Operating Systems
Project 17

Haukur Óskar Þorgeirsson
hth152@hi.is

Matthías Páll Gissurarson
mpg3@hi.is

13. mars 2013

1 First attempt

Let's imagine that all the philosophers get hungry at the same time. They then all grab the right one first, but as they are sitting in a circle, the left one will be occupied by the one on their left. Thus they all wait for the left chopstick, which never becomes available, as they will not release their right chopstick (which is the left one of somebody else) until they get the left chopstick (which is the right one of somebody else). Thus they wait endlessly.

Mutual exclusion Each chopstick can only be used by one philosopher at a time, so there is Mutual exclusion

Hold and wait Each philosopher holds the right chopstick and waits for the left one that the philosopher to the left is holding

No preemption Each philosopher releases their chopstick only when they have eaten, so there is no preemption

Circular wait Each one is waitin for the one on their left to release their chopstick so if we enumerate them, P_1 waits for P_2, P_2 waits for P_3, which waits for P_4, which waits for P_5, which waits for P_1, so we have Circular wait, and thus we have a deadlock.

2 Second attempt

Here the situation is much the same as in the first attempt, except for the fact that there is no circular wait. There can only be a deadlock when all want to eat at the same time, but then, if we let P_1 be left handed, P_1 waits for P_5, P_2 waits for P_3, P_3 waits for P_4, P_4 waits for P_5 which waits for P_1. As there is no circular wait (and cannot be), there is no deadlock.