CIS 2107
Computer Systems and Low-Level Programming
Fall 2010
Midterm

October 21, 2010

Name: ____________________________

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:

<table>
<thead>
<tr>
<th>type</th>
<th>bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>1</td>
</tr>
<tr>
<td>short</td>
<td>2</td>
</tr>
<tr>
<td>int</td>
<td>4</td>
</tr>
<tr>
<td>long</td>
<td>8</td>
</tr>
<tr>
<td>float</td>
<td>4</td>
</tr>
<tr>
<td>double</td>
<td>8</td>
</tr>
<tr>
<td>void*</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Page</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
For the following questions, you can assume that my home directory is the \texttt{jfiore} directory.

1. Unix shell stuff.

(a) If I’m in my home directory, and I run the command \texttt{gcc -c prog.c}, what is the name of the resulting file and what does it contain?

(b) If I run the command \texttt{gcc -E prog.c} to run the preprocessor only on \texttt{prog.c}, what does the resulting file contain \textit{(i.e., how is it different from \texttt{prog.c})}?

(c) What’s the command that allows me to see what’s in \texttt{prog.c} one page at a time?

(d) If I’m in my home directory \textit{i.e., /home/jfiore}, what’s the single command that I can type in order to remove the assignment 2 directory?

(e) What does the \texttt{ar} command do?

2. Why is it that when we declare a struct, functions, \textit{etc} inside a .h file, we enclose it in a \texttt{#ifndef #define #endif} block?

3. We’ve described a computer’s storage hierarchy as a kind of pyramid. What two things are true as we go up the pyramid?

4. (a) 72 mbytes = ? tbits

(b) 8 bytes = ? kbits

(c) 5 minutes = ? microseconds
(1 point) (d) 152 gbits = ? gbytes

(d) ___________

(1 point) (e) 160 tbits = ? kbits

(e) ___________

Convert each of the following from base 10 to base 2 and base 16

(2 points) 5. 196

(2 points) 6. 181

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

(1 point) (a) $2^{37}$

(a) ___________

(1 point) (b) $2^{22}$

(b) ___________

(1 point) (c) $2^{48}$

(c) ___________

(2 points) 8. What is 1011110₂ + 111111₂ in base 2?

\[
\begin{array}{cccccccc}
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0_2 \\
+ & 1 & 1 & 1 & 1 & 1 & 1 & 1_2 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1_2 \\
\end{array}
\]

(2 points) 9. What is B9F₆₁₆ + 359AC₁₆ in base 16?

\[
\begin{array}{ccccccc}
B & 9 & F & 6 & 4_{16} \\
+ & 3 & 5 & 9 & A_{16} \\
\hline
\end{array}
\]

points: _______  
out of a possible 9  
2 of 9  
exam continues...
10. In hex, what is the smallest integer that can be represented by an 8-bit two’s complement int?

10. ____________

11. In hex, what is the largest integer that can be represented by an 8-bit two’s complement int?

11. ____________

12. In hex, what is the largest integer that can be represented by an 8-bit unsigned int?

12. ____________

13. In hex, what is −1 as an 8-bit two’s complement int?

13. ____________

14. What is printed by the following?

```c
unsigned char x = 255;
x += 3;
printf("x=%d\n", x);
```

14. ____________

15. Some bit operations. If we have `char i = 0xC7, j = 0x9F;`, what is the result of the following operations? Your answer must be in the form of exactly two hex digits\(^1\).

(a) \(\sim i\)

(a) ____________

(b) \(!i\)

(b) ____________

(c) \(i \& 0xF0\)

(c) ____________

(d) \(i \wedge j \wedge i\)

(d) ____________

---

\(^1\)Ignore the possibility of promotion to 32-bit ints. Behave as though we’re living in the land of 8-bit arithmetic.
(1 point) (e) i|j

(e) __________

(1 point) (f) i<<2

(f) __________

(1 point) (g) i|0

(g) __________

(2 points) 16. If we’re on a little-endian machine, what is printed by the following code?

```c
int x=0x98765432;
char *p=(char*)&x;
int i;
for (i=0; i<sizeof(x); i++, p++)
    printf("%x\n", (unsigned char)*p);
```

(6 points) 17. 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.
18. If I have the following:

```c
int main(void)
{
    int a=10;
    int b=20;
    int *p=&a;
    char *cp=(char*)&a;

    ...
    (*p)++;
    cp+=2;
    p+=2;
}
```

and memory is laid out like this:

```
  a  1000
  b  1004
  p  1008
  q  1012
  cp 1016
```

what do you see if you print:

(1 point)  (a) a
(a) ___________

(1 point)  (b) &a
(b) ___________

(1 point)  (c) b
(c) ___________

(1 point)  (d) &b
(d) ___________

points: _______
out of a possible 11
(1 point) (e) p

(e) ____________

(1 point) (f) *p

(f) ____________

(1 point) (g) &p

(g) ____________

(1 point) (h) cp

(h) ____________

19. Use the following code to answer the questions.

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

struct Poo {
    int x;
    int A[1];
    char *s;
};

void func(struct Poo *p);

int main(void) {
    struct Poo p;
    char *str;
    str=(char*)malloc(10);
    strcpy(str, "hi");
    p.x=10;
    p.A[0]=20;
    p.s=str;
    func(&p);
    printf("p.x=%d\n", p.x);
    printf("p.A[0]=%d\n", p.A[0]);
    printf("p.s=%s\n", p.s);
    return 0;
}

void func(struct Poo *p) {
    p->x=111;
    p->A[0]=222;
    strcpy(p->s, "bye");
    p=(struct Poo*)malloc(sizeof(struct Poo));
    p->x=1111;
    p->A[0]=2222;
    p->s=(char*)malloc(10);
    strcpy(p->s, "later");
}
```

(a) How many bytes are passed to the function func( )?

(a) ____________

What is the value of each of the following after func( ) is called?

(b) (2 points) p.x

(b) ____________

(c) (2 points) p.A[0]

(c) ____________

(d) (2 points) p.s

(d) ____________

points: ________

out of a possible 1

6 of 9

exam continues...
20. Write a function which takes as an argument an int x. The function returns an int which has the bytes of x in reverse order. Do not assume that ints are 4-byte values.

21. Write a function which will swap the pointers p and q. To receive any credit, it must be a function, and when the function has finished, p points to b and q points to a.

```c
int a=10;
int b=20;

int *p=&a;
int *q=&b;
```
22. Write a C function which is passed a C-string s, a C-string sep, an int multiplier m. The function returns a new string which is the original string duplicated m times, separated by sep. For example, if s="bora", sep=" ", and m=2, the function returns the string "bora bora". The caller is responsible for freeing the memory allocated by your function.

23. Write a function which is passed a C string. The function capitalizes the first letter of every word in the string.