

# GOA COLLEGE OF ENGINEERING

“Bhausahab Bandodkar Technical Education Complex”

## **Experiment No: 5**

### **Threads**

**Aim:** To implement threads in Distributed Operating systems.

### **Theory:**

A thread is a single sequential flow of execution of tasks of a process so it is also known as thread of execution or thread of control. There is a way of thread execution inside the process of any operating system. Apart from this, there can be more than one thread inside a process. Each thread of the same process makes use of a separate program counter and a stack of activation records and control blocks. Thread is often referred to as a lightweight process.

### **User-level thread**

The operating system does not recognize the user-level thread. User threads can be easily implemented and it is implemented by the user. If a user performs a user-level thread blocking operation, the whole process is blocked. The kernel level thread does not know nothing about the user level thread. The kernel-level thread manages user-level threads as if they are single-threaded processes Java thread, POSIX threads, etc.

### **Kernel level thread**

The kernel thread recognizes the operating system. There is a thread control block and process control block in the system for each thread and process in the kernel-level thread. The kernel-level thread is implemented by the operating system. The kernel knows about all the threads and manages them. The kernel-level thread offers a system call to create and manage the threads from user-space. The implementation of kernel threads is more difficult than the user thread.

Any thread has the following components.

1. Program counter
2. Register set
3. Stack space

# GOA COLLEGE OF ENGINEERING

“Bhausahab Bhandodkar Technical Education Complex”

## **Program:**

import threading

# global variable x

x = 0

def increment():

"""

function to increment global variable x

"""

global x

x += 1

def thread\_task():

"""

task for thread

calls increment function 100000 times.

"""

for \_ in range(100000):

increment()

def main\_task():

global x

# setting global variable x as 0

x = 0

# creating threads

t1 = threading.Thread(target=thread\_task)

t2 = threading.Thread(target=thread\_task)

# start threads

t1.start()

t2.start()

# wait until threads finish their job

t1.join()

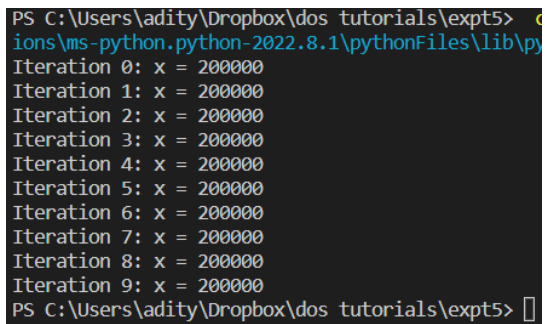
# GOA COLLEGE OF ENGINEERING

“Bhausahab Bandodkar Technical Education Complex”

```
t2.join()
```

```
if __name__ == "__main__":  
    for i in range(10):  
        main_task()  
        print("Iteration {0}: x = {1}".format(i,x))
```

## Output:



```
PS C:\Users\adity\Dropbox\dos tutorials\expt5> c  
ions\ms-python.python-2022.8.1\pythonFiles\lib\py  
Iteration 0: x = 200000  
Iteration 1: x = 200000  
Iteration 2: x = 200000  
Iteration 3: x = 200000  
Iteration 4: x = 200000  
Iteration 5: x = 200000  
Iteration 6: x = 200000  
Iteration 7: x = 200000  
Iteration 8: x = 200000  
Iteration 9: x = 200000  
PS C:\Users\adity\Dropbox\dos tutorials\expt5> █
```

**Conclusion:** Threads execution experiment was implemented and executed successfully.

Deepraj Bhosale Roll Number: 181105016 Batch-A Semester VIII