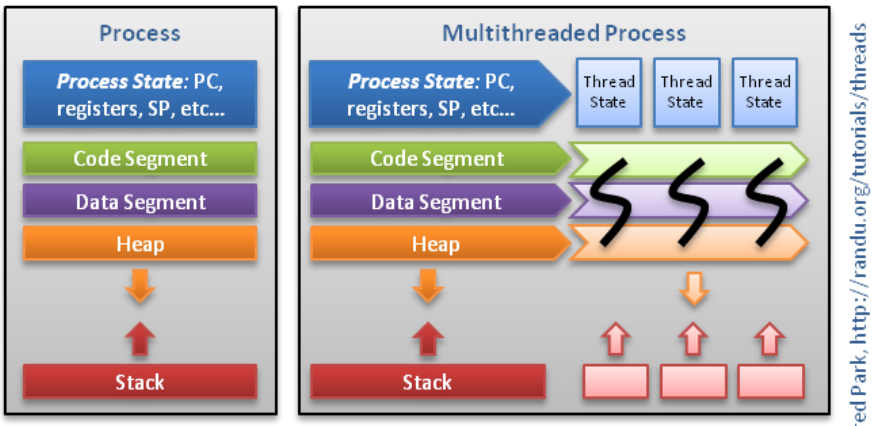

Source: https://randu.org/tutorials/threads/images/process.png

3rd Mandatory Assignment

Operating Systems(62588) Lecture 09

Teacher: Bhupjit Singh

To be handed on DTU Learn as a group.

Overview & Purpose

1. Understanding
    a. Race condition
    b. Mutual Exclusion
    c. Deadlocks
    d. Conditional Variables
    Through a basic multi-threaded application which achieves mutual exclusion.
2. Analyzing various ways of thread pools management.

Features to learn

1. How multithreading works
2. Design multi-threaded application
3. Implement your idea with pthreads/ or any other language

- For c
    - Tutorial from Peter Seebach
- For java (recommended to use semaphores instead of keyword SYNCHRONIZED)
    - Learn multithreading
    - Learn thread synchronisation
    - Semaphores in java
- For python
    - Intro to multithreading
    - Bypassing GIL
    - Python is not singlethreaded

Tasks

1. Read about issues with multithreaded application here and here.
2. Design and implement a multithreaded application program.
    a. The idea is for you to find a specific scenario where you think you will need to implement concurrency to achieve a particular task. Think of a

scenario from your life like coordinating travel plans, buying grocery etc and describe the scenario as well as the need(why) for concurrency to complete the task. One of the example is the rpg roll of dice from the peter seebach's document, another example could be synchronization of printer jobs etc.

b. It should include multithreading and mutual exclusion, it can also include deadlocks, starving or other aspects of concurrency.

How to define your idea:

You must design your idea of the multithreaded application and describe it clearly – Your idea should be a complete scenario with a specific use case, to explain your race conditions, deadlocks and their solutions.

You need to think of which kind of mutual exclusion you would want to use for example mutex, semaphores, monitors etc.

You can discuss it with the TA's or me today or TA's, (it is recommended to have a brief summary of the idea written down and ready for discussion).

In your report, you would need to discuss the following specifically as part of your design of the multithreaded application.

c. What is the idea of the application

d. Why concurrency is needed

e. What could be the potential issues specifically

f. Address race conditions

g. Solution for race condition(for example – mutual exclusion)

h. Address deadlocks and starvation

i. Solution for deadlocks and starvation (for example conditional variables)

3. You need to implement your idea; you can use any language to write your application.

4. You can also draw resource graphs (page 298 figure 6.6 of the book), joint progress diagram (example- figure 6.2 page 292 of the book) or explain your design and issues pictorially or with matrices (figure 6.7 page 303 of the book)