1 Rules of the Exam

This examination is open book and open note (your notes only please). Calculators are permitted. Networked devices are strictly prohibited. The questions are marked as to their relative value, the exam will be scored out of 100% but is worth 25 points towards your course grade. Relax and try to do what you can.

1. Interprocessor Communication/Systems Programming (total 20%):

   (a) (12%) Your company’s Unix based software would benefit if it could modify the size of an already allocated piece of shared memory. Unfortunately the Unix shared memory facility does not allow applications programs to increase the size of a shared memory segment. Your current implementation uses direct addressing of a shared memory segment with all access via pointers stored in their local memory (see Figure 1a).
   
   Over lunch, a coworker suggests using indirect addressing to get around this, by putting a pointer to the shared memory segment in shared memory (see Figure 1b). To protect the integrity of the pointer she suggests using a semaphore to guard the allocation of the shared memory. She has written the pseudocode in Figure 2 on a napkin over lunch. Discuss the advantages and disadvantages of this scheme.

   (b) (8%) Your company sells this software on many platforms, and is concerned about portability. Would the previously suggested approach be likely to work better or worse under another operating system?

2. Synchronization (25 %)

   (a) (15%) A fraternity has N brothers who are about the same size and share an expensive suit, three couches and five operating systems text books. Each brother sleeps for a while, and then goes out on a job interview wearing the suit, removing the suit when they are done. The suit comes in two parts, the pants and the jacket, each must be put on (and removed) separately. In order to fall asleep each brother needs both an operating systems text book and a couch. A brother that wants to sleep or prepare for a job interview must wait until he can acquire the resources he needs to do so (although if he is kept waiting indefinitely he will get angry). Design a semaphore based solution. Be careful to avoid deadlock and starvation.

   (b) (10%) Design a monitor based solution for the previous problem.

3. Concurrency (20 %)

   (a) (10 %) Use Bernstein’s conditions to transform the following sequential code segment into a a maximally parallel code segment using parbegin/parent notation. You can assume that you will translate the code into a multithreaded environment (i.e. the threads can share data).
semaphore mutex := 1;  /* global mutex */
int shm_key = ftok(...);  /* global key for shared memory management */

void *shmrealloc(int *shm_id_ptr, void *shm_ptr, int new_size )
{
    int old_size;
    void *old_shm_ptr;
    sem_wait(mutex);  /* limit access to one process at a time */
    old_size = shmmgets_size(*shm_id_ptr );  /* get shared memory segment’s size */
    if (old_size != new_size){  /* change the segment’s size */
        /* create a new shared memory segment using same key */
        old_shm_ptr = *shm_ptr;
        shm_id_ptr = shmmget( shm_key, new_size,
                           /* create a readable/writeable shared mem segment */
                           SHM_R |SHM_W | IPC_CREAT);
        /* ‘‘attach’’ a pointer to the new shared memory’s segment */
        *shm_ptr = shmat( *shm_id,
                        (void *NULL),  /* use default location */
                        0);  /* use default flags */
        shm_delete( *shm_id_ptr );  /* deallocate old segment */
    }
    sem_signal(mutex);  /* release resource for user access */
}

Figure 1: A Shared Memory Design in Problem 1a

Figure 2: Proposed Pseudocode for Problem 1a
\[ x := x + z; \\
\]
\[ y := y * y; \\
\]
\[ w := w + x; \\
\]
\[ z := y - 3; \\
\]
\[ v := y + w; \]

Assuming that additions and subtractions take one unit of time and multiplications take 4 units of time, what is the minimum execution time.

(b) (10 \%) \textit{Barrier} synchronization constructs are used for ensuring that all processes have arrived at a collective synchronization point. Write an efficient message passing barrier synchronization routine for \( N \) processes. If it helps you may assume that \( N \) is a power of 2.

4. (20 \%) \textbf{Deadlock}

(a) (10 \%) Discuss (briefly) each of the conditions necessary for deadlock and the cost of preventing each from occurring for each of the following resources:

i. A Printer

ii. Main Memory

iii. A message buffer.

(b) (10 \%) Consider a system with a total of 4 identical instances of a resource to be shared among 3 processes, each requiring 2 instances of the resource. Show whether or not the system is deadlock has deadlock unsafe states?

5. (20 \%) \textbf{Intro to Networking}

(a) (10 \%) What compare and contrast the costs of a network model having relatively few layers (say a 4 layer network model, such as TCP/IP) and a network model having many layers (say a a 7 layer model such as the ISO/OSI model).

(b) (10 \%) Give the Point to Point Protocol (PPP) packet containing the message (note there are unprintable ASCII characters represented by \textbackslash{} C style backslash escape sequences, those with a leading 0x are in hexadecimal, otherwise octal). The following text is the data to send:

```
Damn \0x7e Keyboard \0x7d! At Least I have Completed The Final Exam
```

3