

**CS307 System Practicum**  
**Assignment 2**  
**Problem 1**  
**Dining Philosophers Problem**

**Inferences -**

**Approaches used were -**

**Trivial Approach -**

Each philosopher attempts to pick up the left fork first and then the second fork. Use an array of mutex locks to simulate forks which each philosopher can hold. Then each philosopher drops the left fork and then right. This is prone to deadlock as it is possible that each of the philosophers picks up the left fork and then every philosopher will be waiting for a right fork which will never become available.

In circular fashion order is

P0 F1 P1 F2 P2 F3 P3 F4 P4 F0                      // Here, P stands for Philosopher and F for Fork.

**Approach 1 (main.cpp)-**

Change the trivial approach so that the odd numbered philosophers pick up the left fork first and the even philosophers pick up the right fork first. This will prevent any deadlocks from occurring as at least one philosopher can always acquire both forks. Starvation also does not appear to be a problem here but can occur if say P0 is waiting for P1 to drop his left fork but after dropping P1 again picks it up. This is highly unlikely because of the random think times introduced in the question. The main problem with this solution is that P4 will eat the most in this case. This is because P0 and P1 will fight for F1 and P2 and P3 will fight for F3 but there will always be both forks available for P4 , i.e. F0 and F4 will always be initially available. Thus, this solution gives P4 a slight advantage over others.

**Approach 2 (main2.cpp)-**

Change the trivial approach so that all philosophers except the last one pick up the right fork first. The last philosopher picks up the left fork first. This will prevent any deadlocks from occurring as at least one philosopher can always acquire both forks. Starvation is also unlikely because of the random times of thinking. So, in our case P0, P1, P2 and P3 pickup the right fork first and P4 picks up the left fork first. Here also, we observe that P0 does the most eating in each case because P4 and P3 will fight for F4 and both F0 and F1 will always be available for P0 initially. Thus, this solution gives P0 a slight advantage over others.

**How to fix uneven distribution?**

This is highly unlikely to solve in the classical problem. In each solution to this we have to modify the problem by some amount.

1. We can introduce a waiter (queue) which keeps hold of how much food each philosopher has eaten and assign forks accordingly.

2. We can also modify Approach 2 we used above to cycle the philosopher with the opposite fork picking pattern after some interval of time, say every 2 minutes. This will ensure that each philosopher gets a fair chance and will ensure a fair distribution.

#### Observations -

##### Approach 1 (main.cpp)-

out1.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	616	616	616
P1	634	635	635
P2	649	649	649
P3	673	673	673
P4	692	693	693

out2.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	614	614	615
P1	639	639	639
P2	648	648	648
P3	664	665	665
P4	699	699	700

out3.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	612	612	613
P1	636	637	637
P2	648	648	648
P3	670	671	671
P4	698	698	698

out4.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	614	614	614
P1	640	640	640
P2	650	650	650
P3	664	665	665
P4	697	697	697

out5.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	620	620	620
P1	636	636	636
P2	653	654	654
P3	661	661	661
P4	694	695	695

Approach 2 (main2.cpp)-

out2\_1.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	691	691	691
P1	680	680	680
P2	664	665	665
P3	618	618	618
P4	658	658	658

out2\_2.txt

Philosopher	Thinking	Eating	1 Fork Acquired
P0	698	698	699

<b>P1</b>	679	679	679
<b>P2</b>	668	668	668
<b>P3</b>	615	615	615
<b>P4</b>	650	651	651

out2\_3.txt

<b>Philosopher</b>	<b>Thinking</b>	<b>Eating</b>	<b>1 Fork Acquired</b>
<b>P0</b>	688	688	688
<b>P1</b>	671	671	672
<b>P2</b>	648	648	648
<b>P3</b>	595	596	596
<b>P4</b>	641	641	641

out2\_4.txt

<b>Philosopher</b>	<b>Thinking</b>	<b>Eating</b>	<b>1 Fork Acquired</b>
<b>P0</b>	700	700	700
<b>P1</b>	665	665	665
<b>P2</b>	642	642	642
<b>P3</b>	599	600	600
<b>P4</b>	637	637	637

out2\_5.txt

<b>Philosopher</b>	<b>Thinking</b>	<b>Eating</b>	<b>1 Fork Acquired</b>
<b>P0</b>	692	692	693
<b>P1</b>	678	678	678
<b>P2</b>	671	672	672
<b>P3</b>	612	612	613
<b>P4</b>	656	657	657