Please read the instructions and questions carefully. You will be graded for clarity and correctness. You have 50 minutes to complete this exam, so focus on those questions whose subject matter you know well. This is a closed book exam. Write legibly and check your answers before handing it in.

**Short Answer - some will be one or two words – no more than 1 sentence. (50/100 points)**

1. (5pts) What does the command “up” do in gdb?

2. (5pts) What is a protection system in an operating system?

3. (5pts) Given the following structure:
   ```c
   typedef union {
     uint32_t a;
     uint64_t b;
     char c;
   } MYVALUE;
   ```
   What does the function `sizeof(MYVALUE)` return?

4. (5pts) What is a functional prototype?

5. (5pts) What utility sets up a program to run?

6. (5pts) Define a function `passBy` signature that accepts an integer `a` by value and another integer `b` by reference and returns an integer.
7. (5pts) Call the function passBy using c and d (as a and b, respectively, if the are defined int c, d;

8. (5pts) Name one reason memory mapping is better than sbrk/brk for implementing malloc.

9. (5pts) What is the difference between the network and transport layer?

10. (5pts) What are the three things you need to know to setup a connection?

Long Answer - no more than 4 sentences (20/100 points)

11. (10pts) A process opens two files "one.txt" for reading and "two.txt" for writing. Draw the file descriptor table for this process assuming the operating system assigns handles sequentially from zero.

12. (10pts) Assume you have an array of double values of size 10 and a character pointer char *x pointing to the address of the first (zeroth) value of the array. How would you index the 7th double value without using array indexing in the following statement?

    printf( "Value %f", ??? );
13. (10pts) Assume you have a data file whose contents are formatted:

10, Ford, Red
20, Chevy, Blue
30, Toyota, Yellow
40, Honda, White

(a) Define a type for a data structure that will contain this information.

(b) Create a function that receives string containing one line, parses the information into fields, dynamically allocates the structures, fills the data into it, and returns the value.
int func( int a ) {
    if ( a >= 32 ) {
        printf( "%d : OUT A\n", a );
        return( 0 );
    }
    if ( (a%21) == 4 ) {
        printf( "%d : OUT B\n", a );
        return( 0 );
    }
    if ( ( a & (4 >> 2) ) ) {
        printf( "%d : OUT C\n", a );
        return( 0 );
    }
    if ( (a & 0x16) & 0x10 ) {
        printf( "%d : OUT D\n", a );
        return( 0 );
    }
    printf( "%d : OUT E\n", a );
    return( 0 );
}

Figure 1: Function func

14. (10pts) Consider the function in Figure 1. Provide an integer input value \( a \) for each of the following outputs and explain why the associated condition is successful, if any.

(a) OUT A : \( a = \)

(b) OUT B : \( a = \)

(c) OUT C : \( a = \)

(d) OUT D : \( a = \)

(e) OUT E : \( a = \)
15. (10pts) Write a function `thread_main` that will serve as the entry point to a thread. This function should print its thread ID to standard out and then return.

Write a one-line C statement to create the thread using `thread_main`. 