CIS 2107
Computer Systems and Low-Level Programming
Fall 2011
Midterm

October 25, 2011

Name: _____

Page	Points	Score
1	5	
2	10	
3	10	
4	7	
5	8	
6	15	
7	4	
8	7	
9	16	
10	18	
Total:	100	

Instructions

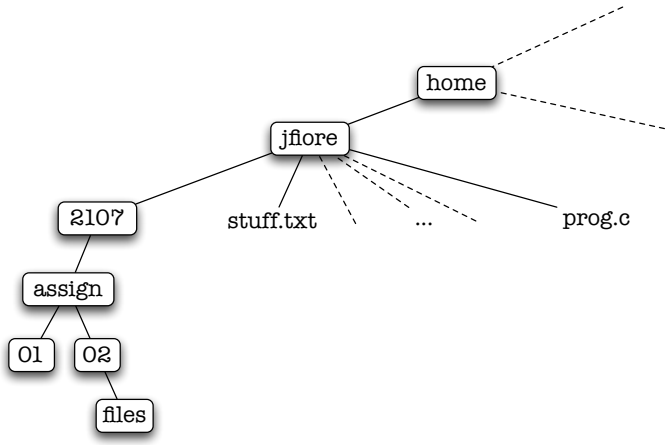
The exam is closed book, closed notes. You may *not* use a calculator, cell phone, etc.

For each of the questions of this quiz, you can assume the following sizes for C data types:

type	**bytes**
char	1
short	2
int	4
long	8
float	4
double	8
void*	4

For the following questions, you can assume that my home directory is the `jfiore` directory.

1. Unix shell stuff.



(1 point) (a) If I'm in my home directory *i.e.*, `/home/jfiore`, what command can I type in order to run the C compiler on `prog.c`, but not the linker?

(1 point) (b) If I'm in my home directory, what's the one command that I can type in order to create a `files` directory within the assignment 1 directory?

(1 point) (c) What command can I type to see a list of all of the files in my current directory?

(1 point) (d) If I'm in my home directory, what's the one command that I can type to move `prog.c` to the `files` directory inside assignment 2?

(1 point) (e) If I run the command `gcc -E prog.c` to run the preprocessor only on `prog.c`, what does the resulting file contain (*i.e.*, how is it different from `prog.c`)?

---

2. Some conversions.

(1 point) (a) 104 tbytes = ? kbits

(a) _____

(1 point) (b) 3 minutes = ? microseconds

(b) _____

(1 point) (c) 48 mbytes = ? tbytes

(c) _____

(1 point) (d) 128 mbytes = ? kbits

(d) _____

(1 point) (e) 1 hour = ? nanoseconds

(e) _____

3. Convert 246_{10} to:

(1 point) (a) base 2

(1 point) (b) base 16

4. Using the approximation trick that we talked about in class, about how much are each of the following?

(1 point) (a) 2^{31}

(a) _____

(1 point) (b) 2^{29}

(b) _____

(1 point) (c) 2^{43}

(c) _____

(2 points) 5. What is $111101011_2 + 11101110_2$ in base 2?

$$\begin{array}{cccccccccc} & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1_2 \\ + & & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0_2 \\ \hline \end{array}$$

(2 points) 6. What is $3967B7_{16} + 2E67_{16}$ in base 16?

$$\begin{array}{cccccccc} & 3 & 9 & 6 & 7 & B & 7_{16} \\ + & & & 2 & E & 6 & 7_{16} \\ \hline \end{array}$$

7. **data representation**. For these questions, please remember to answer in hex, not binary.

(1 point) (a) In hex, what is the smallest integer that can be represented by a 16-bit two's complement int?

(a) _____

(1 point) (b) In hex, what is the largest integer that can be represented by a 16-bit two's complement int?

(b) _____

(1 point) (c) In hex, what is the largest integer that can be represented by a 16-bit unsigned int?

(c) _____

(1 point) (d) In hex, what is -1 as a 16-bit two's complement int?

(d) _____

(2 points) 8. What is printed by the following?

```
char x = 50;
char signed_prod;
unsigned char unsigned_prod;

signed_prod = x*3;
unsigned_prod = x*3;

printf("%d\n", signed_prod);
printf("%u\n", unsigned_prod);        /* recall that %u means to print as unsigned */
```

---

points: _____
out of a possible 10

exam continues...

9. **Some bit operations.** If we have `char x = 0x53, y = 0xF9;`, what is the result of the following operations? Your answer must be in the form of exactly two hex digits[1].

(1 point) (a) x|y

(a) _____

(1 point) (b) x||y

(b) _____

(1 point) (c) x<<2

(c) _____

(1 point) (d) ~ x

(d) _____

(1 point) (e) x&0x0F

(e) _____

(1 point) (f) x^y

(f) _____

(1 point) (g) x&&1

(g) _____

[1]Ignore the possibility of promotion to 32-bit ints. Behave as though we're living in the land of 8-bit arithmetic.

(6 points) 10. For this question, we're doing 5-bit two's complement representation of integers. Fill in the empty boxes in the following table. Addition and subtraction should be performed based on the rules for 5-bit, two's complement arithmetic. Recall that in your book's notation, TMin is defined to be the smallest negative two's complement number that we can represent, and TMax is the largest positive one.

Name	**Decimal Rep.**	**Binary Rep.**
Zero	0	**0 0000**
n/a	6	
n/a	−5	
n/a		**1 1011**
n/a		**0 1010**
TMax		
TMin		
TMin+2		
TMax+2		

11. If I have the following:

```
int main(void)
{
  int a=10;
  int b=20;

  int *p=&b;
  int *q=p;

  (*p)++;
  q++;
```

and memory is laid out like this:

q	1000	
p	1004	
b	1008	
a	1012	

what do you see if you print:

(1 point) (a) a

(a) _____

(1 point) (b) &a

(b) _____

points: _____
out of a possible 8

question 11 continues...

(1 point) (c) b

(c) _____

(1 point) (d) &b

(d) _____

(1 point) (e) p

(e) _____

(1 point) (f) *p

(f) _____

(1 point) (g) &p

(g) _____

(1 point) (h) q

(h) _____

(1 point) (i) *q

(i) _____

(1 point) (j) &q

(j) _____

12. fun with floats

(3 points) (a) How would we represent the number 191.6875_{10} in fixed-point binary?

(1 point) (b) Normalize your answer from part (a).

(3 points) (c) How would 191.6875_{10} be stored in a C 32-bit `float` variable? (Remember that for 32-bit floats, the bias value is 127.)

13. **Recognizing the value of a floating-point variable.** In this question, consider 6-bit floating-point numbers. Two bits are used for the mantissa and 3 bits for the exponent.

(1 point) (a) For a 3-bit exponent field, what is the bias?

(3 points) (b) What floating-point value does the bit string 0 01 001 represent, where 0 is the sign bit, 01 is the mantissa and 001 is the exponent? Please show all work.

14. Use the following code to answer the questions.

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4
5  typedef struct {
6    int x;
7    int *p;
8    int A[3];
9  } Stuff;
10
11 void func01(int[]);
12 void func02(char*);
13 void func03(char*);
14 void func04(Stuff);
15
16 int main(void) {
17   int A[]={10,20,30};
18   int i=40;
19   Stuff s;
20   char msg[100];
21
22   s.x=50;
23   s.p=&i;
24   s.A[0]=60;
25
26   strcpy(msg, "aspirin");
27
28   func01(A);
29   func02(msg);
30   func03(msg);
31   func04(s);
32
33   return 0;
34 }
35
36 void func01(int A[]) {
37   A[0]++;
38 }
39
40 void func02(char *s) {
41   strcpy(s, "half way there");
42 }
43
44 void func03(char *s) {
45   s = malloc(10);
46   strcpy(s, "coffee");
47 }
48
49 void func04(Stuff s) {
50   s.x=5555;
51   *(s.p)=4040;
52   s.p=(int*)malloc(sizeof(int));
53   *(s.p)=4444;
54   s.A[0]=6666;
55 }
```

(1 point) (a) How many bytes are passed to the function `func01( )`?

(a) _____

(1 point) (b) How many bytes are passed to the function `func02( )`?

(b) _____

(1 point) (c) How many bytes are passed to the function `func04( )`?

(c) _____

What is the value of each of the following after `func04( )` has been called?

(1 point) (d) A[0]

(d) _____

(1 point) (e) i

(e) _____

(1 point) (f) s.x

(f) _____

(1 point) (g) *(s.p)

(g) _____

(1 point) (h) s.A[0]

(h) _____

(1 point) (i) msg (What's the string?)

(i) _____

(7 points) 15. Write a function which is passed an `int A[ ]` of positive integers and `A`'s length. The function returns the largest item in `A`. Do not use the `[ ]` operator.

16. big-endian/little-endian

(7 points) (a) Write a function called `is_big_endian( )` which returns 1 if the machine running the function is big endian or 0 otherwise.

(1 point) (b) If we're on a 32-bit little-endian machine and `int x` has the value `0x01234567`, and `x` is stored starting at address 1000, what is the value of the byte stored at address 1002?

(7 points) 17. Write a function `caps( )` which is passed a string `s`. The function returns a copy of `s`, but with the first letter of each word capitalized. If `s` is NULL or if an error is encountered, the function returns NULL. It is up to the caller to free any memory allocated by `caps( )`.

(10 points) 18. Write a program which reads text from STDIN until EOF is entered. The program prints the length of the longest word and the longest line in the file. (Be careful. You don't need to print the longest line and longest word – just their length.)

(extra space)