**Mr. Powell**

**CIS-087 Python**

**Chapter 10 Exercises**

**Section 10-1**

**Page 364**

**1. What does a thread’s run method do**?

a thread is an object like any other in that it can hold data, be stored in data structures, and be passed as parameters to methods. However, some code defined in a thread can also be executed as a process. To execute this code, a thread’s class must implement a run method.

**2. What is time slicing**?

Most computers running Python programs automatically time-out a running thread every few milliseconds. The process of automatically timing-out, also known as **time slicing**, has the effect of pausing the running thread’s execution and sending it to the rear of the ready queue.

**3. What is a synchronization problem**?

Although the producer always produces all of its data, the consumer can access data that are not there, can miss data, and can access the same data more than once. These are known as **synchronization problems**

**4. What is the difference between a sleeping thread and a waiting thread**?

**sleep**—The making of a thread inactive for a designated period of time.

**wait**—The making of a thread inactive until a condition becomes true.

**5. Give two real-world examples of the producer-consumer problem**.

synchronization problems is not hard to spot in this code. On each pass through their main loops, the threads sleep for a random interval of time. Thus, if the consumer thread has a shorter interval than the producer thread on a given cycle, the consumer wakes up sooner and accesses the shared cell before the producer has a chance to write the next datum. Conversely, if the producer thread wakes up sooner, it accesses the shared data and writes the next datum before the consumer has a chance to read the previous datum.

To solve this problem, we need to synchronize the actions of the producer and consumer threads. In addition to holding data, the shared cell must be in one of two states: writeable or not writeable. The cell is writeable if it has not yet been written to (at start-up) or if it has just been read from. The cell is not writeable if it has just been written to. These two conditions can now control the callers of the setData and getData methods in the SharedCell class

**Section 10-2**

**Page 371**

**1. Give two real-world examples of the reader's and the writers' problem**.

The readers and writers problem,

* readers access the data to observe it
* writers access the data to modify it
* only one writer can be writing at a given time, and that writer must be able to finish
* before other writers or readers can begin writing or reading
* multiple readers can read the shared data concurrently without waiting for each other to

finish, but all active readers must finish before a writer starts writing

**2. State two ways in which the readers and writers problem is different from the producer-consumer problem**.

a. One or more producers are generating a type of data ad placing them in the buffer.

b. A single consumer is taking items out of the buffer one at a time

c. the system is used to prevent overlap of buffer operations only one agent may access the buffer at one time.

**3. Describe how you would make the Student class from Chapter 9 thread-safe for**

**readers and writers**.

Use the threadSafeStudent class using a decorator pattern. The class has the same interface as Students but it would encase Student object within a shared cell that will give you the synchronization for the accessor and mutator methods

**4. Define a new class called PCCell. This class provides an abstraction of a shared cell for the producer-consumer problem. The design pattern should be similar for the one presented for the shared cell for readers and writers, but it should use the mechanism specific to the producer-consumer situation**.

import time, random

from threading import Condition

class PCSharedCell(object):

"""Shared data that sequences reading before writing."""

def \_\_init\_\_(self, data):

"""Can produce but not consume at startup."""

self.data = data

self.writeable = True

self.condition = Condition()

def write(self, data, readerFunction):

"""Second caller must wait until someone has

consumed the data before resetting it."""

self.condition.acquire()

while not self.writeable:

self.condition.wait()

result = writerFunction(self.data)

self.writeable = False

self.condition.notify()

self.condition.release()

return result

def read(self, readerFunction):

"""Caller must wait until someone has produced

the data before accessing it."""

self.condition.acquire()

while self.writeable:

self.condition.wait()

result = readerFunction(self.data)

self.writeable = True

self.condition.notify()

self.condition.release()

return result

**Section 10-3**

**Page**

**1. Explain the role that ports and IP addresses play in a client/server program**.

Every computer on a network has a unique identifier called an **IP address** (IP stands for Internet Protocol). This address can be specified either as an **IP number** or as an **IP name**. An IP number typically has the form *ddd.ddd.ddd.ddd*, where each *d* is a digit. The number of digits to the right or the left of a decimal point may vary but does not exceed three. For example, the IP number of the author’s office computer might be 137.112.194.77.

A port serves as a channel through which several clients can exchange data with the same server or with different servers. Ports are usually specified by numbers. Some ports are dedicated to special servers or tasks.

For example, almost every computer reserves port number 13 for the day/time server, which allows clients to obtain the date and time. Port number 80 is reserved for a Web server, and so forth. Most computers also have hundreds or even thousands of free ports available for use by any network applications.

**2. What is a local-host, and how is it used to develop networked applications**?

that is, on a standalone computer that may or may not be connected to the Internet. The computer’s IP name, in this case, is "localhost", a name that is standard for any computer. The IP number of a computer that acts as a local host is distinct from its IP number as an **Internet host**,

**3. Why is it a good idea for a server to create threads to handle clients’ requests**?

After it creates the clients’ receipt, it goes back to listening for another client request

**4. Describe how a menu-driven command processor of the type developed for an ATM application in Chapter 9 could be run on a network**.

It can run a client application

Client provides username and password

Once logged in you go to menu

Once client makes selection, a string is sent to server to request that service

The string has a command and any agruements needed

Once the server receives the string, it does the transaction with bank and also sends replies to the client with the results of the transaction.

**5. The ATM application discussed in Chapter 9 has a single user. Will there be a synchronization problem if we deploy that application with threads for multiple users? Justify your answer**.

O synch problem to access an individual account as long as they are not joint accounts. If yes, then a locking mechanism needs to be in place to allow that the account is accessed by one writer at a time.