

### MIDTERM 2 EXAM

- This test contains 9 questions with a total of 66 points
  - 1-7           6 points each
  - 8-9           12 points each
- You have 50 minutes to complete this midterm.
- You may **not** use your text, or any other reference material.
- Do not turn this page or turn over the exam until instructed to do so.

### Hints

- Don't spend too much time on short answer questions. If you don't know the answer right away, move on and come back.
- There may be some questions that will require more time to think and organize your thoughts. If you find yourself stuck on one of these questions, save it until you have answered all the questions you can answer quickly. If you spend too much time on one question you may get frustrated and that will impair your ability to answer the other questions.
- Don't turn your test in early. If you have time left over, use it to review your answers. Students who turn tests in early often make trivial mistakes that they would catch if they reviewed their answers.
- Students that get the highest grades usually go over their answers several times.
- The space below a question **does not** indicate how long of an answer I am expecting. Sometimes I want to start the next question on a new page.
- Double check that you understand the question before you answer it. If you don't understand the question, ask me.

1) (6 points) Disk I/O is very slow. If a process reads one character at a time from a file it will have to block for each character and wait for the disk driver to put the character in memory. What can be done to speed up reading from files?

Disk drives have memory used to cache files. Since it is faster to read from memory than from the disk once a file is in the cache it is faster to read from it.

It is also possible to bring an entire file (or large chunks of the file) into main memory. Then when a process reads from the file the data will already be in memory. The process does not need to block it can simply use the data in memory. This mechanism is called *memory mapped files*.

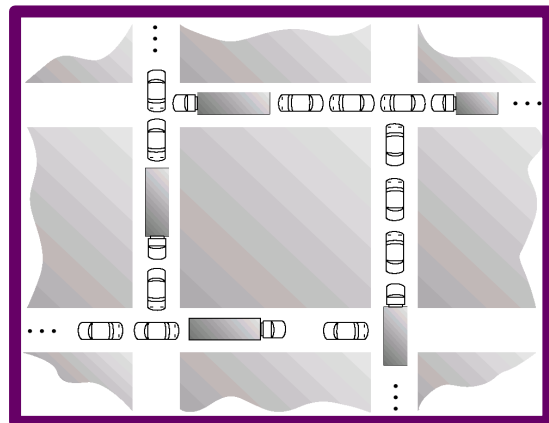
2) (6 points) What is the difference between deadlock and starvation?

When a set of processes are in a state of deadlock no process will ever get to execute. When one or more processes are starved they may never get to execute but other processes are executing.

3) (6 points) Give an example of deadlock. Explain how it meets the four conditions for deadlock.

Any deadlock situation would be acceptable  
Must include the four conditions

Traffic Gridlock.



- 1) Mutual exclusions: only one vehicle can occupy a section of road at any one time
- 2) Hold and wait: each vehicle holds a section of road while waiting for another section
- 3) No preemption: vehicles cannot disappear
- 4) Circular wait: there is a cycle of vehicles waiting for each other (around the block)

4) (6 points) Some operating systems have the ability to detect deadlock. What can be done to end the deadlock (to recover from deadlock) after it has been detected?

Any of the following will end deadlock:

Abort all deadlocked processes.

Abort deadlocked processes one at a time until the cycle is eliminated.

Preempt resources until the the cycle is eliminated.

5) (6 points) On systems with virtual memory, as the degree of multiprogramming increases CPU utilization can decrease drastically. What causes the low CPU utilization? What can be done to solve the problem?

If there are enough processes then each time a process runs it will have a page fault. The CPU will end up spending all its time swapping pages in and out of virtual memory.

Solutions:

More physical memory

Limit the number of active processes

6) (6 points) Define global page replacement and local page replacement. Give one good and one bad aspect of each.

A global page replacement process selects a replacement from from the set of all frames.

Bad: one process can take a frame from another process.

Good: a process that needs a lot of frames can get them (more frames are utilized).

A local page replacement process selects a replacement frame from the current process.

Bad: leads to more unused pages and thus less throughput.

Good: no other process can ever take frames.

7) (6 points) Describe the concept of shared libraries and the mechanism used to implement them. What is the primary advantage of shared libraries?

Shared libraries are libraries that once loaded into memory can be used by multiple independent processes. A virtual memory management system can implement shared libraries by mapping multiple processes' logical memory to the same shared page.

The primary advantage is that the library only has to be loaded into memory once. Thus when a process uses a shared library that is already in memory (i.e. it is already being used by another process so it was loaded in memory) the library does not have to be loaded. This reduces that processes execution time.

8) (12 points) When a producer and a consumer communicate by writing and reading from a buffer, the producer and consumer must not read and write from the same spot in the buffer at the same time. Furthermore, the producer has to wait when the buffer is full and the consumer must wait when the buffer is empty. Write a pseudocode monitor solution to this problem (often called the bounded-buffer problem). Assume that the buffer is an array of characters and is a shared variable inside the monitor.

```
monitor Bounded_buffer
{
    char buffer[MAX];
    int size = 0;
    Conditional_variable full;
    Conditional_variable empty;
    int next_write = 0;
    int next_read = 0;

    void produce(char c)
    {
        while (size == MAX)
            wait(full);
        buffer[next_write] = c;
        next_write = (next_write + 1) % MAX;
        size++;
        signal(empty);
    }

    char consume()
    {
        char c;
        while (size == 0)
            wait(empty);
        c = buffer[next_read];
        next_read = (next_read + 1) % MAX;
        size--;
        signal(full);
    }
}
```

9) (12 points) An old one-lane bridge is on a north-south highway. Since it is a one-lane bridge, at any time all the cars must be going the same direction (north or south). Since it is an old bridge, only three cars can be on it at any time. Write code to simulate cars driving on the bridge. Use semaphores for synchronization. Hint: first write the solution for any number of cars and then modify it so only three cars can be on the bridge at once.

```
semaphore bridge = 1;
semaphore total_north = 3;
semaphore total_south = 3;
semaphore north_mutex = 1;
semaphore south_mutex = 1;
int north_count = 0;
int south_count = 0;
```

north:

```
    wait(total_north);
    wait(north_mutex);
    north_count++;
    if (north_count == 1)
        wait(bridge);
    signal(north_mutex);
    drive_on_bridge();
    wait(north_mutex);
    north_count--;
    if (north_count == 0)
        signal(bridge);
    signal(north_mutex);
    signal(total_north);
```

south:

```
    wait(total_south);
    wait(south_mutex);
    south_count++;
    if (south_count == 1)
        wait(bridge);
    signal(south_mutex);
    drive_on_bridge();
    wait(south_mutex);
    south_count--;
    if (south_count == 0)
        signal(bridge);
    signal(south_mutex);
    signal(total_south);
```