Solutions
15-410, Spring 2006, Homework Assignment 1.

1 Academic Fare (35 pts.)

1.1 5pts

Which of the three deadlock coping strategies is the dean using?

Avoidance. Notice that there are declared usage patterns and varying allocation policies depending on how scarce resources are. The dean is relying on an implicit, non-constructive proof that the given policy will always result in a safe sequence existing.

1.2 5pts

How many forks must the dean hold when a fork request arrives in order for the request to be handled in abundant-fork mode?

Two forks.

1.3 5pts

If forks are not abundant, what additional check must the dean perform before giving a fork to a philosopher?

The dean may issue a fork only to a philosopher who is already holding one. This philosopher will eventually proceed to completion, certainly releasing two forks. Any way that these two forks join the two already distributed among the remaining three philosphers will enable another philosopher to eat.

1.4 10pts

What code would you need in `allocate_ok()` *to implement the policy?*

```
int allocate_ok(int forks_held_by_this_phil, int forks_held_by_dean)
{
  return (forks_held_by_dean > 1) || (forks_held_by_phil);
}
```

1.5 10pts

Is the call to `mutex_avail()` *needed?*

Yes. The forks are non-shareable and non-preemptable. Because the philosophers acquire forks in random order, circular wait would only be a matter of time... if we allowed hold and wait, which is avoided by `mutex_avail()`.

Assume that P1 acquires M1 (`f->right`) . P2 enters trying to grab M1 (`f->left`). It locks `dean_mutex`, then goes to sleep on M1. At this point P0 can't free M1, since it must first lock the `dean_mutex` to do so. This demonstrates that going to sleep with locks is a bad thing, and that partial lock ordering isn't a very good solution to deadlock.

2 “Six of One...”? (10 pts.)

Identify a function which benefits from function parameters being pushed in last-first order, and explain how it benefits.

An obvious example is `printf()`, which uses the first parameter, the format string, to learn the number of following parameters and the type of each. If the format string were buried an unknown distance down on the stack, the stack discipline would need some way to signal to a caller the height of the first parameter.

3 e-Commerce (10 pts.)

Personally I think the two most interesting answers are the *right* answer and the *most wrong* answer, but you won’t be penalized for showing us a different two.

Correct Trace

time	Thread 0	Thread 1
0	`cash = 100;`	
1	`cash += 50;`	
2	`store->cash = 150;`	
3		`cash = 150;`
4		`cash += 20;`
5		`store->cash = 170;`

Most-Wrong Trace

time	Thread 0	Thread 1
0	`cash = 100;`	
1		`cash = 100;`
2		`cash += 20;`
3	`cash += 50;`	
4	`store->cash = 150;`	
5		`store->cash = 120;`

The other final outcome is of course 150.