Name:__________________________________

PSU User ID (e.g.,abc123, not 9 digit numbers): ________________________________

CMPSC 311 Exam 2

April 6, 2022

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

You may not share or discuss exam questions with anyone.

1. **Short Questions (36 pts total, 4pts each, be brief)**

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

      Otherwise, we will get a dangling pointer

      1 pt off if their answer is close to this answer. For instance they forgot the "dangling pointer" term but could tell the actual consequences

   b. What are file descriptors?

      A file descriptor is an index assigned by the kernel into a table of file information maintained in the OS
      File descriptors are used to reference a file when call the I/O system calls
      The kernel accesses the file for the process using the file descriptor
      If student's answer is similar to any of the above colored sentence, then give full points

   c. What is the reason that buffer is often defined as void* ?

      This is a general purpose pointer (point to a raw memory address), which allows the buffer to hold any specific data type (can be typcasted into the intended type later).

      If students answer like red colored sentence, please give 3 points. If he/she adds something similar to the green part, give 1 more point. The blue part is optional.

   d. What is a unix signal?

      A signal is a special message sent through the OS to tell a process (or thread) of some command or event

      (1pt for each colored item)

   e. Write the command to show all the running processes of user Alice.

      ps –U Alice

      if students forget to add the –U option, 1 pt off.

   f. Write a code to open a file using `fopen()` for only reading purpose?

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

1 pt for each colored item

g. 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)
```

1 pt for each of memcpy, &buf1[2](or buf1+2), buf2 and 4.

h. 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

i. What’s the difference between a (char *) and a (char **)?

(char *) is pointer and (char**) is double pointers
Two points for each color

2. Medium Questions (48pts total, 6pts each)

a. Assume that a file has the following access policy

```
rwxrw---x
```

Explain the above access policy for the file.

Owner has read, write, and execute permission
Group has read and write permission
World (or the other) has only execute permission

We see there are access policies for three categories. Please reduce two points for the wrong answer for each category

b. 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

3 points for each of above colored answer.

c. Explain blocking I/O, Non-blocking I/O and Asynchronous I/O.

Blocking: The call waits for the read or write to complete before returning
Non-blocking: The call does not wait for the read or write to complete before returning
Asynchronous: I/O request returns immediately. A callback is generated when I/O completes

2 points for each colored sentence.

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

Something like below. 2 pts for each line of code
void *p=malloc(len);

```
memset(p, 0, len);
```

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

e. Write the unix command with SIGKILL(or its signal number) to kill a process with pid 1000. Further, name one advantage of SIGTERM over SIGKILL.

kill -9 (or –SIGKILL) 1000   (3pts)

SIGTERM allows graceful shutdown of the process (it is okay if students say something more, such as allowing a signal handler to shut down gracefully, cleanup memory, close files, etc.)   (3pts)

f. One difference between static library and dynamic library is that, when you call a library function in your code, the function in your object code is resolved at _(1)__ time and at __(2)_ time, respectively. To build a static library and a dynamic library, you use the command with name ___(3)__and ____(4)___, respectively, and the resulting library has the file extension __(5)__ and __(6)__, respectively. (one word for each blank).

(1). link (or linking, or compilation)   (2). run (or load)

(3). ar   (4). gcc

(5). .a (or a)   (6). .so (or so)

1 pt for each space.

g. 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_
```

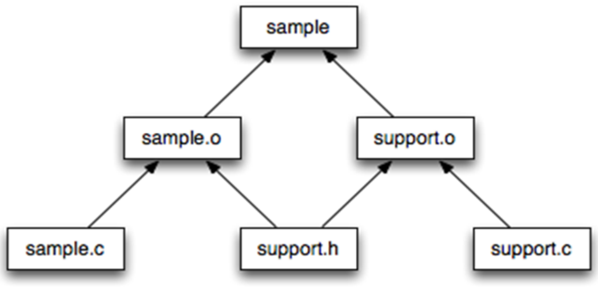
```
#define _MY_MATH_

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.

h. Given the following dependency graph, write your makefile rules. (you need to pay attention to spacing to get a perfect score).



```
sample : sample.o support.o
  gcc sample.o support.o -o sample

sample.o : sample.c support.h
  gcc -c -Wall -I. sample.c -o sample.o

support.o : support.c support.h
  gcc -c -Wall -I. support.c -o support.o
```

2pts for each rule. There is no need to apply the–Wall –I. options though. Since we do not assume explicitly that the support.c and sample.c include support.h, you will need to write support.h in the rules (otherwise 1 pt off in total) .If you do not include it, you need to explicitly write the assumption.

3. Long Questions (16pts total)

The following questions are all concerning Cache.

(a) (4pts) 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) (4pts) 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) (4pts) If the cache adopts the LFU policy, what does it mean?

This is a cache replacement policy (2pts). It means the least frequently accessed item (or cache line) in the cache will be evicted when a new block is brought into a full cache.

If students say “This is a cache replacement policy” (2pts).
Otherwise, if the following sentence is correct, they can still get the perfect score.

(d) (4pts) 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