multiprocessing

June 21, 2023

[ ]:
```
#Q1
```

[ ]:
```
'multiprocessing is the process where the processor contain multiple core in␣
 ↪the single core it contain multiple thread of programme the combination of␣
 ↪all the core and processing it together gives a user friendly output to the␣
 ↪user as the own memory space'
' it is useful as it can perform multitasking'
'it saves the time of the user by multi tasking '
'the multiple core can be performed together output can be easily fetched by␣
 ↪the user'
```

[ ]:
```
#Q2
```

[ ]:
```
.'multiprocessing'
.'in multiprocessor it execute the multiple processor or the core execute␣
 ↪multiple task concurrently'
.'it runs indepentently and runs on own memeory'
.'each process has the own memory '
.'the process communicate with each other using interprocess communication like␣
 ↪pipe etc'
.'each process requires its own resourse such as memory ,cpu timer etc'
.'each process run on its own address space as it can crash the error '
```

[ ]:
```
'multithreading'
.'it involves multiple threads with a single core'
.'it is lighter and share the memory space as they can communicate easily'
.'as it share the same memory space they can access and modifies the data'
.'thread can communicate easily as they can share the data easily between the␣
 ↪thread'
.'as it is sutiable for i/o bound task or overwriting the task'
```

[ ]:
```
#Q3
```

[2]:
```
import multiprocessing
import logging
logging.basicConfig(filename='multiprocess.txt',level=logging.
 ↪INFO,format='%(levelname)s-%(message)s')
```

```
def test():
    logging.info('this is my multiprocessing program')

if __name__== '__main__':
    m= multiprocessing.Process(target=test)
    logging.info('this is my main programme')
    m.start()
    m.join()
```

[ ]:
```
#Q4
```

[ ]:
```
'pool is the feature that is provided in the multiprocessing in python as the␣
 ↪pool can take process the nth number of input to gives the output'.
.'pool is used as they can take the n number of input process the information␣
 ↪gives the output'
.'these are used to work process are then used to execute together'
```

[ ]:
```
#Q5
```

[ ]:
```
1.'import multiprocessing'
2. 'def a work function that will be executed for each and work process this␣
 ↪function should take the necessary input'
3. 'creating a pool object from the microprocessor module'
4. 'use map method for the pool object to distribute the work among the work␣
 ↪processor'
5. 'map takes the two argument that the function need to be executed'
6. 'when the map method completes it gives the result'
```

[5]:
```
import multiprocessing
import logging
logging.basicConfig(filename='pool.txt',level=logging.
 ↪INFO,format='%(levelname)s-%(message)s')
def worker_function(input):
    result = input * 2
    return result
if __name__=='__main__':
    pool = multiprocessing.Pool(processes=4)
    input_data = [1,2,3,45,9]
    results = pool.map(worker_function,input_data)

    for result in results:
        logging.info(result)

    pool.close()
    pool.join()
```

[ ]:
```
#Q6
```

[6]:
```
import multiprocessing
import logging
logging.basicConfig(filename='multipro.txt',level=logging.
 ↪INFO,format='%(asctime)s-%(levelname)s-%(message)s')
def print_number(number):
    logging.info(f'process{number}:{number}')
if __name__=='__main__':
    processes =[]
    for i in range(1,5):
        process = multiprocessing.Process(target=print_number, args=(i,))
        processes.append(process)
        process.start()

    for process in processes:
        process.join()
```

[ ]:

[ ]:

[ ]: