CS5541 - Computer Systems
Assignment Name: Dining Philosophers
Spring 2018
Due: Saturday, March 31, 2018 at 11:59 PM

Jason Eric Johnson

03/20/2018

1 Introduction

The purpose of this assignment is to become more familiar with deadlock and starvation. You will do this by implementing the Dining Philosophers Algorithm. You can find the algorithm in your Operating Systems book.

2 Handout Instructions

> There is no handout for this assignment. You will start from a "blank page" for this assignment, so all of your code should be your own. You can look around for ideas and to understand concepts by finding implementations on the internet or in books, but your final code should be your own. **NO COPYING AND PASTING!** Please include references to any code, websites, and books you use to get ideas and increase your understanding.

3 Assignment

Use threads to implement three versions of the Dining Philosophers problem. One of the programs must be free from both deadlock and starvation. One of them should be vulnerable to deadlock, and the last should be vulnerable to starvation.

4 Programming Rules

This is an individual assignment. Work on your own and do not discuss code with your classmates. As always, you may discuss concepts with me and your classmates during class time, but do not share code.

You may use any language you wish to implement this assignment, with certain restrictions. You must **explicitly** use threads.

Remember that all of your code should be your own. You can look around for ideas and to understand concepts by finding implementations on the internet or in books, but your final code should be your own. **NO COPYING AND PASTING!** Please include references to any code you use to get ideas and/or increase your understanding.

Finally, you will write a report explaining your implementation and addressing the following questions:

1. What method did you use to ensure that one version of your program was susceptible to starvation but not deadlock?
2. What method did you use to ensure that one version of your program was susceptible to deadlock but not starvation?
3. What method did you use to ensure that one version of your program was not susceptible to starvation OR deadlock?

5 Evaluation

Your score will be computed out of a maximum of 100 points based on the following distribution:

60 Correctness points.
20 Report points.
20 Presentation points.

Correctness points. Your solution must produce the correct output and you must have all three versions specified above.

Report points. Your report for this assignment should be more extensive than for previous assignments. Your report should explain exactly how your implementation works and include snippets from your code where applicable.

Presentation points. You will demonstrate your code to me. You must demonstrate the vulnerabilities specified above and explain in your own words exactly how you made each version of your program susceptible to the correct vulnerability.

6 Handin Instructions

Submit Philosophers using Elearning. Follow all the instructions on the "Homework Guidelines" page carefully.

You will submit your assignment as a single `.zip` file, named as specified in the "Homework Guidelines". It should include your report `PDF` file, which I would like you to name:

`Philosophers-Report-[YourName].pdf`

Also include your program file(s) and a `README.txt` file, which will contain authorship information and any references.

Please name your `.zip` file in the following way:

`CS5541-[YourName]-Philosophers.zip`