CSE 2040 Programming IV Lecture #35

What will we learn today

- Threads
- Single Tasking
- Multitasking
- Differences between a Process and a Thread
- Concurrent programming and GIL
- Uses of Threads

Threads

- A thread represents a separate path of execution of a group of statements.
- In a python program, if we write a group of statements, then these statements are executed by Python Virtual Machine (PVM) one by one.
- This execution is called a thread, because PVM uses a thread to execute these statements.

Cont'd

```
import threading


print('Let us find the current thread')
print('Currently running thread:')
print(threading.current_thread().getName())
```

```
====== RESTART: C:/Users/MIITKK
Let us find the current thread
Currently running thread:
MainThread
>>>
```

Cont’d

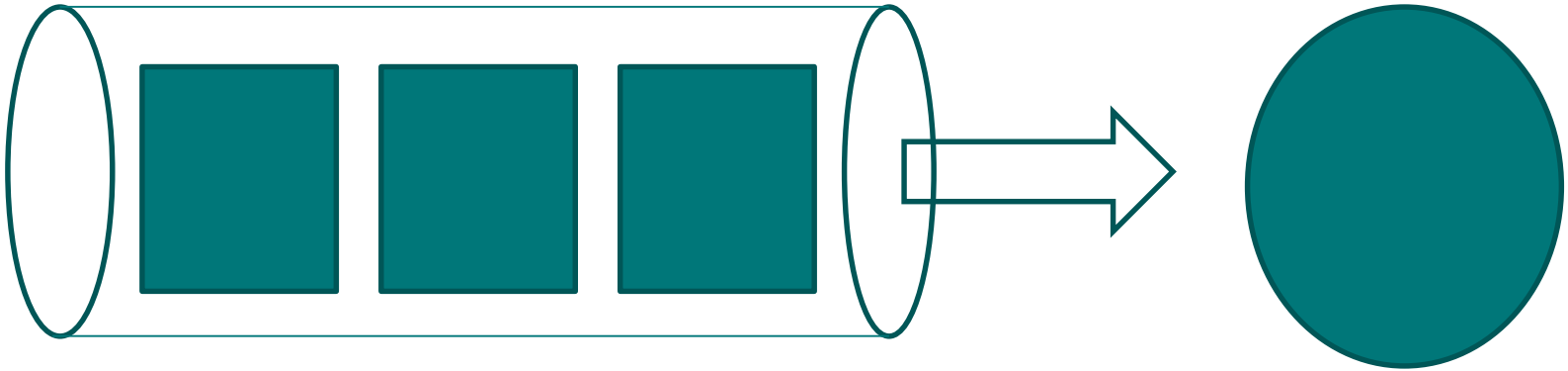
A thread represents execution of statements. The way the statements are executed is of two types:

1. Single Tasking
2. Multitasking

Single Tasking

- A task means doing some calculation, processing, etc.
- Nowadays, processors can execute millions of instructions per second, even if the student has written a program of 100 lines, the processor will not take more than a fraction of a second to complete its execution.
- Executing the tasks only one at a time is called single tasking.

Single Tasking



Jobs or Programs

Processor

Multitasking

- To use the processor’s time in an optimum way
- Executing more than one task at a time is called multitasking.
- The main advantage of multitasking is to use the processor time in a better way.
- Complete several tasks at a time, and thus achieve good performance.

Two Types of Multitasking

- Multitasking is of two types:
  - Process-based multitasking: several programs are executed at a time by the microprocessor.
  - Thread-based multitasking: several parts of the same program is executed at a time by the microprocessor.

Differences between a Process and a Thread

- A process represents a group of statements which are executed by the PVM using a main thread. We can take a running program as an example for a process. Each process will have its own memory, a program counter that keeps track of the instruction being executed and a stack that holds the data. Any program utilizes resources like memory and processor time. Hence it is called heavy weight processes.

Cont'd

- A thread also represents a group of statements within a program. When we want to use threads, we have to create them separately which are in turn run by the main thread. Threads will not have their own memory and program counter. The data of one thread is shared easily by another thread. So, it is possible that a thread can easily modify the data of another thread.
- A thread is a small part of a program that takes very less memory and processor time. Hence threads are called light weight processes.

Concurrent Programming

- To create multiple threads and set them to execute different parts of the program simultaneously.
- Executing the tasks or parts of a program simultaneously is called ‘concurrent programming’.

Global Interpreter lock (GIL)

- When more than one thread is running at a time, the data of one thread is available to another thread. In such cases, that the data may undergo unwanted manipulations. This will lead to wrong results.
- PVM is not thread safe. Hence, PVM uses an internal global interpreter lock (GIL) that allows only a single thread to execute at any given moment.

Uses of Threads

- Threads are mainly used in server-side programs
- Threads are also used to create games and animation.
  - Animation means moving objects from one place to another.

Creating Threads in Python

- Python provides Thread class of threading module that is useful to create threads.
- Ways to create a thread:
  - Creating a thread without using a class
  - Creating a thread by creating a sub class to Thread class
  - Creating a thread without creating sub class to Thread class

Reading Assignment: page no – 543 to 547 of reference book!

Single Tasking using a Thread

- A thread can be employed to execute one task at a time.
- Let’s plan a program for preparation of tea in three steps as:
  - Boil milk and tea powder for 5 minutes.
  - Add sugar and boil for 3 minutes.
  - Filter it and serve.

Multitasking using Multiple Threads

- In multitasking, several tasks are executed at a time.
- Using more than one thread is called ‘multi threading’ and multi threading is used in multitasking.

Multitasking using Multiple Threads

- When we go to a movie theatre, generally a person is there at the door-checking and cutting the tickets. When we enter the hall, there are another person who shows the chairs to us. Suppose there is only one person (1 thread) doing these two tasks. He has to first cut the ticket and then come along with the first person to show the chair. Then he goes back to the door to cut the second ticket and then again walk with the second person to show the chair. Like this, if he does the things one by one, it takes a lot of time, and even though the show is over, there will be still people left outside the door waiting to enter the hall. This is pretty well known to the theatre management. So what they do?

References

- Dr. R. Nageswara Rao, Core Python Programming, Second Edition, 2018