

**DISTRIBUTED SYSTEMS**  
**LAB ASSIGNMENT-3**

**SUBMITTED TO**  
**Professor ARUNKUMAR.T**

**NAME:SINDUMANI.M**  
**REGNO:19MIC0002**  
**COURSE CODE:CSI3012**  
**SLOT:L15+L16**

# 1.DESIGN AND IMPLEMENT LAMPORT ALGORITHM FOR MUTUAL EXCLUSION

## **What is LAMPORT ALGORITHM?**

Lamport proposed a bakery algorithm, a software solution, for the n process mutual exclusion problem. This algorithm solves a critical problem, following the fairest, first come, first serve principle.

## **The Lamport algorithm meets all the requirements of the critical section problem:**

- **Mutual Exclusion:** We are aware that a process with the lowest number is permitted to enter its critical section when no processes are already running in it. If two processes share the same token number, then what? In that situation, the process with the lowest process ID is chosen as each process's process ID is unique and there will only ever be one process running in a process' crucial section at any given moment. Therefore, the condition of mutual exclusion is satisfied.
- **Progress:** A waiting process examines whether any other waiting processes have a higher priority to enter its critical section after choosing a token. If such a process does not exist, P will immediately start its critical phase. achieving the conditions for progress.
- **Bounded Waiting:** As awaiting, the process will enter its critical section when no other process is in its critical section and
  - If its token number is the smallest among other waiting processes.
  - If token numbers are the same, it has the lowest process ID among other waiting processes

## **ADVANTAGES:**

- Lamport's algorithms are not starved for resources.
- This algorithm uses the FIFO method.
- Atomic registers are how this algorithm operates.
- For the general case of the N process, this algorithm is one of the most straightforward known solutions to the mutual exclusion problem.
- This algorithm makes sure that shared resources are used effectively in a multithreaded setting.

## **DISADVANTAGES:**

Lamport's bakery algorithm is unreliable because it will stop working if any one of the processes fails. It enters and exits the crucial part with  $3(N - 1)$  messages, which is a high message complexity.

## **CODE:**

```
from multiprocessing import Process, Pipe
from os import getpid
from datetime import datetime

def local_time(counter):
    return ' (LAMPORT_TIME={ },
LOCAL_TIME={ })'.format(counter,datetime.now())

def calc_rcv_timestamp(rcv_time_stamp, counter):
    return max(rcv_time_stamp, counter) + 1

def event(pid, counter):
    counter += 1
    print('Something happened in { } !'.\
        format(pid) + local_time(counter))
    return counter

def send_message(pipe, pid, counter):
    counter += 1
    pipe.send(('Empty shell', counter))
    print('Message sent from ' + str(pid) + local_time(counter))
    return counter

def rcv_message(pipe, pid, counter):
    message, timestamp = pipe.recv()
    counter = calc_rcv_timestamp(timestamp, counter)
    print('Message received at ' + str(pid) + local_time(counter))
    return counter
```

```
def process_one(pipe12):  
    pid = getpid()  
    counter = 0  
    counter = event(pid, counter)  
    counter = send_message(pipe12, pid, counter)  
    counter = event(pid, counter)  
    counter = recv_message(pipe12, pid, counter)  
    counter = event(pid, counter)
```

```
def process_two(pipe21, pipe23):  
    pid = getpid()  
    counter = 0  
    counter = recv_message(pipe21, pid, counter)  
    counter = send_message(pipe21, pid, counter)  
    counter = send_message(pipe23, pid, counter)  
    counter = recv_message(pipe23, pid, counter)
```

```
def process_three(pipe32):  
    pid = getpid()  
    counter = 0  
    counter = recv_message(pipe32, pid, counter)  
    counter = send_message(pipe32, pid, counter)  
if __name__ == '__main__':  
    oneandtwo, twoandone = Pipe()  
    twoandthree, threeandtwo = Pipe()
```

```
process1 = Process(target=process_one,  
                    args=(oneandtwo,))  
process2 = Process(target=process_two,  
                    args=(twoandone, twoandthree))  
process3 = Process(target=process_three,  
                    args=(threeandtwo,))
```

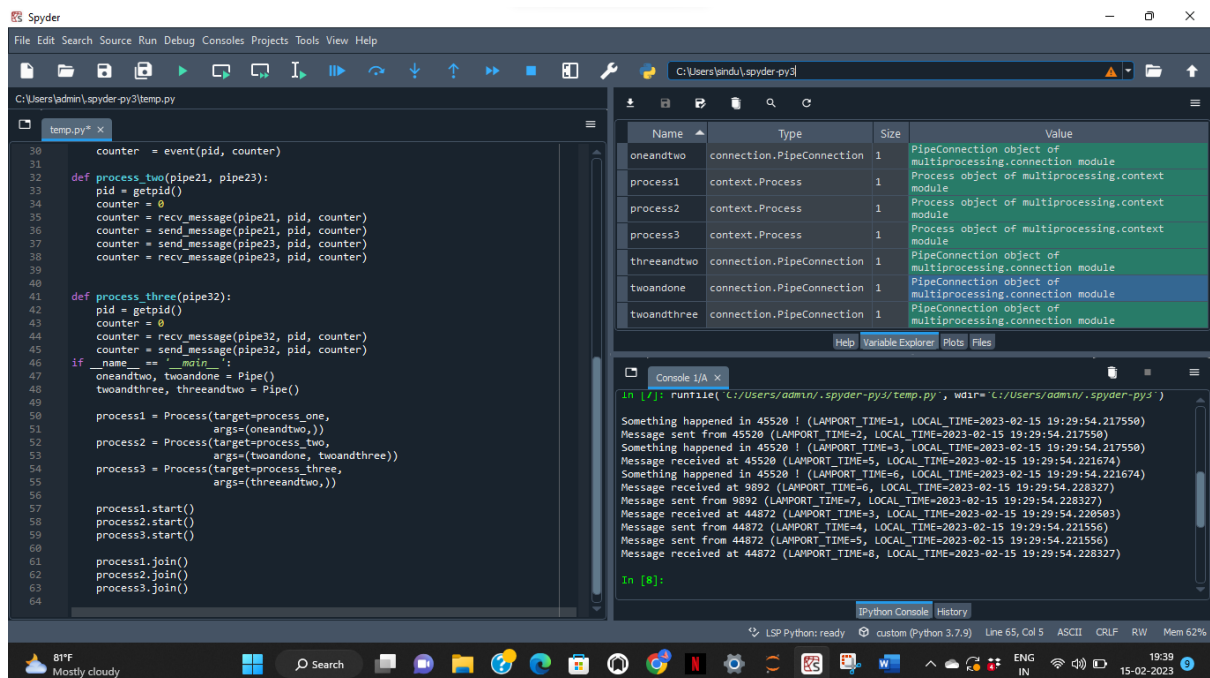
```
process1.start()  
process2.start()  
process3.start()
```

```
process1.join()  
process2.join()  
process3.join()
```

```
from time import sleep
```

```
def process_one(pipe12):  
    pid = getpid()  
    counter = 0  
    counter = event(pid, counter)  
    counter = send_message(pipe12, pid, counter)  
    sleep(3)  
    counter = event(pid, counter)  
    counter = recv_message(pipe12, pid, counter)  
    counter = event(pid, counter)
```

## **IMPLEMENTATION:**



i) The type of events and the updated Lamport timestamp.

The screenshot displays the Spyder Python IDE interface. The top panel shows the Variable Explorer with a table of variables. The bottom panel shows the IPython Console with a series of log messages.

Name	Type	Size	Value
oneandtwo	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
process1	context.Process	1	Process object of multiprocessing.context module
process2	context.Process	1	Process object of multiprocessing.context module
process3	context.Process	1	Process object of multiprocessing.context module
threeandtwo	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
twoandone	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
twoandthree	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module

Console 1/A x

```
In [1]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
```

Something happened in 45520 ! (LAMPOR\_TIME=1, LOCAL\_TIME=2023-02-15 19:29:54.217550)  
Message sent from 45520 (LAMPOR\_TIME=2, LOCAL\_TIME=2023-02-15 19:29:54.217550)  
Something happened in 45520 ! (LAMPOR\_TIME=3, LOCAL\_TIME=2023-02-15 19:29:54.217550)  
Message received at 45520 (LAMPOR\_TIME=5, LOCAL\_TIME=2023-02-15 19:29:54.221674)  
Something happened in 45520 ! (LAMPOR\_TIME=6, LOCAL\_TIME=2023-02-15 19:29:54.221674)  
Message received at 9892 (LAMPOR\_TIME=6, LOCAL\_TIME=2023-02-15 19:29:54.228327)  
Message sent from 9892 (LAMPOR\_TIME=7, LOCAL\_TIME=2023-02-15 19:29:54.228327)  
Message received at 44872 (LAMPOR\_TIME=3, LOCAL\_TIME=2023-02-15 19:29:54.220503)  
Message sent from 44872 (LAMPOR\_TIME=4, LOCAL\_TIME=2023-02-15 19:29:54.221556)  
Message sent from 44872 (LAMPOR\_TIME=5, LOCAL\_TIME=2023-02-15 19:29:54.221556)  
Message received at 44872 (LAMPOR\_TIME=8, LOCAL\_TIME=2023-02-15 19:29:54.228327)

In [8]:

IPython Console History

LSP Python: ready custom (Python 3.7.9) Line 39, Col 1 ASCII CRLF RW Mem 61%

19:35 15-02-2023

ii) Order of the events has changed, while our timestamps have stayed the same.

The screenshot displays the Spyder IDE interface. The top panel shows the Variable Explorer with a table of variables. The bottom panel shows the IPython Console with a log of messages.

Name	Type	Size	Value
oneandtwo	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
process1	context.Process	1	Process object of multiprocessing.context module
process2	context.Process	1	Process object of multiprocessing.context module
process3	context.Process	1	Process object of multiprocessing.context module
threeandtwo	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
twoandone	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
twoandthree	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module

Console 1/A x

```
In [8]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
```

Message received at 44796 (LAMPOR\_TIME=6, LOCAL\_TIME=2023-02-15 19:42:05.676677)  
Message sent from 44796 (LAMPOR\_TIME=7, LOCAL\_TIME=2023-02-15 19:42:05.676677)  
Message received at 20628 (LAMPOR\_TIME=3, LOCAL\_TIME=2023-02-15 19:42:05.669681)  
Message sent from 20628 (LAMPOR\_TIME=4, LOCAL\_TIME=2023-02-15 19:42:05.670680)  
Message sent from 20628 (LAMPOR\_TIME=5, LOCAL\_TIME=2023-02-15 19:42:05.670680)  
Message received at 20628 (LAMPOR\_TIME=8, LOCAL\_TIME=2023-02-15 19:42:05.676677)

Something happened in 29232 ! (LAMPOR\_TIME=1, LOCAL\_TIME=2023-02-15 19:42:05.667680)  
Message sent from 29232 (LAMPOR\_TIME=2, LOCAL\_TIME=2023-02-15 19:42:05.667680)  
Something happened in 29232 ! (LAMPOR\_TIME=3, LOCAL\_TIME=2023-02-15 19:42:08.675370)  
Message received at 29232 (LAMPOR\_TIME=5, LOCAL\_TIME=2023-02-15 19:42:08.675370)  
Something happened in 29232 ! (LAMPOR\_TIME=6, LOCAL\_TIME=2023-02-15 19:42:08.675370)

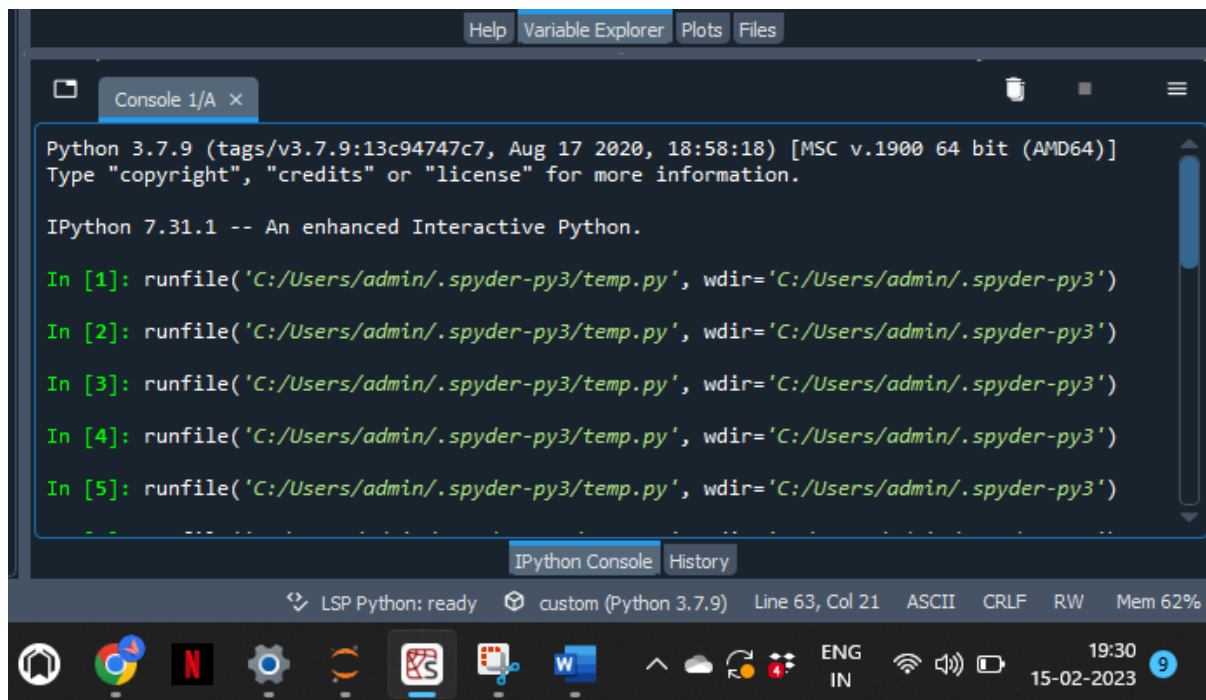
IPython Console History

LSP Python: ready custom (Python 3.7.9) Line 77, Col 19 ASCII CRLF RW Mem 62%

19:47 15-02-2023



## OUTPUT:

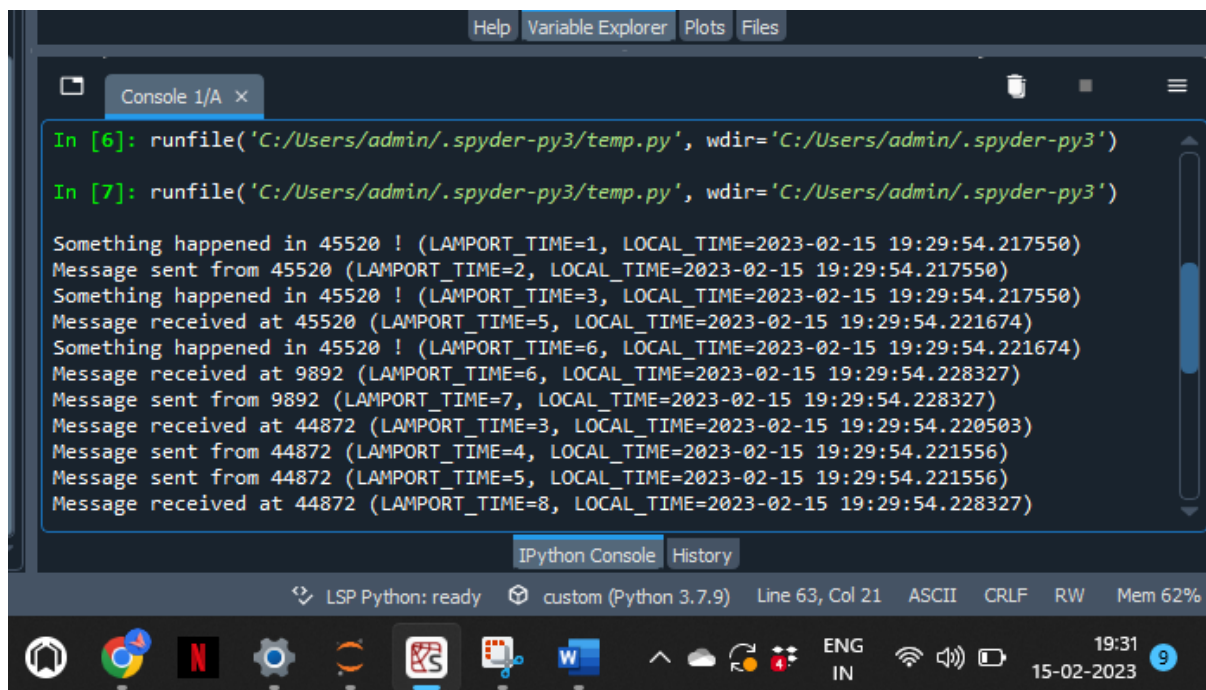


```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
In [2]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
In [3]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
In [4]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
In [5]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
```

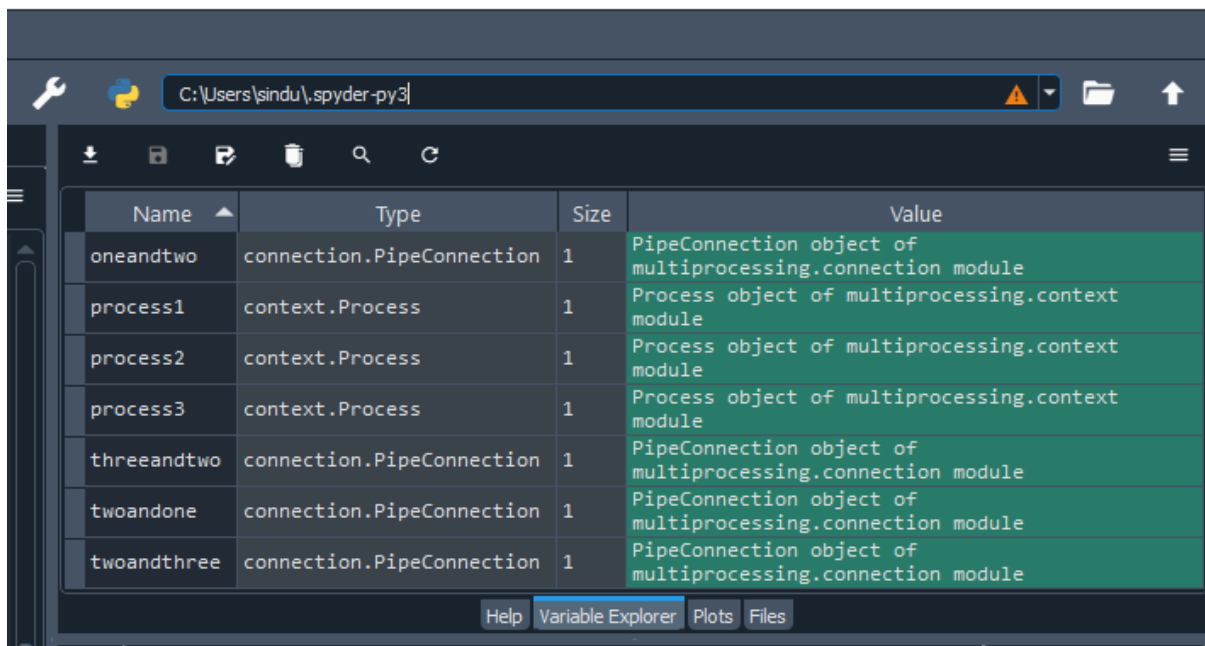
i)The type of events and the updated Lamport timestamp.



```
In [6]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')
In [7]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')

Something happened in 45520 ! (LAMPOR_TTIME=1, LOCAL_TTIME=2023-02-15 19:29:54.217550)
Message sent from 45520 (LAMPOR_TTIME=2, LOCAL_TTIME=2023-02-15 19:29:54.217550)
Something happened in 45520 ! (LAMPOR_TTIME=3, LOCAL_TTIME=2023-02-15 19:29:54.217550)
Message received at 45520 (LAMPOR_TTIME=5, LOCAL_TTIME=2023-02-15 19:29:54.221674)
Something happened in 45520 ! (LAMPOR_TTIME=6, LOCAL_TTIME=2023-02-15 19:29:54.221674)
Message received at 9892 (LAMPOR_TTIME=6, LOCAL_TTIME=2023-02-15 19:29:54.228327)
Message sent from 9892 (LAMPOR_TTIME=7, LOCAL_TTIME=2023-02-15 19:29:54.228327)
Message received at 44872 (LAMPOR_TTIME=3, LOCAL_TTIME=2023-02-15 19:29:54.220503)
Message sent from 44872 (LAMPOR_TTIME=4, LOCAL_TTIME=2023-02-15 19:29:54.221556)
Message sent from 44872 (LAMPOR_TTIME=5, LOCAL_TTIME=2023-02-15 19:29:54.221556)
Message received at 44872 (LAMPOR_TTIME=8, LOCAL_TTIME=2023-02-15 19:29:54.228327)
```

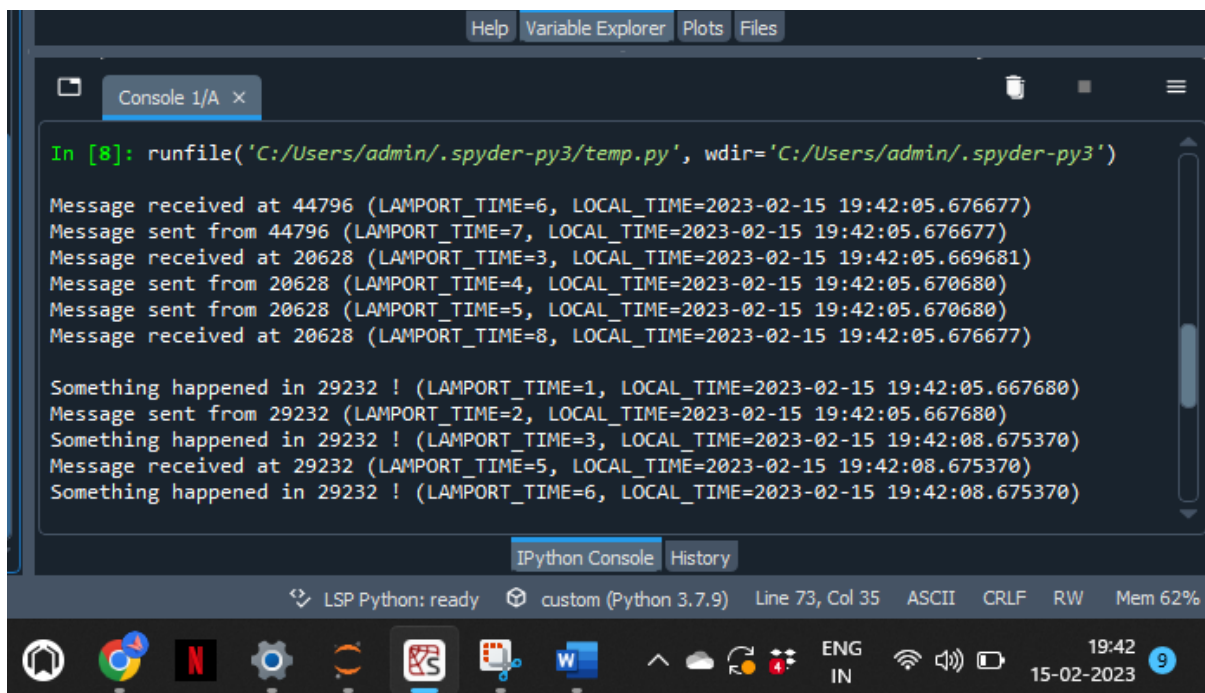
## ii) Execution of process



The screenshot shows the Variable Explorer in Spyder, displaying a list of variables and their values. The variables are: oneandtwo, process1, process2, process3, threeandtwo, twoandone, and twoandthree. Each variable is of type connection.PipeConnection or context.Process, and its size is 1. The values are PipeConnection objects or Process objects of multiprocessing modules.

Name	Type	Size	Value
oneandtwo	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
process1	context.Process	1	Process object of multiprocessing.context module
process2	context.Process	1	Process object of multiprocessing.context module
process3	context.Process	1	Process object of multiprocessing.context module
threeandtwo	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
twoandone	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module
twoandthree	connection.PipeConnection	1	PipeConnection object of multiprocessing.connection module

iii) Order of our events has changed, while our timestamps have stayed the same.



The screenshot shows the IPython Console with the output of a runfile command. The output displays a sequence of messages received and sent, along with timestamps and LAMPORT\_TIME values. The messages are: Message received at 44796 (LAMPORT\_TIME=6, LOCAL\_TIME=2023-02-15 19:42:05.676677), Message sent from 44796 (LAMPORT\_TIME=7, LOCAL\_TIME=2023-02-15 19:42:05.676677), Message received at 20628 (LAMPORT\_TIME=3, LOCAL\_TIME=2023-02-15 19:42:05.669681), Message sent from 20628 (LAMPORT\_TIME=4, LOCAL\_TIME=2023-02-15 19:42:05.670680), Message sent from 20628 (LAMPORT\_TIME=5, LOCAL\_TIME=2023-02-15 19:42:05.670680), Message received at 20628 (LAMPORT\_TIME=8, LOCAL\_TIME=2023-02-15 19:42:05.676677), Something happened in 29232 ! (LAMPORT\_TIME=1, LOCAL\_TIME=2023-02-15 19:42:05.667680), Message sent from 29232 (LAMPORT\_TIME=2, LOCAL\_TIME=2023-02-15 19:42:05.667680), Something happened in 29232 ! (LAMPORT\_TIME=3, LOCAL\_TIME=2023-02-15 19:42:08.675370), Message received at 29232 (LAMPORT\_TIME=5, LOCAL\_TIME=2023-02-15 19:42:08.675370), and Something happened in 29232 ! (LAMPORT\_TIME=6, LOCAL\_TIME=2023-02-15 19:42:08.675370).

```
In [8]: runfile('C:/Users/admin/.spyder-py3/temp.py', wdir='C:/Users/admin/.spyder-py3')

Message received at 44796 (LAMPORT_TIME=6, LOCAL_TIME=2023-02-15 19:42:05.676677)
Message sent from 44796 (LAMPORT_TIME=7, LOCAL_TIME=2023-02-15 19:42:05.676677)
Message received at 20628 (LAMPORT_TIME=3, LOCAL_TIME=2023-02-15 19:42:05.669681)
Message sent from 20628 (LAMPORT_TIME=4, LOCAL_TIME=2023-02-15 19:42:05.670680)
Message sent from 20628 (LAMPORT_TIME=5, LOCAL_TIME=2023-02-15 19:42:05.670680)
Message received at 20628 (LAMPORT_TIME=8, LOCAL_TIME=2023-02-15 19:42:05.676677)

Something happened in 29232 ! (LAMPORT_TIME=1, LOCAL_TIME=2023-02-15 19:42:05.667680)
Message sent from 29232 (LAMPORT_TIME=2, LOCAL_TIME=2023-02-15 19:42:05.667680)
Something happened in 29232 ! (LAMPORT_TIME=3, LOCAL_TIME=2023-02-15 19:42:08.675370)
Message received at 29232 (LAMPORT_TIME=5, LOCAL_TIME=2023-02-15 19:42:08.675370)
Something happened in 29232 ! (LAMPORT_TIME=6, LOCAL_TIME=2023-02-15 19:42:08.675370)
```