Fall 2024 CS 692 Systems Capstone Exam

Please read the following instructions before you start the exam. Please avoid asking a question that is addressed below:

1) You can write your answers on both sides of the paper. If you run out of space as you answer the questions on the front pages, continue your answer on the back of the page. Please do not ask the instructor whether you can write on the back side of the paper. The answer is YES YOU CAN.

2) You may use the last paper sheet (front and back) as scratch paper. It is also marked as scratch paper. Please note that the instructor will NOT grade the scratch paper. The scratch sheet stays with the instructor. Please do not detach it.

3) Do NOT detach the exam sheets. If at any point the papers got detached, raise your hand and request for them to be stapled immediately.

4) You are NOT allowed to use Calculator.

5) You are NOT allowed to have your cell phone, e-watch, or other electronic devices nearby (on the desk or inside your pocket). They should be turned off and stay inside your bag /backpack which will be placed in the back of the room.

6) Exam duration: 3:00 to 4:30 pm.

Choose two of the three questions. Please indicate the questions you have completed for grading below. If they are not indicated, we will assume that the first two questions that you attempted are to be graded.

- ❑ Question #1
- ❑ Question #2
- ❑ Question #3

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Question 1 parts a, b (20 points total)

1.a (5pts) Which requirements must be satisfied for any solution to the critical section problem?

1.b (15pts) Does the code below satisfy these requirements? If a requirement is satisfied, explain why. If a requirement is not satisfied, explain why.

```
#define NUM_THREADS 2

bool flag[NUM_THREADS];
int turn;

void *process (void *id) {
  int i = *((int *) id);
  while (1) {
      flag[i] = true;
      turn = (i + 1) % NUM_THREADS;
      while (flag[(i + 1) % NUM_THREADS] && turn == (i + 1) % NUM_THREADS) ;
        // critical section
      printf("Thread %d in critical section\n", i);
      sleep(1);
      flag[i] = false;
      // non-critical section
  }
  return NULL;
}
int main() {
   pthread_t threads[NUM_THREADS];
   int thread_id[NUM_THREADS];
   for (int i = 0; i < NUM_THREADS; i++) {
      thread_id[i] = i;
      pthread_create(&threads[i], NULL, process, &thread_id[i]);
  }
  for (int i = 0; i < NUM_THREADS; i++) {
    pthread_join(threads[i], NULL);
  }
  return 0;
}
```

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Question 2 parts a, b, c, d, e, f (20 points total)

2.a (4pts) List the four standard necessary conditions of Deadlock.

2.b (2pts) True or False: Releasing all resources before requesting a new resource is a valid deadlock prevention scheme.

2.c (2pts) True or False: In a Resource Allocation Graph (RAG), where there is one instance of each resource, a cycle implies that there is deadlock.

Let T be a set of threads and R be a set of resources. R1 has 2 instances. R2 has 1 instance, and R3 has 2 instances.

The sets P and R, are as follows: T = {T1, T2, T3, T4, T5} R = {R1, R2, R3, R4, R5}

T1 is assigned an instance of R3 and wants an instance of R1

T2 is assigned an instance of R1

T3 is assigned an instance of R1 and wants an instance of R2

T4 is assigned an instance of R2 and wants an instance of R3

T5 is assigned an instance of R3

2.d (2pts) Show the set E based on the above assignments and wants

2.e (4pts) Show the resource allocation graph

2.f (6pts) Is there deadlock in this situation? Briefly explain your answer.

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Question 3 parts a, b, c, d, e (20 points total)

3.a (4pts) There are multiple techniques for structuring page tables in large address spaces. Choose one and describe it.

3.b (4pts) Assume 4 page frames have been allocated to a process. Show the contents of the page frames for the following page requests and show if a page fault has occurred using the **First in First Out** page replacement algorithm:

Pages: 3, 4, 5, 6, 4, 3, 7, 8, 4, 3, 4, 5, 9

3.c (4pts) Assume 4 page frames have been allocated to a process. Show the contents of the page frames for the following page requests and show if a page fault has occurred using the **Least Recently** used page replacement algorithm:

Pages: 3, 4, 5, 6, 4, 3, 7, 8, 4, 3, 4, 5, 9

3.d (4pts) Assume 4 page frames have been allocated to a process. Show the contents of the page frames for the following page requests and show if a page fault has occurred using the **Optimal** page replacement algorithm:

Pages: 3, 4, 5, 6, 4, 3, 7, 8, 4, 3, 4, 5, 9

3.e (4pts) For this particular set of pages, which method is best and why? Describe any disadvantage of the "best" method if there is one.

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Full name: _____ Net ID:_____

Scratch paper