**You have the following code in your C program**

**char \*x = “hello\n”;**

**char x1[] = “hello\n”;**

**Which segments are the values of x and x1 stored in memory, respectively?**

\*x stores in read-only-memory

X[] stores in stack

Two points for each colored sentence

**Given an integer A=0x11223344, write its four bytes in the correct order in the memory according to the little endian system, assuming the memory address increases from the left to the right**

Ans: 0x44 0x33 0x22 0x11

**You defined two buffers in your code**

**buf1[8] = {‘a’,’a’,’a’,’a’,’a’,’a’,’a’,’a’};**

**buf2[4] = {‘b’,’b’,’b’,’b’}**

**Now, write a code to copy buf2 to buf1 and make buf1 as {‘a’,’a’,’b’,’b’,’b’,’b’,’a’,’a’} using memcpy function.**

memcpy(&buf1[2], buf2,4)

**Why it is a good practice to set a pointer to NULL after freeing it?**

Otherwise, we will get a dangling pointer

Give partial score if their answer is close to this answer. For instance they forgot the “dangling pointer” term but could tell the actual consequences

**Write a code to open a file using fopen() for only reading purpose?**

FILE \*file = fopen( filename, "r+" );

**What is the command to set a breakpoint in line 10 in the gdb mode?**

Ans: b 10 (or break 10)

**Convert the decimal number 200 into hexadecimal, and then directly convert the hexadecimal representation into binary representation. Show your work.**

Ans:

The algorithm for reference

Graphical user interface, text, application

Description automatically generated

(3pts) 200 = 16\*12+8 -> 0xC8

Text

Description automatically generated

(3pts) 0xC8 -> 11001000 (students need to use the above idea. If they directly convert the decimal into binary and get the correct answer, give 1pt only)

**What is the purpose of fflush(FILE \*stream). If the stream argument is NULL then what will happen?**

File\* based write are buffered, they may be data written, but not yet pushed to the OS. fflush() forces a write of all buffered data

If the stream argument is NULL, fflush() flushes all open output streams

4 points for each of above colored answer.

**Given the decimal number 40, whose binary representation is (00101000), in an 8-bit system, what is the binary representation of -40 according to two’s complement number system? Show your work.**

Using two's complement,

11010111 (invert bits) --> 11010111 + 1 (add 1) = 11011000 --> (-40)

**Suppose you have a header file named mymath.h, which includes a function declaration “void foo()”. Add conditional compilation directives (such as #define, #if, #ifdef, #ifndef,#endif) into mymath.h to make sure the function declaration will only happen once in your project when mymath.h is included in multiple source files. (write your code before or after the following code).**

#ifndef \_MY\_MATH\_

#define \_MY\_MATH\_

void foo();

#endif

(students may use another name instead of \_MY\_MATH\_)

To get perfect score, students to use these directives correctly, not only the names, but also the correct order. 2 pts for each.

**How can you implement a function similar to calloc using malloc and memset?**

This is the actual implementation of calloc. A student doesn’t need to be syntactically correct in this answer. But they need to be logically correct. I want to know whether they understand the functions of malloc and memset and the definition of calloc.

void \*calloc(size\_t n, size\_t size)

{

size\_t total = n \* size;

void \*p = malloc(total);

if (!p) return NULL;

return memset(p, 0, total);

}

If students define size for malloc, return NULL if !p is true and use malloc and memset properly then give full points.

If they don’t return NULL if !p is true then reduce one point

Don’t need to reduce point if they don’t define size for malloc

If the student explain in sentence not pseudocode or actual C code but they are logically correct reduce one points

The following questions are all concerning Cache.

**(a) Name the two types of cache locality and briefly explain the intuitions behind them (one sentence for each).**

* + Spatial locality: data to be accessed tend to be close to data you already accessed
  + Temporal (or time) locality: data that is accessed is likely to be accessed again soon

(**b) The L2 cache of Intel core I5 is claimed to be 8-way set associative. What does it mean?**

(This is a cache placement policy), it means each new block from the main memory (or level k+1 cache) may be placed in one of 8 cache lines (or cache blocks) in the cache (or level k cache).

If students say “This is a cache placement policy” (2pts).

Otherwise, if the following sentence is correct, they can still get the perfect score.

**(c) Assume a memory access to main memory on a cache "miss" takes 100 ns and a memory access to the cache on a cache "hit" takes 5 ns. If 80% of the processor's memory requests result in a cache "hit", what is the average memory access time?**

5+(1-80%) \* 100 = 25ns