

# CS 111 Midterm Examination #1

## Spring Quarter, 2021

You have 90 minutes for this examination; the number of points for each question indicates roughly how many minutes you should spend on that question. While taking this exam, you may not consult any materials other than information you have explicitly prepared ahead of time. If there is a trivial detail that you need for one of your answers but cannot recall, you may ask the course staff. Fill in the fields of this PDF file, then submit it at [www.gradescope.com](http://www.gradescope.com).

*By filing this examination electronically, you acknowledge and accept the Stanford University Honor Code, and you affirm that you have neither given nor received aid in answering the questions on this examination.*

---

(Name)

---

(SUNet email id)

Problem	#1	#2	#3	#4	#5	#6	Total
Score							
Max	10	3	6	10	10	40	79





#### Problem 4 (10 points)

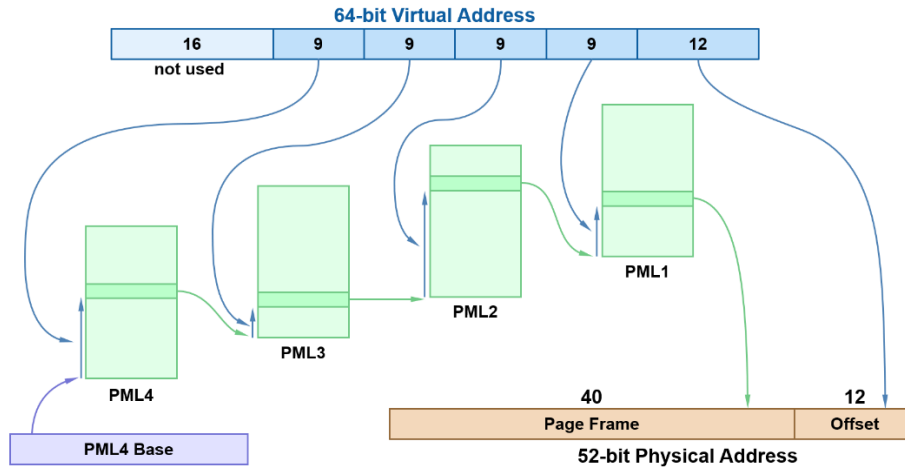
Consider the following code:

```
int x = 0;
...
std::thread t1 = new std::thread([&x] {
    int y = computeY();
    x += y;
    long_running_func_to_do_other_stuff();
}
usleep(100000);          /* Delays for 100 milliseconds */
std::thread t2 = new std::thread([&x] {
    int z = computeZ();
    x = 2*x - z;
    another_long_running_func();
}
t1.join();
t2.join();
std::cout << "The value of x is " << x << std::endl;
```

The goal of this code is to serialize the updates to `x` (with `t1` running first), while still allowing the two long-running functions to execute in parallel. Is the value that will be printed for `x` deterministic? If so, explain why; if not, explain why and modify the code to fix the problem. You may assume that `computeY` and `computeZ` are deterministic.

### Problem 5 (10 points)

The figure below shows the address translation mechanism used for 64-bit mode in Intel x86 processors, which was discussed in lecture. Starting with a 48-bit virtual addresses, it uses 4 levels of page table to translate the address to a 52-bit physical address.



The discussion in lecture showed how to use this structure for a simple process containing one page of code, one page of data, and one page of stack; the scheme shown in lecture required seven page maps. However, it is possible to execute the simple process with fewer than seven page maps.

- (a) What is the smallest number of page maps that could be used for the simple process?
  
  
  
  
  
  
  
  
  
  
- (b) How would those page maps be used for the process's code, data, and stack? Note: your solution must use separate pages for code, data, and stack.
  
  
  
  
  
  
  
  
  
  
- (c) What is the disadvantage of this approach?

### Problem 6 (40 points)

Caltrans has hired you to build a control system for a bridge. Your program must restrict the number of cars on the bridge in order to conform to a weight limit. You must define a C++ class `Bridge` with a constructor and two methods:

```
Bridge(int limit);  
void arrive(int direction, int weight);  
void leave(int direction, int weight);
```

The `limit` argument to the constructor indicates the weight limit of the bridge, in pounds. When a car arrives at the bridge it will invoke the `arrive` method, which must not return until it is safe for the car to cross the bridge. When the car finishes crossing the bridge, it will invoke `leave`. Cars can travel in two directions on the bridge; the `direction` argument to `arrive` and `leave` will be 0 or 1 to indicate the car's direction. The `weight` argument to `arrive` and `leave` gives the weight of the car, in pounds.

Your solution must satisfy the following requirements:

- The total weight of cars on the bridge must never exceed the weight limit.
- Each direction should be allowed to use half of the total weight limit. However, if the total weight of cars traveling or waiting to travel in one direction does not consume half of the limit, then the other direction may consume the remainder.
- You do not need to ensure fairness among the cars travelling in a given direction, and there will not be a penalty for unnecessary wakeups.
- You may assume that the total weight limit is at least 2x the weight of the heaviest possible car.
- You must write your solution in C++, using locks and condition variables.
- Use the monitor style for your code, as discussed in class and required for Project 3.
- Your solution must not use busy-waiting.
- Try to make your solution simple and obvious; we will reduce your score by up to 20% if your solution is overly complicated.

**Additional working space for Problem 6:**

**Additional working space for any problem, if needed**