Section Merge: CMPSC311-SP22 - Intro to Systems Programming Midterm 1

Shi Qiu

TOTAL POINTS

32.5 / 100

QUESTION 1

1 1.a **2 / 3**

- **0 pts** Correct
- ✓ **- 1 pts** **Partially correct**
- **3 pts** Incorrect

QUESTION 2

2 1.b **0 / 3**

- **0 pts** Correct
- ✓ **- 3 pts** **Incorrect**
- **0.5 pts** Partially Correct
- **1.5 pts** Partially Correct

QUESTION 3

3 1.c **0 / 3**

- **0 pts** Correct
- ✓ **- 3 pts** **Incorrect**
- **1.5 pts** Partially Correct
- **2 pts** mostly incorrect

QUESTION 4

4 1.d **0 / 3**

- **0 pts** correct.
- ✓ **- 3 pts** **incorrect**
- **1.5 pts** not answered when it is created
- **1.5 pts** Partially correct
- **3 pts** not answered

QUESTION 5

5 1.e **0 / 3**

- **0 pts** Correct
- **3 pts** No answer
- ✓ **- 3 pts** **Executable run the code, object file is created after compiling**
- **1.5 pts** Object files are non runnable and created after compilation
- **1.5 pts** Executable is runnable code

QUESTION 6

6 1.f **1.5 / 3**

- **0 pts** Correct
- **3 pts** Incorrect, did not mention that C doesn't have bounds checking.
- **3 pts** Not answered
- ✓ **- 1.5 pts** **Partially correct**

QUESTION 7

7 1.g **3 / 3**

- **3 pts** Incorrect - Missing keyword address
- **1 pts** The value being passes is address of the memory location of the variable a using pointer. Missing necessary keywords.
- **3 pts** Incorrect
- ✓ **- 0 pts** **Correct**
- **3 pts** Did not answer What instead answered Why
- **3 pts** Unattended

QUESTION 8

8 1.h **0 / 3**

- **0 pts** Correct
- **1.5 pts** Partially correct.
- ✓ **- 3 pts** **Incorrect, null character identifies end of string**
- **3 pts** No answer
- **2 pts** Mostly incorrect

QUESTION 9

9 1.i **0 / 3**

- **0 pts** Correct

✓ **- 3 pts** **Incorrect**

- 1.5 pts Partial

- 3 pts Not answered

QUESTION 10

10 1.j **3 / 3**

✓ **- 0 pts** **Correct**

- 0 pts Missing 0x in each number

- 1.5 pts Incorrect form/ partially correct

- 3 pts Incorrect order

QUESTION 11

11 1.k **0 / 3**

- 0 pts Correct

✓ **- 3 pts** **Incorrect Answer**

- 2 pts Partially Correct

- 1 pts It's the data segment.

Data Segment. or Bss

QUESTION 12

12 1.l **3 / 3**

✓ **- 0 pts** **Correct command to set breakpoint (`b` or `break`) and correct specification of break location (`10` or `filename:10`)**

- 1 pts Correct command to set breakpoint (`b` or `break`) and incorrect specification of break location (not `10` or `filename:10`)

- 2 pts Incorrect command to set breakpoint (not `b` or `break`) and correct specification of break location (`10` or `filename:10`)

- 3 pts Incorrect command to set breakpoint (not `b` or `break`) and incorrect specification of break location (not `10` or `filename:10`)

QUESTION 13

13 2.a **2 / 6**

- 0 pts Correct

- 1 pts Mostly Correct

- 3 pts Partially Correct

✓ **- 4 pts** **Not a clear and correct explanation**

- 6 pts Totally incorrect

QUESTION 14

14 2.b **0 / 6**

- 0 pts Correct explanation of abstraction emphasizing on simplify/easier/less complex/more user-friendly, and relevant example.

✓ **- 3 pts** **Incorrect or missing explanation of abstraction**

✓ **- 3 pts** **Irrelevant, incorrect or missing example**

- 1.5 pts Partially incorrect explanation of abstraction (the emphasis is not on simplify/less complex/easier/less complex/more user-friendly)

QUESTION 15

15 2.c **0 / 6**

- 0 pts Correct

- 3 pts Missing direct interaction with hardware in privileged mode

- 3 pts No mention of security and safety

✓ **- 6 pts** **Incorrect**

QUESTION 16

16 2.d **0 / 6**

- 0 pts Correct

- 2 pts Partially correct. Either command is incorrect or definition.

- 3 pts Click here to replace this description.

- 5 pts Mostly incorrect.

✓ **- 6 pts** **Incorrect**

- 4 pts Very partially correct.

- 1 pts Mostly correct.

Piping contacts two commands in shell. First command's output is used as input for second command

(The Unix systems allow stdout of a command to be connected to stdin of another command. You can make it do so by using the pipe character 'l')

QUESTION 17

17 2.e **0 / 6**

- **0 pts** Correct

✓ **- 6 pts C has no garbage collection and if memory is not freed will lead to memory leaks**

- **3 pts** C need to free allocated memory manually otherwise will lead to memory leaks
- **3 pts** Java has garbage collector
- **6 pts** No answer
- **4 pts** Dynamic memory is in the heap not the stack
- **3 pts** Partially correct only the garbage collection for Java and memory allocation in C part

QUESTION 18

18 2.f **6 / 6**

✓ **- 0 pts Correct**

- **3 pts** Wrong Hex Conversion
- **3 pts** Wrong Binary Conversion
- **4 pts** Incorrect Procedure
- **4 pts** No Procedure shown
- **1 pts** Correctly converting from a wrong hex to binary

QUESTION 19

19 2.g **6 / 6**

- **6 pts** Unattended
- **3 pts** Missing step 2 to add 1 to the compliment
- **3 pts** Incorrect step 2 to add 1 to the compliment
- **6 pts** Incorrect

✓ **- 0 pts Correct**

- **2 pts** Incorrect representation of binary number 30
- **2 pts** Minor mistakes but complete understanding of the concept

QUESTION 20

20 3.1(a) **1 / 4**

- **0 pts** Correct
- **1.5 pts** one array not identified

✓ **- 3 pts Both arrays not identified**

- **1 pts** integer not identified
- **4 pts** wrong

QUESTION 21

21 3.1(b) **0 / 4**

- **0 pts** Correct
- **2 pts** (Why) is wrong

✓ **- 4 pts Incorrect**

- **2 pts** value wrong

QUESTION 22

22 3.2(a) **1.5 / 4**

- **0 pts** Correct

✓ **- 0.5 pts Missing semicolon `;` at the end**

✓ **- 1 pts Should be `char stu_name[32];`**

- **1 pts** Should be `uint32_t stu_id;`

✓ **- 1 pts Should be` unsigned int stu_id;`**

- **1 pts** Missing all the semicolons `;`
- **1 pts** Missing brackets `{}`
- **1 pts** Should be `Struct StudentData{ };`
- **1.5 pts** The struct should be `struct StudentData{ };`
- **1.5 pts** Partially Correct
- **3 pts** Partially Correct
- **3.5 pts** Incorrect
- **4 pts** Incorrect

QUESTION 23

23 3.2(b) **1.5 / 4**

- **0 pts** Correct
- **1 pts** Partially correct
- **0.5 pts** _int_ is unnecessary
- **0.5 pts** Missing `,` inside of the enum type

✓ **- 0.5 pts Missing `; `at the end of `enum{}`**

- **0.5 pts** Should be `,` instead of `;` inside of the enum type
- **2 pts** Should be `enum{ ... }stu_sec;`

✓ **- 2 pts Should be `{SECTION1 =1, ...}`**

- **3 pts** Incorrect
- **3.5 pts** Incorrect
- **4 pts** Incorrect

QUESTION 24

24 3.2(c) **2 / 6**

✓ **- 1 pts incorrectly initialized p with the address of alice**

- **2 pts** incorrectly initialized the section id of alice with SECTION2 through the pointer p.

- **1 pts** "." instead of "->"

✓ **- 3 pts part 3 -> incorrect**

- **6 pts** unanswered
- **0 pts** correct
- **2 pts** incorrectly initialized p and alice
- **1 pts** incorrectly initialized p

Name: Shi Qin

PSU ID (e.g.,abc123): sbq 5043

CMPSC 311 Exam 1

February 14, 2022

Closed book, closed neighbor, no electronic tools or additional papers.

You may not share or discuss exam questions with anyone.

1. Short Questions (36pts total, 3pts each, be brief)

a. What is swap space?

Where you can swap with address

b. What is the potential address space of a system which uses 32-bit word size?

$0x000000$ $\frac{32}{4} = 8$

c. What is the purpose of device drivers?

to compile program language into computer language

d. When a stack frame is created and when it is destroyed?

First time process it
and
after the program is terminated

e. How an executable file is different than an object file in the C compilation process?

An executable file is different from an object file in C compilation in the way they are also, for example you will need main beside other functions.

f. Why C compiler doesn't give any error in the following C code?

```
char foo[20];
foo[20] = 'a';
```

C does not know the length of a array

g. What does a caller pass to a callee in a *pass by reference* technique?

define veriable by address

h. Why does a C String need a NULL character?

some time you want to return null.

i. What is the main difference between strongly typed language and weakly typed language?

Strongly typed language like java and weakly typed language like c give you different level of control to memory and etc.

j. 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.

44 33 22 11 0x

0x00

k. Which section of the process space stores global variables?

outer, whole

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

b 10

2. Medium Questions (42pts total, 6pts each)

a. What is the difference between Typ1 1 and Type 2 hypervisors?

Typ1 and type2 hypervisors are different in the way they process codes.

b. Explain the term *abstraction* with one example.

abstraction is not directly referring to the value, but to refer by address ;

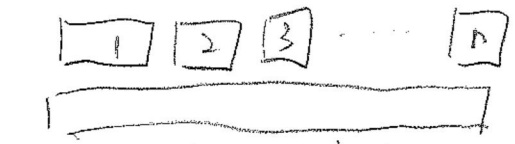
```
int a=5
int *b=a;
```

*b is memory address.

c. Why does the CPU need to be in privileged mode while OS is doing any task? Explain.

So it does not be, interuped by other task, if ~~two~~ task use same memory, errors may occur.

d. What is piping in the bash shell? Write a shell command using pipe(s).



store from 1 to 2 to 3 .... to N, one by one

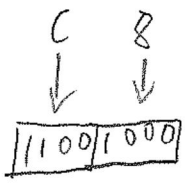
```
def add2(int c)   def add1(int c)   def main()
  c=c+1;            c=c+1;            c=2
  return c;         return            c=add1(c);
                                      c=add2(c);
```

e. Why a C program is vulnerable to memory leak bugs but a Java program is not? Explain.

for example, array in C and Java are different array in C does not know it's own length, while in Java, it knows, so when you access a large number over C's array's number, your will meet memory leak bugs. you have more control in memory.

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

16 | 200 | 12 → C
   | 192 | 8 → 8
1 | 8 |



g. Given the decimal number 30, whose binary representation is (00011110), in an 8-bit system, what is the binary representation of -30 according to two's complement number system? Show your work.

128 64 32 16 8 4 2 1

11100010

00011110

11100001

+1

11100010

$-128+(4+32+2=-30$

3. Programming Questions (22pts total)

(1) a. [4pts] Based on the code snippet, finish the function prototype for the *foo* function, clearly showing the type of each argument.

```
void foo( int &a, int b, grid c );
....

int main() {
int *a, b=0, c[5]={[0,2]=4};
a = c;
foo((float *)&a[0], b, c);
}
```

b. [4 pts] In the above question, what is the value of *(a+4)? why?

the current address of [c + 4*4 (16)]

(2) a. [4 pts] Write code which creates a struct type called StudentData to represent the registration data of each student in our class. The structure has two fields: the name field *stu_name* that uses (up to) 32 characters, and the id field *stu_id* that is an 32-bit unsigned integer.

```
struct {
    char stu_name;
    Unit_8 stu_id;
} StudentData
```

b. [4 pts] In the above question, let us assume the struct type in addition has the third field variable *stu_sec* to represent the student section id, which is of an enum type with three section names SECTION1 (representing 1), SECTION2 (representing 2) and SECTION3(representing 3) as its member fields. Write the code to correctly declare the enum type.

```
enum {
    SECTION1
    SECTION2
    SECTION3
} stu_sec.
```

c. [6 pts] Continue with the above question. Write code to (1) declare a variable *alice* and a pointer *p* of this struct type; (2) initialize *p* with the address of *alice;* (3) initialize the section id of *alice with* SECTION2 through the pointer *p*.

```
deftype alice StudentData;
deftype *p;
*p = alice;
*p. stu_sec = "1";
```