## Week 4

### L 4.4 dining philosophers (II)

Implement a deadlock free solution with condition variables, by avoiding the *hold-and-wait* deadlock condition.

Hint (change font color):   
use Booleans to indicate if the forks are in use and condition variables to notify changes in the status of the forks. There is no need for separate mutexes for each fork.   
(end of hint)

### M 5.1 dining savages

Implement with condition variables with the following requirements:

* real storage for servings (use MyBag)
* carnivore + vegetarian savages
* carnivore + vegetarian cooks

So the pot contains a variety of servings, and there are a variety of savages around the pot. And sometimes it might occur that the pot does contain servings, but not suited for the available savages.

Ensure that an arbitrary number of carnivore-savages and vegetarian-savages can be started.

### N 4.2 readers-writers

Implement with condition variables, and make it configurable who has priorities.

Ensure that an arbitrary number of reader and writer threads can be started (e.g. N=7).

Hint (change font color):   
mutex = MyMutex("mutex")  
cv\_reader = MyConditionVariable(mutex,"cv\_reader")  
cv\_writer = MyConditionVariable(mutex,"cv\_writer")  
nrof\_reader\_busy = MyInt(0,"reader\_busy")

nrof\_writer\_busy = MyInt(0,"writer\_busy")  
nrof\_reader\_wait = MyInt(0,"reader\_wait")  
nrof\_writer\_wait = MyInt(0,"writer\_wait")  
writer\_prio = MyBool(False,"writer\_prio")   
(end of hint)