

1> Final exam format. Conflict exam? cs241admin

12/13/2016 Tue 1:30 PM

2> Scoring appeals/attendance points in by Wed 11:59pm

3> curl http://illinois.edu and press return. What happens?

4> Traversing inodes and directories.

A program executes

```
FILE *f = fopen("/etc/hosts", "r");
```

The o/s first read the directory listing at /

Next it reads the directory listing at /etc

Then get the first byte of the file /etc/hosts?

How many disk blocks in total are read? Assume the only mounted directory is at the root directory. Assume all inode data is cached but directory listings are not cached.

5> Add one character to fix the mistake in the given code.

```
char* result = NULL;
myconcat( result, "CS", "241");
puts(result); // prints "CS241" and a newline
```

then implement `myconcat` cat using `sprintf` to copy the strings.

Challenge: How would replace `sprintf` and implement `myconcat` using only `strcpy(char*dest,char*src)` to copy the strings?

Challenge: How would replace `sprintf` and implement `myconcat` using only `strcat(char*dest,char*src)` to copy the strings?

6> Producer Consumer

Implement a fixed capacity, multithreaded producer consumer. Do not allow more than 100 items to be in the queue (or call `remove_raw` on an empty queue). Use two counting semaphores and a mutex. Why is the mutex necessary? State the initial values of the semaphores.

```
void add(void* value) { // Blocks if 100 items are in the
    // queue. Call add_raw(void*) to enqueue.
```

```
void* remove() { // Blocks while the queue is empty.
    // Call void* remove_raw() to dequeue.
```

7> Working with file metadata

What is the value of `result.st_mode` and `result.st_size` if the file "abc" does not exist? Complete the code to only print output when "abc" is a directory.

```
    struct stat result;
    stat("abc", &result);

    if( S_ISDIR(_____) )
        puts("Is a dir!");
```

8. Fix the following code. Hint there's at least two errors.

```
struct stat result;

stat(path, &result);

if( S_ISLNK( result ) ) {
    printf("%s is a symbolic link", path);
}
```

9. If you keep calling `opendir` and never call `closedir` what will you run out of?