binary: 1110 1111

b2) Convert -17₁₀ to hexadecimal

2s complement Binary: 1110 1111

1110 = 14 → E

1111 = 15 → F

Hexadecimal: EF

c1) Convert +23.0₁₀ to 32-bit IEEE floating point in binary

$2^0 = 1$	+23 = +16 + 4 + 2 + 1 = +10111
$2^1 = 2$	$0.0 \times 2 = 0.0$
$2^2 = 4$	
$2^3 = 8$	+23.0= +10111.00000000000000000000000000000000
2 ⁴ = 16	= +1.011100000000000000000000000000000000
2 ⁵ = 32	Sign = 0, P = 4
$2^6 = 64$	Exp = bias + p = 127 + 4 = 131 = 1000 0011) ₂
	Binary: 0 1000 0011
	01110000000000000000000

c2) Convert +23.0 $_{10}$ to 32-bit IEEE floating point in hexadecimal

 $0100 = 4 \rightarrow 4$ $0001 = 1 \rightarrow 1$

1011 = 13 → B

1000 = 8 → 8

 $0000 = 0 \rightarrow 0$

 $0000 = 0 \rightarrow 0$

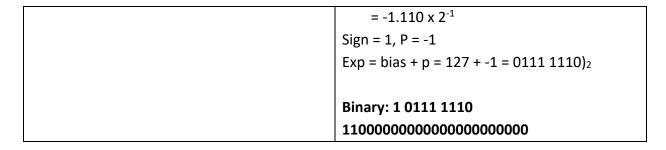
 $0000 = 0 \rightarrow 0$

0000 = 0 → 0

Hexadecimal: 41B80000

d1) Convert -0.875₁₀ to 32-bit IEEE floating point in binary

, = 01	•
$2^0 = 1$	-0 = -0
$2^1 = 2$	0.875 x 2 = 1.75
$2^2 = 4$	1.75 x 2 = 1.5
$2^3 = 8$	0.5 x 2 = 1.0
2 ⁴ = 16	$0.0 \times 2 = 0.0$
2 ⁵ = 32	
2 ⁶ = 64	-0.875 = -0.110



d2) Convert -875 $_{10}$ to 32-bit IEEE floating point in hexadecimal

```
1011 = 13 → B

1111 = 15 → F

0110 = 6 → 6

0000 = 0 → 0

0000 = 0 → 0

0000 = 0 → 0

0000 = 0 → 0

0000 = 0 → 0

Hexadecimal: BF600000
```

e) Represent the ASCII string "A String for 250!" (not including the quotes) in hexadecimal

A = 41	"A String for 250!" →
Space = 20	Hexadecimal: 41 20 53 74 72 69 6E 67 20 66
S = 53	6F 72 20 32 35 30 21
t = 74	
r = 72	
i = 69	
n = 6E	
g = 67	
f = 66	
o = 6F	
r = 72	
2 = 32	
5 = 35	
0 = 30	
! = 21	

f) Give an example of a number that cannot be represented as a 32-bit signed integer. 2³³ is a number that cannot be represented as a 32-bit signed integer.

Q2. Memory as an Array of Bytes

(a) Where do each of the following variables live (global data, stack, or heap)?

```
a. u: lives in the stack
b. v_ptr: lives in the stack
c. *v_ptr: lives in the global
d. w_ptr: lives in the stack
e. *w ptr: lives in the heap
```

(b) What is the value returned by main ()?

The main function should return 1 since the variable c is greater than 10.5. If the main is compile, the foo function would be call and return the value of 7 + 4 which equals 11. 11 is greater than 10.5 so 1 would be the output.

Q3. Compiling and Testing C Code

```
jun@login-teer-12 [production] ~ $ g++ -00 -o myProgramUnopt prog.c
jun@login-teer-12 [production] ~ $ time ./myProgramUnopt
C[111][392]=-1801792042

real 0m1.085s
user 0m1.074s
sys 0m0.002s
jun@login-teer-12 [production] ~ $ g++ -03 -o myProgramOpt prog.c
jun@login-teer-12 [production] ~ $ time ./myProgramOpt
C[111][392]=-1801792042

real 0m0.403s
user 0m0.391s
sys 0m0.001s
```

The lines above are the result of running the two command lines, one unoptimized and the other optimized. In unoptimized trial the time taken was 0m1.074s, whereas in the optimized trial the time was 0m0.391s. This indicates that the optimized code ran about 2.7 times faster than the unoptimized code.