CS 4431
Final Exam (MAX = 100 points)
OPEN BOOK AND NOTES
CONCISE BULLETS GET MORE CREDIT THAN WORDY SENTENCES
LEGIBLE WRITING IS A REQUIREMENT NOT AN OPTION!

1. (1 pt, 1 min) Spell the name of the Serbian leader who is being a thorn in resolving the conflict in Kosovo.

2. (9 pts, 5 min) Explain how IPC may be implemented at the user level without compromising the authenticity of the communicating threads, and the integrity of the data being communicated between the threads during call and return.

3. (10 pts, 5 min) (*Answer True or False with justification*). In bus-based shared memory multiprocessors, a spinlock algorithm that employs static assignment of delay slots to spinning processors tends to perform better than one which employs dynamic assignment of delay slots.

4. (10 pts, 5 min) Explain the concept of *false sharing* with an example.

5. (10 pts, 5 min) Explain the error message that you may have seen several times in using the Unix system: "Stale NFS handle."

6. (10 pts, 5 min) Consider the following execution in a distributed shared memory system:

```
Process 1                                   Process 2
---------                                   ---------
S1: R(X):1                                  S3: R(Y):1
S2: W(Y):1                                  S4: W(X):1
```

X and Y are shared memory variables, that are both initialized to zero. R(X):1 signifies that the read of variable X returned a value 1. Similarly W(X):1 signifies that the process wrote a value 1 to the location X.

Is the execution as shown sequentially consistent? If yes, show an order of execution that would result in the values shown. If not, explain why not.

7. (30 pts, 30 min) Consider three processes P1, P2, and P3 manipulating a shared file f. Assume the following:

- the processes are executing on distinct workstations in the Sprite distributed system,
- **ALL** the client workstations have f locally encached prior to the start of the following execution,
- all writes append to the current end of file.

```
                P1                    P2                      P3
      -------------------------------------------------------|
      |
    5 |------ open(f,r)
      |         .
   10 |------------------------ open(f,r)
      |         .
   15 |------ close(f)            .                      .
      |         .
   20 |--------------------------------------------- open(f, w)
      |         .                 .
   25 |--------------------------------------------- write(f)
      |         .                 .
   30 |------------------------ read(f)
      |         .                 .
   32 |---------------------------------------------close(f)
      |         .                 .
   35 |------------------------ close(f)
      |         .                 .
   40 |------ open(f,w)
      |         .
   42 |------ read(f)
      |         .
   45 |------ write(f)
      |         .
   50 |------ close(f)            .                      .
      |                           .                      .
   60 |--------------------------------------------- open(f, r)
      |         .                 .
   65 |--------------------------------------------- read(f)
      |         .                 .
   70 |---------------------------------------------close(f)
      |         .                 .
      |
      V
    time
    (secs)
```

Explain what interactions take place between the clients and the server with respect to the above execution. Let f_{old} be the original version of the file; and let f_3 and f_1 be the two versions of the files after the writes by P3 and P1, respectively. Clearly indicate what version of the file is obtained by each read operation.

Additional Workspace:

8. (20 pts, 30 min) For this question ssume the following (with respect to the RPC implementation we studied in Birrel and Nelson paper):

- RPC retransmission timeout = 10 ticks,
- upon "import", the RPC runtime at a client initializes the starting sequence number for the client to 100, and the conversation-id of the client to be equal to the real-time clock value,
- upon "export", the RPC runtime at a server assigns a unique-id of the server to be equal to the current real-time clock value,
- zero time for message transmission between clients and server,
- the RPC server executes incoming RPC calls one-at-a-time in FCFS order,
- the call $f()$ takes 15 ticks at the server for execution in the absence of any waiting,
- the client C1 has machine-id (mid) = 0, and process-id (pid) = 100, and gets the same id upon crash and restart,
- the client C2 has machine-id = 1, and process-id = 100,
- the call packet sent by the RPC client looks as follows:

```
-------------------------------------------------------------------------
| uid  | conv-id   | mid  | pid  | seq-num  |  func-call  |  args  |
-------------------------------------------------------------------------
```

where uid is the unique-id of the server; and conv-id is the conversation id.

Given below is an execution timeline of calls from 2 RPC clients, and a server. These calls result in communication among the client, the server, and the nameserver as discussed in the RPC paper by Birrel and Nelson. Show the interactions among the RPC clients, the server, and the nameserver for this execution.

```
                  C1                  C2                     S
       -------------------------------------------------------|
       |
     0 |------------------------------------------------- export(f)
       |
     5 |------ import(f)
    10 |------ f()
    15 |------------------------- import(f)
    22 |------------------------- f()
    40 |------ crash
    45 |------ import(f)
    50 |------ f()
    55 |------------------------------------------------- crash
    60 |------------------------------------------------- export(f)
       |
       V
     time
    (ticks)
```

Additional Workspace: