

1> Final exam date and format.

2> Scoring appeals/attendance points? – do it now!

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> Docker Images & Linux Containers

#### *Example dockerfile*

```
FROM python:3.8.15-slim-buster

RUN apt-get update && apt-get install -y curl ...

COPY ./stuff/requirements.txt requirements.txt
RUN pip install --no-cache-dir -r requirements.txt
COPY ./stuff .
CMD [ "nice", "-n", "18", "ionice", "-c", "2", "-n", "6",
      "python3", "-u", "/server.py" ]
```

#### *Example docker-compose.yml*

```
db:
  image: postgres:11.7
  volumes:
    - "${DATA:-
~/docker_data}/pgvolume:/var/lib/postgresql/data"
  ports:
    - "5432:5432"
  env_file:
    - ".env"
  environment:
    - POSTGRES_USER=${ADMIN_USER_ID:-guest}
    - POSTGRES_PASSWORD=${ADMIN_PASSWORD:-guest}
  container_name: "db"
```

```
To test:          docker-compose build
To run :          docker-compose up -d
On github actions: docker-compose push
On server:        docker-compose pull
```

## 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? Fix and 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?