جامعة بيرزيت

BIRZEIT UNIVERSITY

First programming task

matrix multiplication

---

student name : Hamza AL Shaer

ID : 1211162

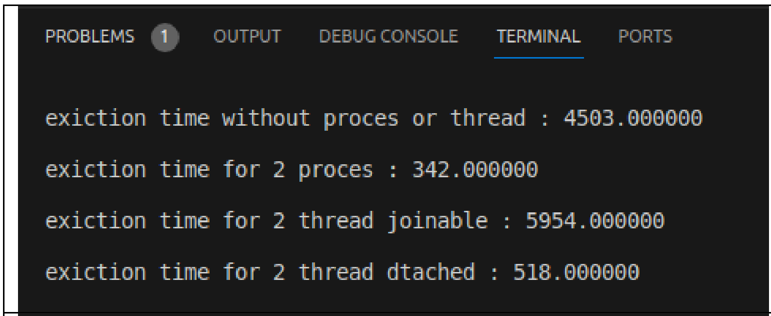
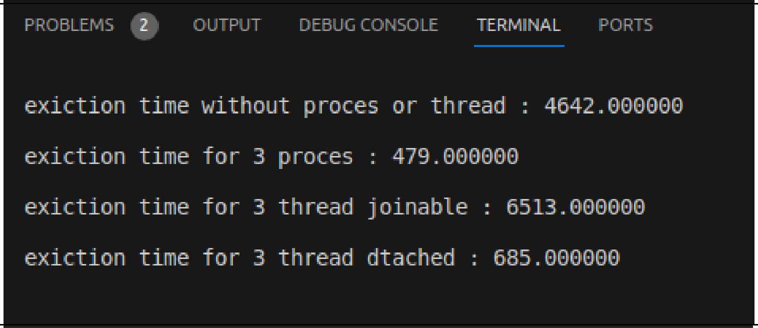
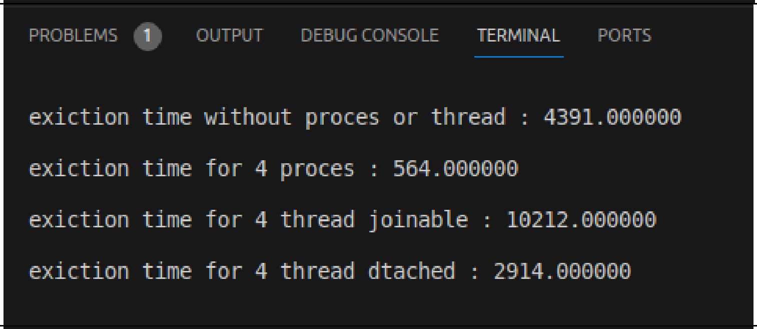
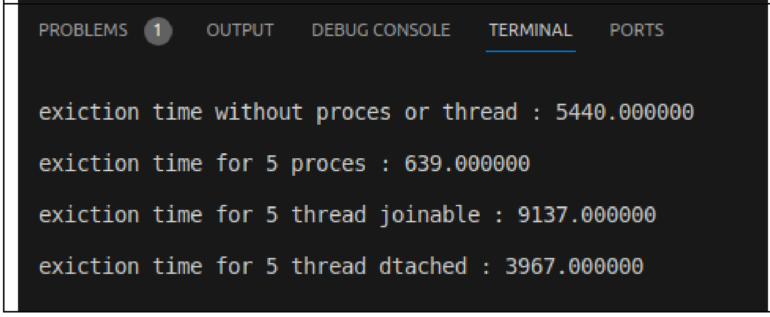
Instructor: Bashar Tahayna

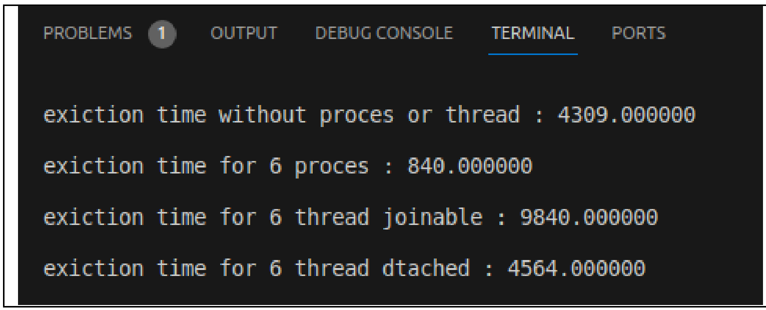
Sec : 4

- **compare the following four approachesMeasure the time it takes to complete the program in each case.**

1. **naive approach**
2. **uses multiple child processes running in parallel**
3. **uses multiple joinable threads running in parallel**
4. **uses multiple detached threads running in parallel**

all your experiments: (time is millseconds)

 exiction time without proces or thread : 4503.000000 exiction time for 2 proces : 342.000000 exiction time for 2 thread joinable : 5954.000000 exiction time for 2 thread dtached : 518.000000	**2 thread join** **2 thread detached** **2 process**
 exiction time without proces or thread : 4642.000000 exiction time for 3 proces : 479.000000 exiction time for 3 thread joinable : 6513.000000 exiction time for 3 thread dtached : 685.000000	**3 thread join** **3 thread detached** **3 process**
 exiction time without proces or thread : 4391.000000 exiction time for 4 proces : 564.000000 exiction time for 4 thread joinable : 10212.000000 exiction time for 4 thread dtached : 2914.000000	**4 thread join** **4 thread detached** **4 process**
 exiction time without proces or thread : 5440.000000 exiction time for 5 proces : 639.000000 exiction time for 5 thread joinable : 9137.000000 exiction time for 5 thread dtached : 3967.000000	**5 thread join** **5 thread detached** **5 process**

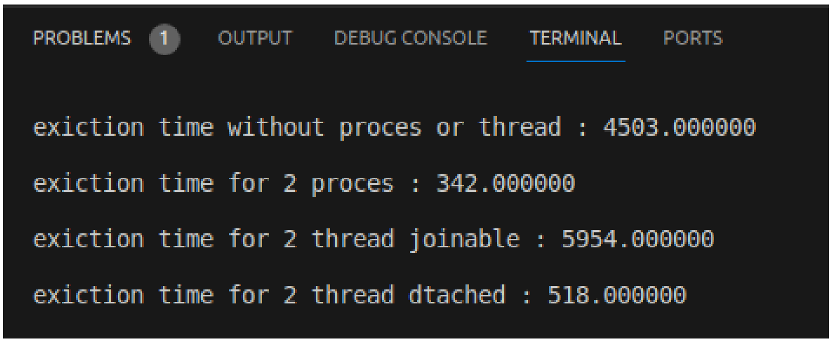


```
exiction time without proces or thread : 4309.000000

exiction time for 6 proces : 840.000000

exiction time for 6 thread joinable : 9840.000000

exiction time for 6 thread dtached : 4564.000000
```

6 thread join
6 thread detached
6 process

use for #6 for all process and threads to track the increase and decrease in execution time in each and determine the best suitable number for threads and process (virtual box linux) ..

Best case and less execution time for task , (2 of detached threads) and (2 of process) and (2 0f joinable threads) Because :



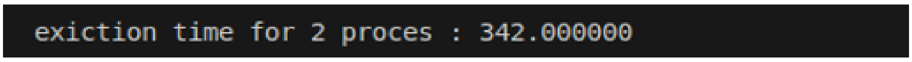
```
exiction time without proces or thread : 4503.000000
exiction time for 2 proces : 342.000000
exiction time for 2 thread joinable : 5954.000000
exiction time for 2 thread dtached : 518.000000
```

- ✓ less execution time with (process), but when increase number of process then has more execution time ,, so best case for process use 2 of process and is : 342 mill second ..

- ✓ after this best execution time is (threads detached) and it same when increase number of detached then has more execution time ,, so best case for process use 2 of detached and is : 518 mill second..

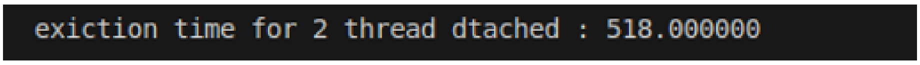
- ✓ after this best execution time is (without any process or threads) and is : 4503 mill second and is less form joinable ,, maybe because number of thread join it take some over execution time ..

- ✓ after this best execution time (threads joinable) and best case for joinable is 2 of joinable and is : 5954 mill second ..



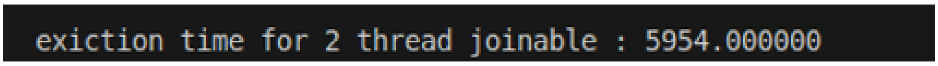
```
exiction time for 2 proces : 342.000000
```

The number of process divide for all row in matrix (100 row) ,, and each process run as parallelism with other process ,, this make process fast and fixable in multiplication, and return execution time fast and very low compered to native function, and throughput time equal 1/342 =292 sec-1.



```
exiction time for 2 thread dtached : 518.000000
```

The detached thread he is run asynchomys with main ,, main does not need to explicitly wait for all child thread to finish ,,the child can continue running independently ,,this make it fast in execution compered to joinable threads and native, and throughput time equal 1/518 =193 sec.



```
exiction time for 2 thread joinable : 5954.000000
```

The parent of joinable thread use function (pthraed_join) ,, to wait for the termination of child thread before it can proceed ,, so it not asynchomys as detached thread ,, so it take more execution time for wait the main to execute ,, and

throughput time equal 1/5954 =1679 sec-1.