COP 4634 SYS & NET I - PROJECT 4

OVERVIEW

This assignment will provide experience with multi-threaded applications and the synchronization primitives that are required for their proper use. You will create a program that synchronizes access between threads to a shared data structure.

THE SCENARIO

My house has a herd of lizards and two cats. The cats sleep most of the time and their favorite toy is a lizard. The lizards live in the sago palm but find food across the driveway in the monkey grass. If too many lizards try to cross the driveway at once, the cats will see them and “play” with them. Your job is to synchronize the lizard crossings to prevent any lizards from becoming cat toys.

THE PROGRAM

Write a program in ANSI C to create *N* threads representing lizards. Each lizard thread will sleep for some random amount of time and wake up hungry. The hungry lizard thread will attempt to cross the driveway in a safe manner as shown in the pseudo code below. Crossing the driveway will take some number of seconds. Once on the other side of the driveway, the lizard will eat for some random amount of time in the monkey grass. After eating, the lizard will return home to the sago palm as soon as it safely can and sleep again.

Use one or more locks, semaphores, and/or monitors to control access to the driveway (the shared resource). Make sure your implementation follows these rules:

- Do not allow too many lizards to cross the driveway at once.
- Do not use busy waits.
- Allow the maximum possible lizards to simultaneously cross.

An additional challenge which is NOT required for full credit but could result in additional credit is to prevent bidirectional travel. That is, if one or more lizards are crossing in one direction, other lizards wanting to cross in the opposite direction must wait. This addition is not simple and could take a great deal of time to complete. Again, this is not required, but should be considered a bonus challenge.

IMPLEMENTATION SUGGESTIONS

Each lizard thread will follow an algorithm similar to the one given below. The algorithm is in pseudo code and **NOT IN C**. Do not attempt to make this code run as-is. However, the names of the functions provided in `lizard.c` will be very familiar when you read the code and comments.

```
while (world has not ended)
    sleep for up to MAX_LIZARD_SLEEP seconds
    wait until [sago -> monkey grass] crossing is safe
    cross [sago -> monkey grass]
        it takes up to CROSS_SECONDS seconds to cross
```

```
eat in the monkey grass
    it takes up to MAX_LIZARD_EAT seconds to eat
wait until [monkey grass -> sago] crossing is safe
cross [monkey grass -> sago]
    it takes up to CROSS_SECONDS  seconds to cross
```

STARTER CODE

You will be provided with a single file that contains a beginning implementation of the project solution. The file `lizard.c` contains many comments and hints about completing the project. The file can be downloaded through *eLearning*.

DELIVERABLES

Your project submission should follow the instructions below. Any submissions that do not follow the stated requirements will not be graded.

1. Follow the submission requirements of your instructor as published on *eLearning* under the Content area.
2. You should have at a minimum the following files for this assignment:
    a. `lizards.c`
    b. analysis.doc (the results from an experiment and the changes in the file)
    c. Makefile

The report *analysis.doc* describes results from multiple runs with different run times for the simulation and discusses changes you made to the code and any issues you encountered in developing your solution. In the source code, use comments with your team member names to mark the changes you made in your solution. If you do not include comments in your source code to make it more readable, points will be deducted.

GRADING

This project is worth 100 points total. The points will be given based on the following criteria:

- 70 points for correct implementation of code,
- 20 points for the report describing changes you made to implement the solution and issues you may have encountered,
- 10 points for appropriate documentation of source code.

Your code must compile and execute for you to receive credit. If compiler errors or warnings occur, no points will be given for the assignment.

DUE DATE

The project is due Monday, Oct. 23rd by 3:00 pm to the Dropbox for project 4 in *eLearning*. I will not accept submissions emailed to me or the grader. Upload ahead of time, as last minute uploads may fail. Please review the policy in the syllabus regarding late work.

COMMENTS

The provided code is designed to accept an optional argument on the command line. If the `–D` option is given debugging output statements will be printed during execution. I suggest using the debugging option while developing your program.