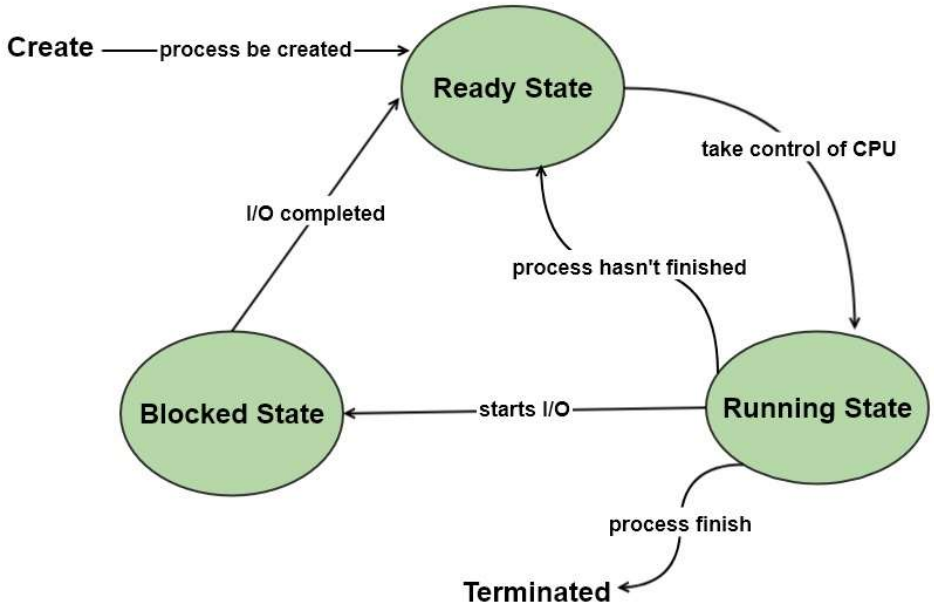
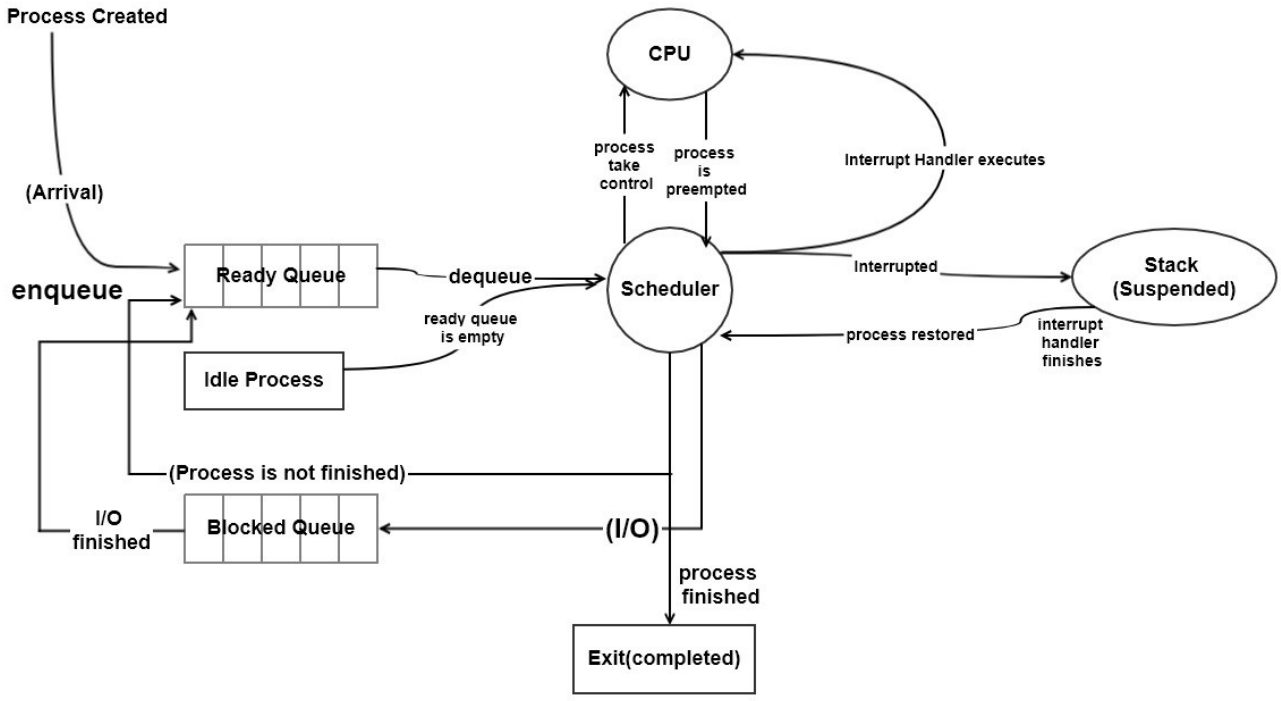
Name: Haozhe Wang Student NO: 117101193

QUESTION 1:

GRAPH:





- **What is the difference between the blocked and suspended transitions of a process?**

  1. For blocked transition, process will be added to a blocked queue as well as its state will be changed. However, for suspended transition, the process state will be saved on the stack.
  2. After I/O finishes, the process will be transmitted to ready queue. But after interrupt handler finishes, the suspended process will go to the CPU directly.

TABLE

Event	Action	Comment
Arrival (Process is created and arrive)	1. Add the process to ready queue 2. Change process state to "Ready State"	After process is created and ready to execute, process is inserted in the ready queue and is ordered by the time of arrival in the ready queue.
Interrupt	1. Suspend the running process 2. Add the process to the stack 3. Interrupt handler take control of CPU	When interrupt occurs, the process in the CPU will be preempted and saved on the stack so that the process can resume after the interrupt handler finishes. At the meantime, interrupt handler will be executed.
Interrupt Complete	1. The suspended process on the stack is resumed and take control of the CPU	After interrupt handler finished, the interrupted process on the stack is resumed and take control of the CPU.

I/O Trigger	1. The process will be preempted 2. Switch the state to “Blocked state” 3. Added to the Block Queue 4. The next ready process in ready queue takes control of CPU	When the process is running and requires I/O, the process will be preempted, switch to blocked state and be added to the blocked queue. Then the next ready process in the ready queue will take control of CPU.
I/O complete	1. Switch the process from the blocked state to the ready state 2. Process is inserted to the ready queue	I/O completion triggers the switch of the process from the blocked state to the ready state and insertion to the ready queue.
Process execution complete	1. The process will be terminated (exits the system) 2. The next process in the ready queue will take control of the CPU	If a process completes its execution during one time quanta, it will be terminated and exit the system, and the next process in the ready queue take control of the CPU
Time quanta Ends	1. The process is preempted from the CPU 2. Switch the process from the running state to the ready state 3. Add the process to the ready queue 4. The next process in the ready queue will take control of the CPU	If the process is not finished after the time quanta has expired, the process will be preempted from the CPU and goes back to ready state and be inserted to the ready queue. Then the next process in the ready queue will take control of the CPU.

Question 2

```
#Name: Haozhe Wang    Student NO: 117101193
class Queue():
    def __init__():
        self._queue=[]

    def enqueue(process):
        append process to the end of the queue

    def dequeue():
         if there is process in the Queue
            remove the first process of the queue and return the process

    def length():
        return the queue's length
    def first():
        return the first process of the queue
'
class IOoperation():
    def __init__(self,IO_startTime,IO_timeSpan):
        self._IO_startTime = IO_startTime
        self._IO_timeSpan = IO_timeSpan
    def getIO_startTime(self):
        return IO startTime
    def setIO_startTime(self,IO_startTime):
        set instance IO_startTime to IO_startTime
    def getIO_timeSpan(self):
        return    IO timeSpan
    def setIO_timeSpan(self,IO_timeSpan):
        set instance IO_timeSpan to IO_timeSpan

class Process():
    //define state varaible
    READY='ready'
    BLOCKED='blocked'
    RUNNING='running'
    CREATED='created'
    TERMINATED='terminated
    def __init__():
        self._name=''
        self._state=CREATED
        //required timeslices of the process
        self._timeRequired=0
```

```
            self._IO=IOoperation()

        def changeState(state):
            change the instance state to state
        def decreamentTime():
            deduct 1 from the process' required time slice
        def decreamentBlockTime(t):
            deduct t from the process' block time span

        def ifFinish():
            if time required is 0:
                return True
            else:
                return False
        def ifIOStart():
            if IO start time is reached
                return True
            else:
                return false
        def resetIO():
            change IO_startTime to None indicates IO finish
        def getIOSpan():
            return IO_timeSpan value
        def getName():
            return the process name

class Stack():
        def __init__():
            self._stack=[]
        def push(process):
            append process to the stack
        def pop():
            if stack length not equal to 0:
                pop the process from the stack and return the value
        def length():
            return the number of elements of the stack
        def top(self):
            if stack length not equal to 0:
                return the last element of the stack

class Interrupt():
        def __init__(self):
            self._interruptValue=get random number of 1 to 10
        def ifInterrupt(self,value):
```

```
            if interruptValue equals some value :
                return True
            return False
    def initInterrupt(self):
            generate new random number for interruptValue
    def runInterruptHandler(self):
            run interrupt handler function
class IdleProcess():
    def __init__(self):
            self._process='Idle Process'
    def runIdleProcess(self):
            run Idle Process and sleep the system(change the power on state)

class Scheduler():
    def __init__():
            initialize scheduler

            self._readyQueue=Queue()
            self._blockedQueue=Queue()
            self._interruptedStack=Stack()
            //assume this attribute represents the system is on
            self._powerOn=True
            //this attribute indicates the process running in the cpu
            self._cpu=None
            self._interrupt=Interrupt()
            self._idleProcess=IdleProcess()


    def createProcess(timeRequired,IO_startTime=None,IO_timeSpan=0):
            create a new process instance and pass arguments in the process(set the instance IO_startTime = timeRequired-
IO_startTime )
            call addReadyProcess(process)

    def powerOFF():
            change the powerOn state to False

    def addReadyProcess(process):
            add process to ready queue
            change the process state to ready state

    def addBlockedProcess():
            remember and take the process out of the cpu(call preempt function)
            add process to blocked queue
            change process to blocked state
```

```
def addSuspended():
    remember and take the process out of the cpu(call preempt function)
    add the process to interrupt stack
    run interrupt handler(change the cpu to interrupt handler process and print running interrupt handler)
    pop the interrupt stack
    resume the process(call run process function)


def runProcess():
    if there is process in the interrupt stack:
        change cpu attribute to the process
    else if ready queue is not empty:
        dequeue a process from ready queue
        change the process state to running
        change the cpu to this process
    else if ready queue is empty but there is process in block queue:
        run idle process to wait process in block queue finish
    else:
        run idle process(print running idle process and change cpu to 'idle process')
        change the powerOn state to False so that the system can sleep
def preemptProcess():
    change cpu attribute to None
    return the preempted process
def checkBlockedQueue(t):
    if there is process in blocked queue
        decrement t to first process of the block queue
        if the process IO_timeSpan less than 0:
            take the process out of the block queue
            add it to the ready queue (call function add ready process)
def schedule():
    while ready queue is not empty or block queue is not empty:
        call the function to run process( if ready queue is empty but block queue is not empty, run idle process)
        for 3 time slice of one time quanta
            decreament 1 timeslice for the first process of block queue
            if process in the cpu is not idle process:
                if process requires I/O:
                    call add blocked process function
                    break
                init interruption
                if process are interrupted:
                    call add suspended function


                process in the cpu decreament 1 time slice


                if process is not finished and time quanta ends:
```

```
                    call preempt process function
                    add process to ready queue
                elif process execution completes:
                    preempt process from cpu
                    set the process state to TERMINATED
                    break
        call runProcess to run idle process in order to powerOFF function to sleep the system
    def getSchedulerState():
        return if scheduler is running or powered off
class Test():
    def __init__():
        create Scheduler instance
        create processes(call method)
        run processes ( call method )
    def createProcess(self):
        create 4 processes with different time slices and IO requirements
    def runProcess(self):
        while Scheduler powerOn is True:
            execute Scheduler schedule method
```

Question 3

```
#Name: Haozhe Wang  Student NO: 117101193
import time
import random

TIME_QUANTA=3

'''
This class is representation of all of the queues(blocked queue, ready queue)
encapsulates the functionality of the queue
'''
class Queue():
    def __init__(self):
        self._queue = []

    # add item to the back of the queue
    def enqueue(self,process):
        self._queue.append(process)

    # remove the item from the start of the queue
    def dequeue(self):
        if self.length() != 0:
            return self._queue.pop(0)

    # return the number of elements in the queue
    def length(self):
        return len(self._queue)
    # return the first element of the queue
    def first(self):
        if self.length() !=0:
            return self._queue[0]
    # return a meaningful queue representation
    def __str__(self):
        if self.length() == 0:
            return '[empty]'
        processes=self._queue[0].__str__()
        for k in self._queue[1:]:
            processes += ' <-- '
            processes+=k.__str__()
        return processes

"""
An encapsulation of all the required states used for describing IO operations
```

```
"""
class IOoperation():
    def __init__(self,IO_startTime,IO_timeSpan):
        # represents the IO starting point(time)
        self._IO_startTime = IO_startTime
        # represents how long will the IO operation last
        self._IO_timeSpan = IO_timeSpan

    # getter of the IO_startTime property
    def getIO_startTime(self):
        return self._IO_startTime

    # setter of the IO_startTime property
    def setIO_startTime(self,IO_startTime):
        self._IO_startTime=IO_startTime

    # getter of the IO_timeSpan property
    def getIO_timeSpan(self):
        return self._IO_timeSpan

    # setter of the IO_timeSpan property
    def setIO_timeSpan(self,IO_timeSpan):
        self._IO_timeSpan=IO_timeSpan
    IO_startTime=property(getIO_startTime,setIO_startTime)
    IO_timeSpan=property(getIO_timeSpan,setIO_timeSpan)

'''
This is the representation of one process
This class includes states for the process and the process required behaviour
'''
class Process():
    # define state variables(easier for using)
    READY = 'ready'
    BLOCKED = 'blocked'
    RUNNING = 'running'
    CREATED = 'created'
    TERMINATED = 'terminated'
    def __init__(self,id,timeRequired,IO_startTime,IO_timeSpan):
        self._id=id
        self._state=Process.CREATED
        #required timeslices of the process
        self._timeRequired=timeRequired
        #create description of the process IO operation
        self._IO=IOoperation(IO_startTime,IO_timeSpan)
```

```
    #change the process state(READY,BLOCKED...)
    def changeState(self,state):
        self._state=state

    #decreament required time slice of the process
    def decreamentTime(self):
        self._timeRequired-=1

    #decreament the remaining time of the process' IO operation
    def decreamentBlockTime(self,t):
        self._IO._IO_timeSpan-=t

    #return true if the process has finished
    def ifFinish(self):
        if self._timeRequired <= 0:
            return True
        else:
            return False

    #return true if the process now needs IO operation
    def ifIOStart(self):
        if self._IO._IO_startTime == self._timeRequired:
            return True
        else:
            return False

    #after IO operation finshed, set the IO start time to negative as a representation of
finished IO
    def resetIO(self):
        #indicates IO finish
        self._IO._IO_startTime=-1

    # getter of the IO_timeSpan property
    def getIOSpan(self):
        return self._IO._IO_timeSpan

    # get the process Id
    def getId(self):
        return self._id

    # return a meaningful representation of the process
    def __str__(self):
        blockMsg=''
        if self._state==Process.BLOCKED:
```

```
            blockMsg=' | Block time remaining: %d'%(self._IO.IO_timeSpan)
        return '%s(runtime remaining: %d%s)'%(self._id,self._timeRequired,blockMsg)

    IO_timeSpan=property(getIOSpan)
    id=property(getId)

'''
This class represents the Interrupt event
The class will simulate how interrupt be generated, and provide the interrupt handler
function
'''
class Interrupt():
    def __init__(self):
        #interrupt value represents a random outside condition
        self._interruptValue=random.randint(1,10)

    #when the interrupt value equals to certain number, return true, representing the
interrupt should occur
    def ifInterrupt(self,value):
        if self._interruptValue == value:
            return True
        return False

    # reset the interrupt value, as representation of changing of outside world
    def initInterrupt(self):
        self._interruptValue = random.randint(1, 10)

    #run interrupt handler
    def runInterruptHandler(self):
        print('Running Interrupt Handler')
        time.sleep(1)

'''
This class represents the idle process
The class will provide idle process' functionality
'''
class IdleProcess():
    def __init__(self):
        self._process='Idle Process'

    # return the representation of idle process
    def __str__(self):
        return self._process
```

```
    # function of running idle process
    def runIdleProcess(self):
        print('Idle Process is running...')

'''
This is a stack class (representation of a memory space for suspended processes)
'''
class Stack():
    def __init__(self):
        self._stack=[]

    # push an item onto the top of the stack
    def push(self,process):
        self._stack.append(process)

    # pop an item from the top of the stack
    def pop(self):
        if self.length() != 0:
            process=self._stack.pop()
            return process

    # return the number of elements of the stack
    def length(self):
        return len(self._stack)

    #return the top element of the stack
    def top(self):
        if self.length() != 0:
            return self._stack[-1]

    #give a meaningful output of the stack
    def __str__(self):
        if self.length() == 0:
            return '[empty]'

        processes = self._stack[0].__str__()
        for k in self._stack[1:]:
            processes += ' <-- '
            processes += k.__str__()
        return processes

'''
This is the class of scheduler
'''
```

```
class Scheduler():
    def __init__(self):
        print('Scheduler initializing...\n')
        time.sleep(1)

        self._readyQueue=Queue()
        self._blockedQueue=Queue()
        self._interruptedStack=Stack()
        #assume this attribute represents the system is on
        self._powerOn=True
        #this attribute indicates the process running in the cpu
        self._cpu=None
        self._interrupt=Interrupt()
        self._idleProcess=IdleProcess()

    # create a new process and add the process to the ready queue
    def createProcess(self,processId,timeRequired, IO_startTime=-1, IO_timeSpan=0):
        process=Process(processId,timeRequired,timeRequired-IO_startTime,IO_timeSpan)
        print('%s is created' % (process.id))
        self.addReadyProcess(process)

    # turn the powerOn state to False as a representation of shutting down the system
    def powerOFF(self):
        self._powerOn=False

    # add process to the ready queue
    def addReadyProcess(self,process):
        process.changeState(process.READY)
        self._readyQueue.enqueue(process)
        print('%s is added to the ready queue'%(process.id))

    # take out the process from the cpu and add it to the blocked queue
    def addBlockedProcess(self):
        process=self.preemptProcess()
        process.changeState(process.BLOCKED)
        self._blockedQueue.enqueue(process)
        print('%s requires I/O operation --> is moved to the block queue'%(process.id))

    # run interrupt handler and then restore the process back to the cpu
    def addSuspended(self):
        process = self.preemptProcess()
        # push the process temporary onto the interrupt stack
        self._interruptedStack.push(process)
        # interruption taking control of the cpu
```

```
            self._cpu=self._interrupt
            # run interrupt handler
            self._cpu.runInterruptHandler()
            print('\tInterruption Stack: %s' % (self._interruptedStack))
            # restore the process, which is on the interrupt stack, back to cpu
            self.runProcess()

        # this method will take charge of selecting appropriate process to the cpu
        def runProcess(self):
            # to see if there is suspended process in the interrupt stack
            if self._interruptedStack.length() != 0:
                # if there is suspended process, pop the process form the stack and restore if
back to the cpu
                self._cpu = self._interruptedStack.pop()
                # now self._cpu must have the suspended process
                print('Interrupt Handler finished --> %s is restored'%(self._cpu.id))
                print('%s is running in the CPU'%(self._cpu.id))
            # to see if there is an process ready to execute
            elif self._readyQueue.length() != 0:
                process=self._readyQueue.dequeue()
                process.changeState(process.RUNNING)
                self._cpu=process
                print('%s is moved to the CPU'%(process.id))
            # if there is no process in the ready queue, but have process in the blocked queue
            elif self._blockedQueue.length() !=0:
                self._cpu = self._idleProcess
                self._cpu.runIdleProcess()
            # if there is no any process to be executed, run idle process and terminate system
            else:
                self._cpu=self._idleProcess
                self._cpu.runIdleProcess()
                print('*******System Sleep!*******')
                self._powerOn = False

        # take out the process which is in the cpu
        def preemptProcess(self):
            process=self._cpu
            self._cpu=None
            return process

        # decrease the IO time of the first process in the blocked queue
        def checkBlockedQueue(self,t):
            process= self._blockedQueue.first()
            if process:
```

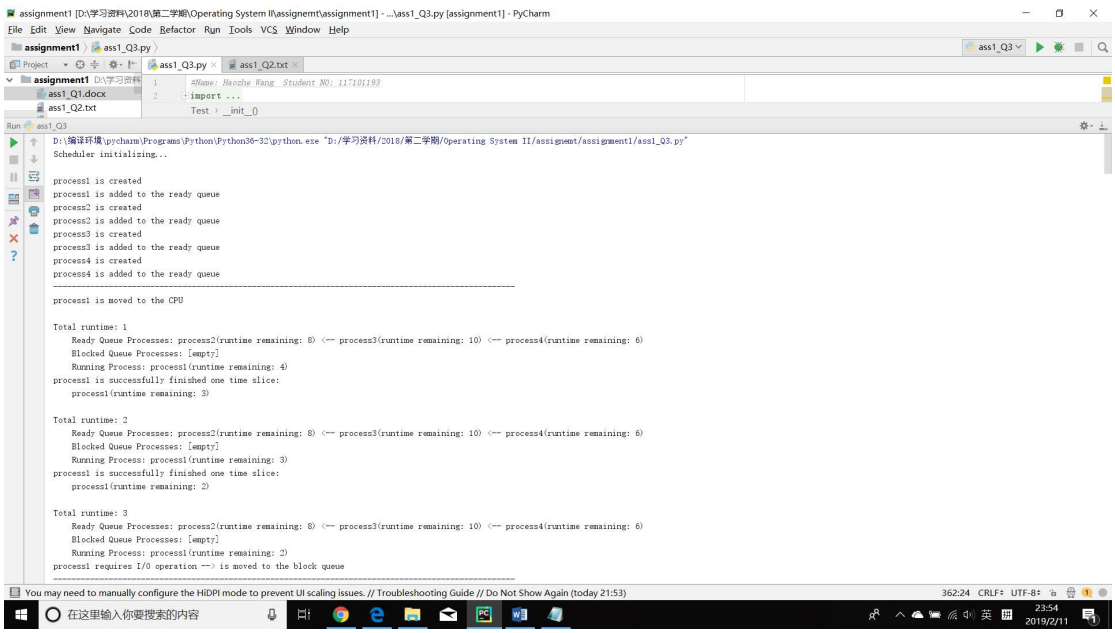
```
            process.decreamentBlockTime(t)
            # if the process IO operation is finished
            if process.IO_timeSpan <= 0:
                self._blockedQueue.dequeue()
                process.resetIO()
                self.addReadyProcess(process)
                print('%s IO finished, moved to the ready queue'%(process.id))

    # this is the main flow of the scheduler
    # this will run processes in round robin algorithm, and make every processes time-
sharing
    def schedule(self):
        # total_runTime represents the time of cpu has been running since it started
        total_runTime=0
        while self._readyQueue.length() != 0 or self._blockedQueue.length() != 0:
            print('-'*100)
            self.runProcess()
            for i in range(TIME_QUANTA):
                total_runTime += 1
                print('\nTotal runtime: %d' % (total_runTime))
                print('\tReady Queue Processes: %s' % (self._readyQueue))
                print('\tBlocked Queue Processes: %s' % (self._blockedQueue))
                print('\tRunning Process: %s' % (self._cpu))
                # decrement the IO operation time in the blocked queue
                self.checkBlockedQueue(1)
                if isinstance(self._cpu, Process):
                    # check if the current process in the cpu needs IO
                    if self._cpu.ifIOStart():
                        self.addBlockedProcess()
                        break
                    # check if interrupt occurs
                    self._interrupt.initInterrupt()
                    if self._interrupt.ifInterrupt(5):
                        self.addSuspended()

                    # the process in the cpu successfully ran one time slice
                    self._cpu.decreamentTime()
                    print('%s is successfully finished one time
slice:\n\t%s'%(self._cpu.id,self._cpu))

                    # if process haven't finished after the time quanta ends
                    if not self._cpu.ifFinish() and i == TIME_QUANTA-1:
                        print('%s is not finished' % (self._cpu.id))
                        self.addReadyProcess(self._cpu)
```

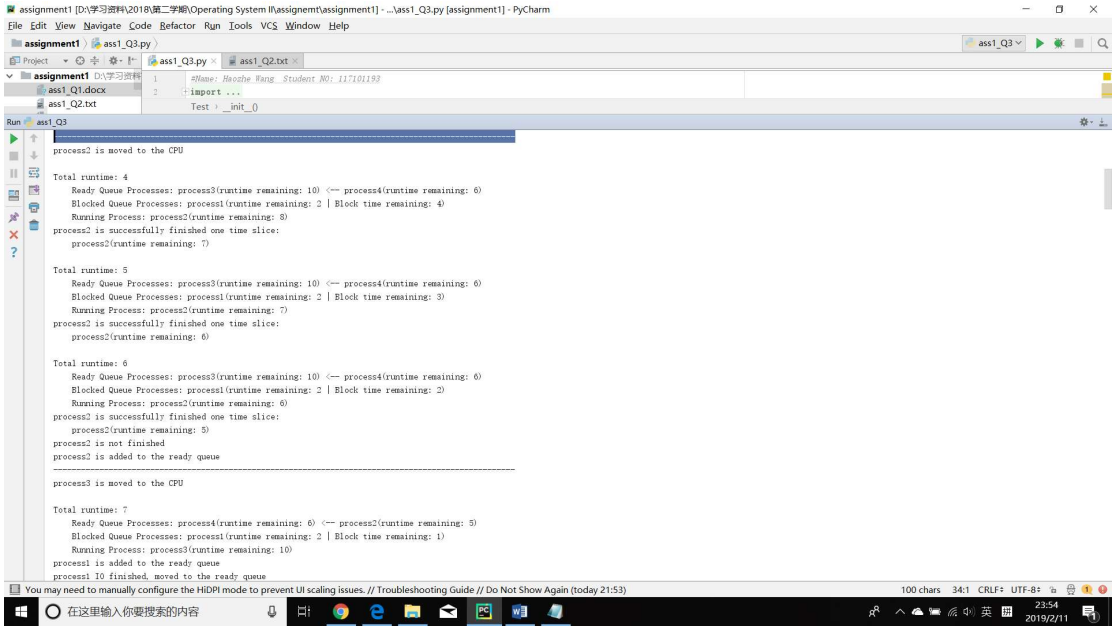
```
                    self.preemptProcess()

                # if the process finishes its execution and terminates
                elif self._cpu.ifFinish():
                    print('%s is finished' % (self._cpu.id))
                    process=self.preemptProcess()
                    process.changeState(process.TERMINATED)
                    break
        # When there is no any process will run in the cpu, cpu will run idle process and
then turn the system off
        print('='*100)
        print('=' * 100)
        self.runProcess()
    # provide an api for outside so that system will know if system can be powered off
    def getSchedulerState(self):
        return self._powerOn
```

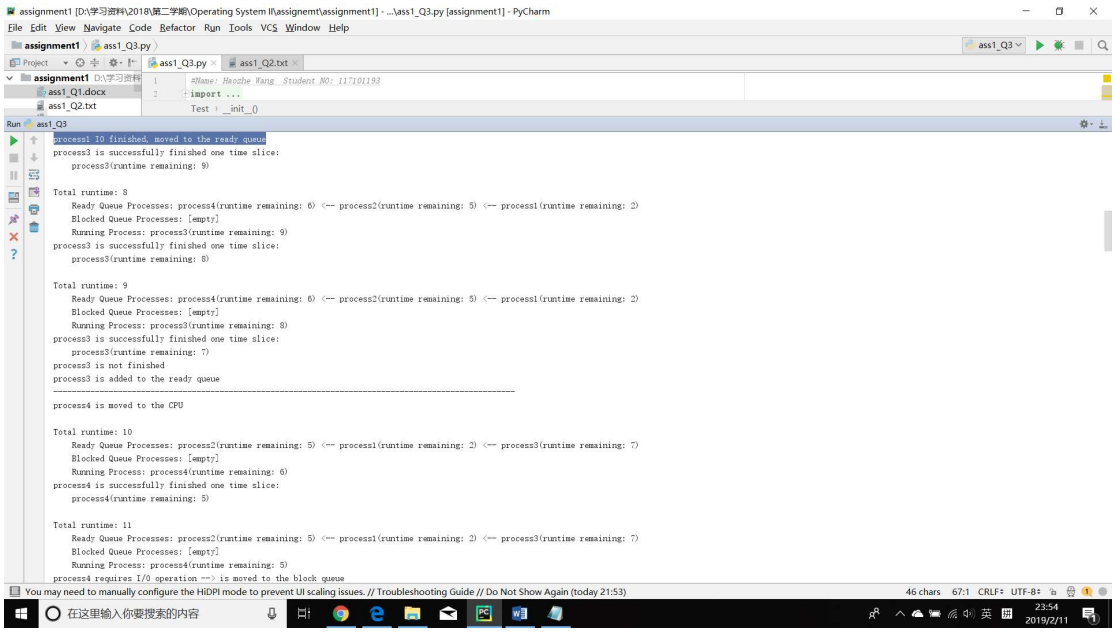
Question 4

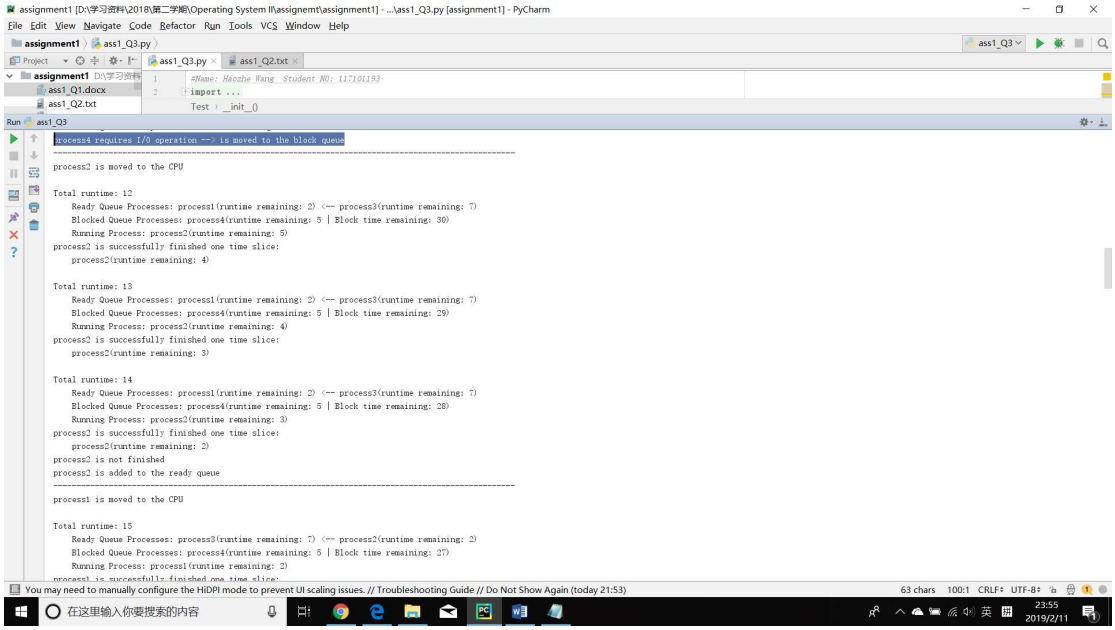
```
class Test():
    def __init__(self):
        self._scheduler = Scheduler()
        self.createProcess()
        self.runProcess()
    # create new processes
    def createProcess(self):
        self._scheduler.createProcess('process1', 4, 2, 4)
        self._scheduler.createProcess('process2', 8)
        self._scheduler.createProcess('process3', 10)
        self._scheduler.createProcess('process4', 6, 1, 30)
    # run scheduler
    def runProcess(self):
        # while the scheduler not finishes all of its operations
        while self._scheduler.getSchedulerState():
            self._scheduler.schedule()

test=Test()
```

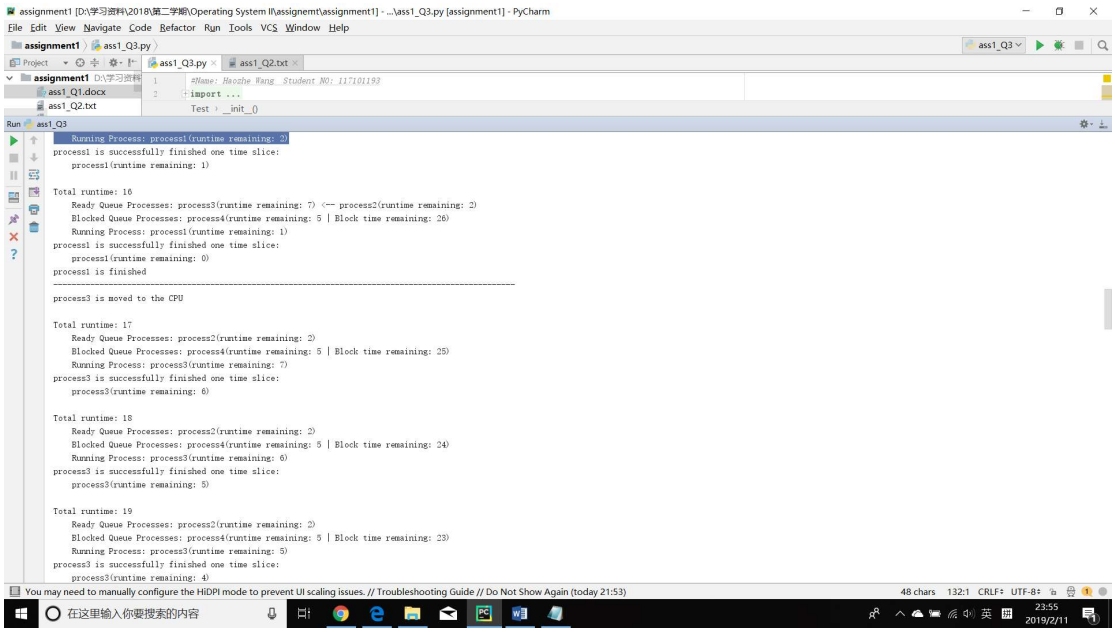
Simulation Screen Shot

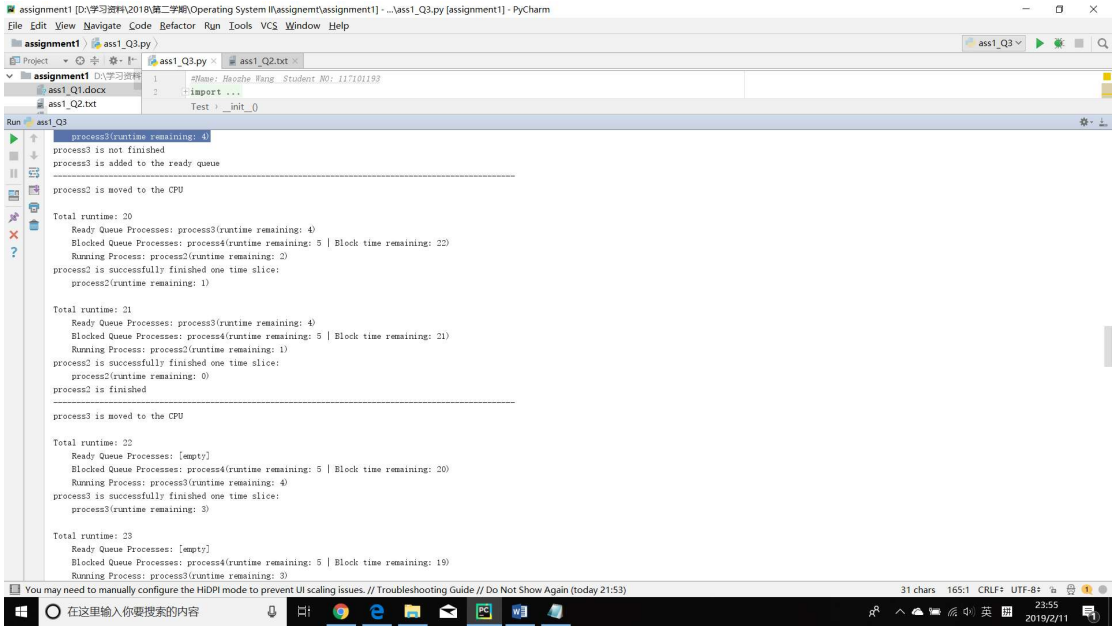


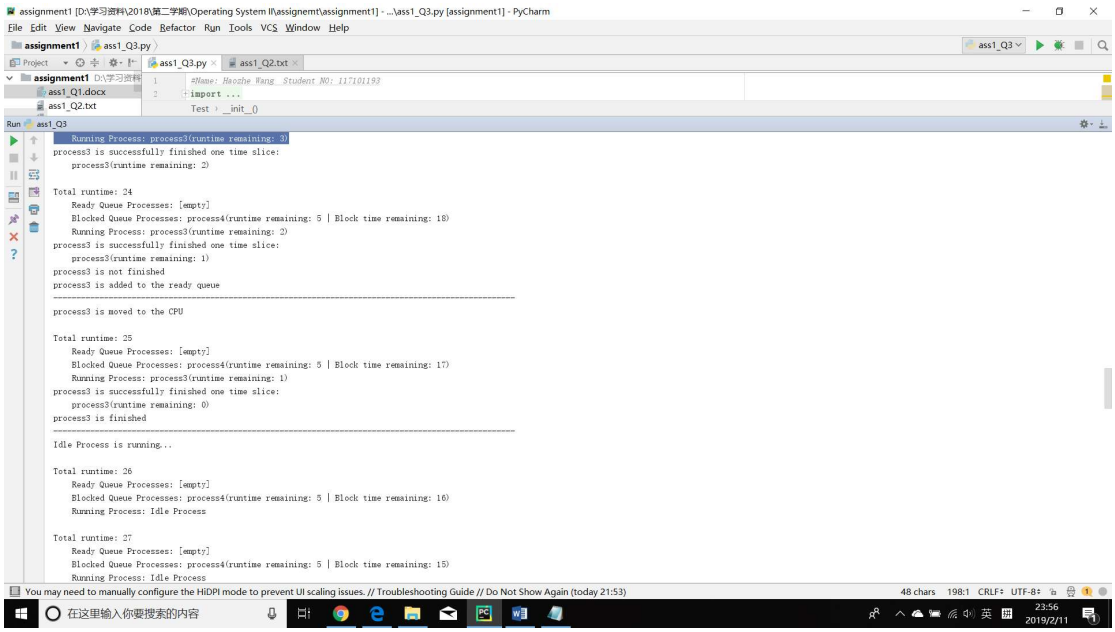

assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3
Project
ass1_Q3.py
ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
process2 is moved to the CPU
Total runtime: 4
Ready Queue Processes: process3(runtime remaining: 10) <-- process4(runtime remaining: 6)
Blocked Queue Processes: process1(runtime remaining: 2 | Block time remaining: 4)
Running Process: process2(runtime remaining: 8)
process2 is successfully finished one time slice:
process2(runtime remaining: 7)
Total runtime: 5
Ready Queue Processes: process3(runtime remaining: 10) <-- process4(runtime remaining: 6)
Blocked Queue Processes: process1(runtime remaining: 2 | Block time remaining: 3)
Running Process: process2(runtime remaining: 7)
process2 is successfully finished one time slice:
process2(runtime remaining: 6)
Total runtime: 6
Ready Queue Processes: process3(runtime remaining: 10) <-- process4(runtime remaining: 6)
Blocked Queue Processes: process1(runtime remaining: 2 | Block time remaining: 2)
Running Process: process2(runtime remaining: 6)
process2 is successfully finished one time slice:
process2(runtime remaining: 5)
process2 is not finished
process2 is added to the ready queue
process3 is moved to the CPU
Total runtime: 7
Ready Queue Processes: process4(runtime remaining: 6) <-- process2(runtime remaining: 5)
Blocked Queue Processes: process1(runtime remaining: 2 | Block time remaining: 1)
Running Process: process3(runtime remaining: 10)
process1 is added to the ready queue
process1 IO finished, moved to the ready queue
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
100 chars 34:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:54
2019/2/11

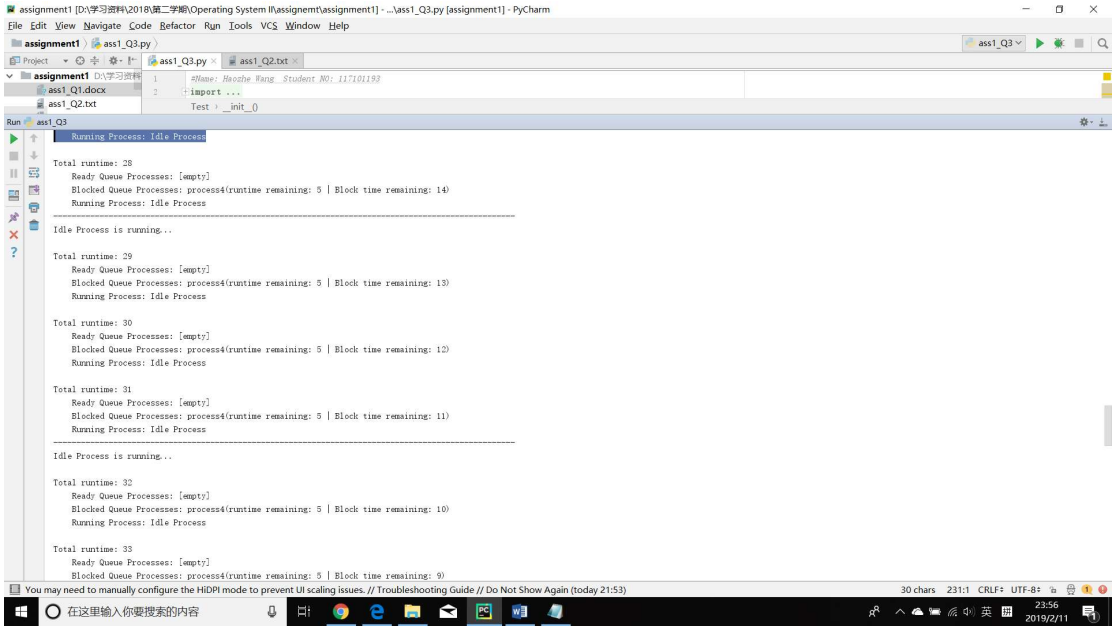

assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3
Project
ass1_Q3.py
ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
process1 IO finished, moved to the ready queue
process3 is successfully finished one time slice:
process3(runtime remaining: 9)
Total runtime: 8
Ready Queue Processes: process4(runtime remaining: 6) <-- process2(runtime remaining: 5) <-- process1(runtime remaining: 2)
Blocked Queue Processes: [empty]
Running Process: process3(runtime remaining: 9)
process3 is successfully finished one time slice:
process3(runtime remaining: 8)
Total runtime: 9
Ready Queue Processes: process4(runtime remaining: 6) <-- process2(runtime remaining: 5) <-- process1(runtime remaining: 2)
Blocked Queue Processes: [empty]
Running Process: process3(runtime remaining: 8)
process3 is successfully finished one time slice:
process3(runtime remaining: 7)
process3 is not finished
process3 is added to the ready queue
process4 is moved to the CPU
Total runtime: 10
Ready Queue Processes: process2(runtime remaining: 5) <-- process1(runtime remaining: 2) <-- process3(runtime remaining: 7)
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 6)
process4 is successfully finished one time slice:
process4(runtime remaining: 5)
Total runtime: 11
Ready Queue Processes: process2(runtime remaining: 5) <-- process1(runtime remaining: 2) <-- process3(runtime remaining: 7)
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 5)
process4 requires I/O operation --> is moved to the block queue
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
46 chars 67:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:54
2019/2/11

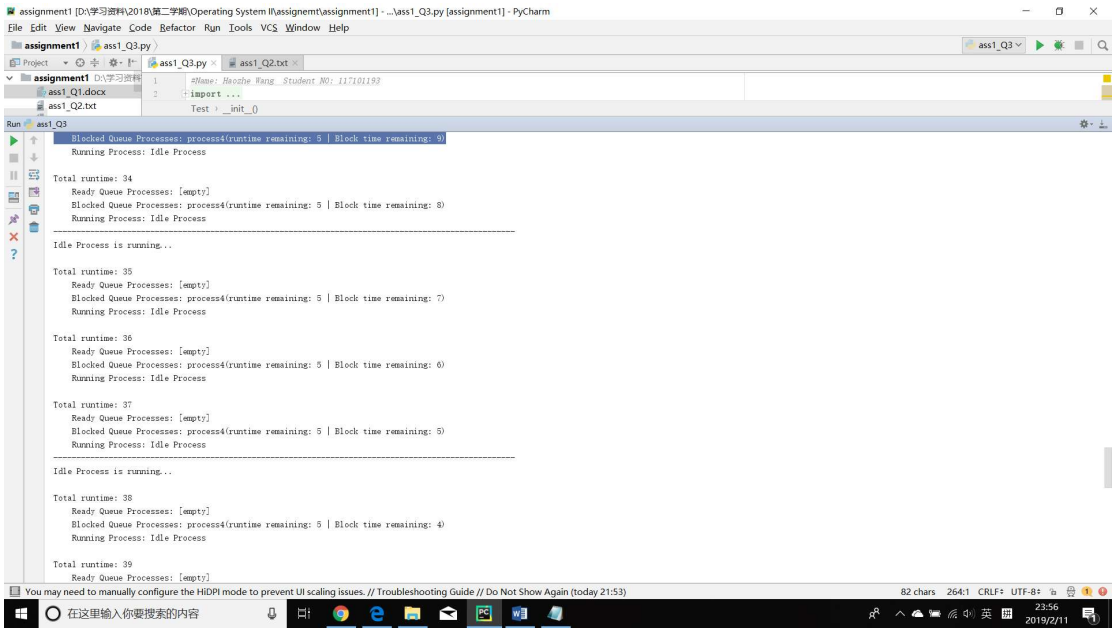

assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3
Project
ass1_Q3.py
ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
process4 requires I/O operation --> is moved to the block queue
process2 is moved to the CPU
Total runtime: 12
Ready Queue Processes: process1(runtime remaining: 2) <-- process3(runtime remaining: 7)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 30)
Running Process: process2(runtime remaining: 5)
process2 is successfully finished one time slice:
process2(runtime remaining: 4)
Total runtime: 13
Ready Queue Processes: process1(runtime remaining: 2) <-- process3(runtime remaining: 7)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 29)
Running Process: process2(runtime remaining: 4)
process2 is successfully finished one time slice:
process2(runtime remaining: 3)
Total runtime: 14
Ready Queue Processes: process1(runtime remaining: 2) <-- process3(runtime remaining: 7)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 28)
Running Process: process2(runtime remaining: 3)
process2 is successfully finished one time slice:
process2(runtime remaining: 2)
process2 is not finished
process2 is added to the ready queue
process1 is moved to the CPU
Total runtime: 15
Ready Queue Processes: process3(runtime remaining: 7) <-- process2(runtime remaining: 2)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 27)
Running Process: process1(runtime remaining: 2)
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
63 chars 100:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:55
2019/2/11

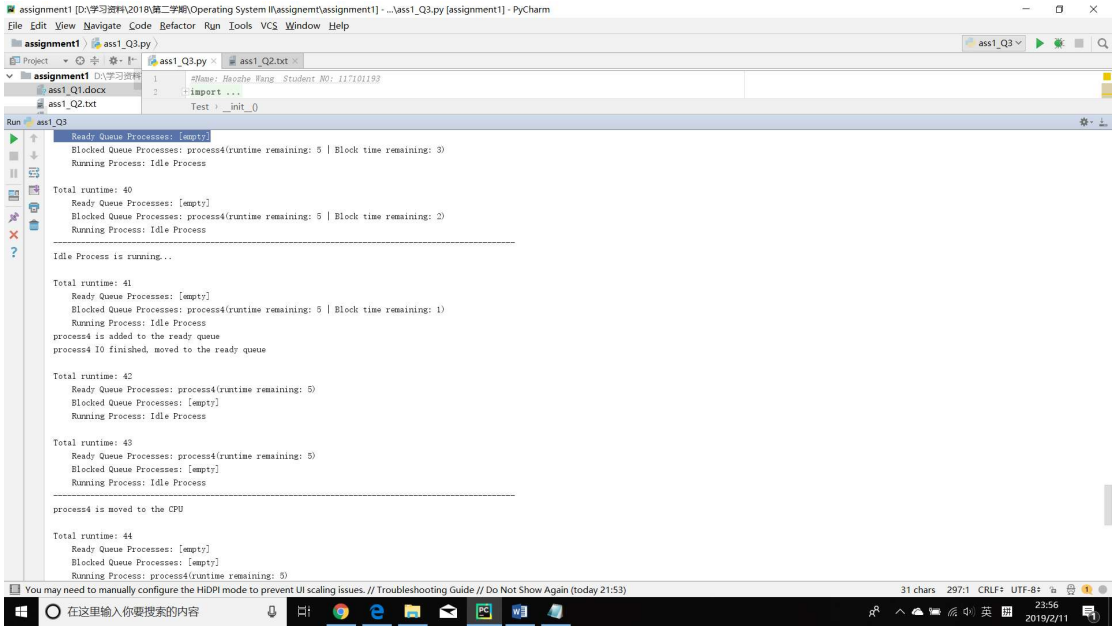

assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3
Project
ass1_Q3.py
ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
Running Process: process1(runtime remaining: 2)
process1 is successfully finished one time slice:
process1(runtime remaining: 1)
Total runtime: 16
Ready Queue Processes: process3(runtime remaining: 7) <-- process2(runtime remaining: 2)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 26)
Running Process: process1(runtime remaining: 1)
process1 is successfully finished one time slice:
process1(runtime remaining: 0)
process1 is finished
process3 is moved to the CPU
Total runtime: 17
Ready Queue Processes: process2(runtime remaining: 2)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 25)
Running Process: process3(runtime remaining: 7)
process3 is successfully finished one time slice:
process3(runtime remaining: 6)
Total runtime: 18
Ready Queue Processes: process2(runtime remaining: 2)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 24)
Running Process: process3(runtime remaining: 6)
process3 is successfully finished one time slice:
process3(runtime remaining: 5)
Total runtime: 19
Ready Queue Processes: process2(runtime remaining: 2)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 23)
Running Process: process3(runtime remaining: 5)
process3 is successfully finished one time slice:
process3(runtime remaining: 4)
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
48 chars 132:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:55
2019/2/11

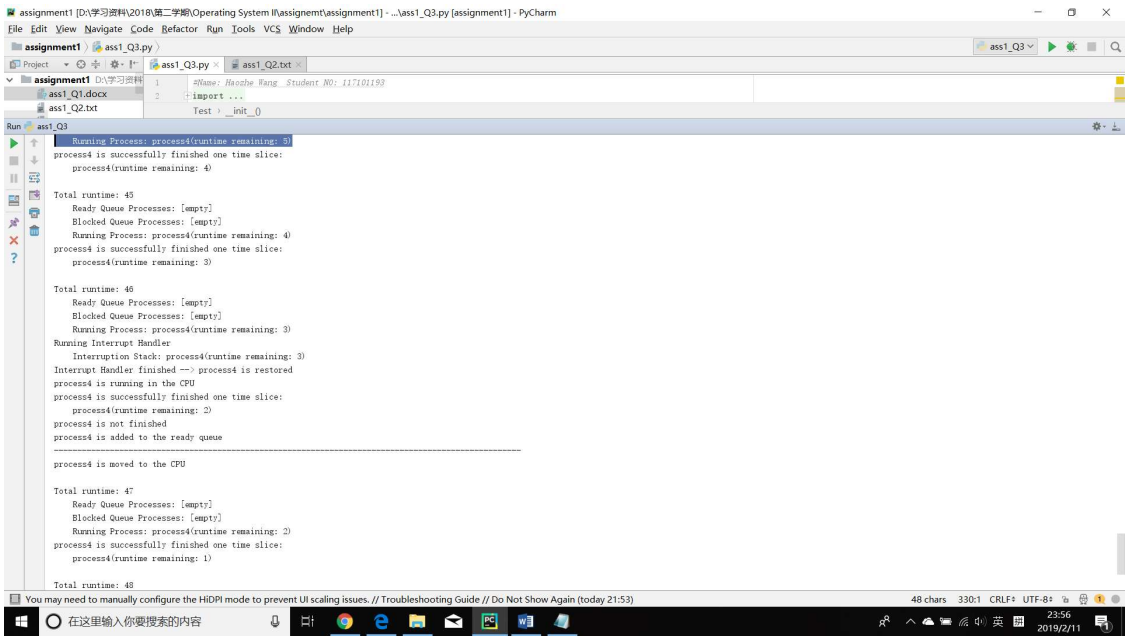

assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3.py ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
process3(runtime remaining: 4)
process3 is not finished
process3 is added to the ready queue
process2 is moved to the CPU
Total runtime: 20
Ready Queue Processes: process3(runtime remaining: 4)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 22)
Running Process: process2(runtime remaining: 2)
process2 is successfully finished one time slice:
process2(runtime remaining: 1)
Total runtime: 21
Ready Queue Processes: process3(runtime remaining: 4)
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 21)
Running Process: process2(runtime remaining: 1)
process2 is successfully finished one time slice:
process2(runtime remaining: 0)
process2 is finished
process3 is moved to the CPU
Total runtime: 22
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 20)
Running Process: process3(runtime remaining: 4)
process3 is successfully finished one time slice:
process3(runtime remaining: 3)
Total runtime: 23
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 19)
Running Process: process3(runtime remaining: 3)
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
31 chars 165:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:55
2019/2/11

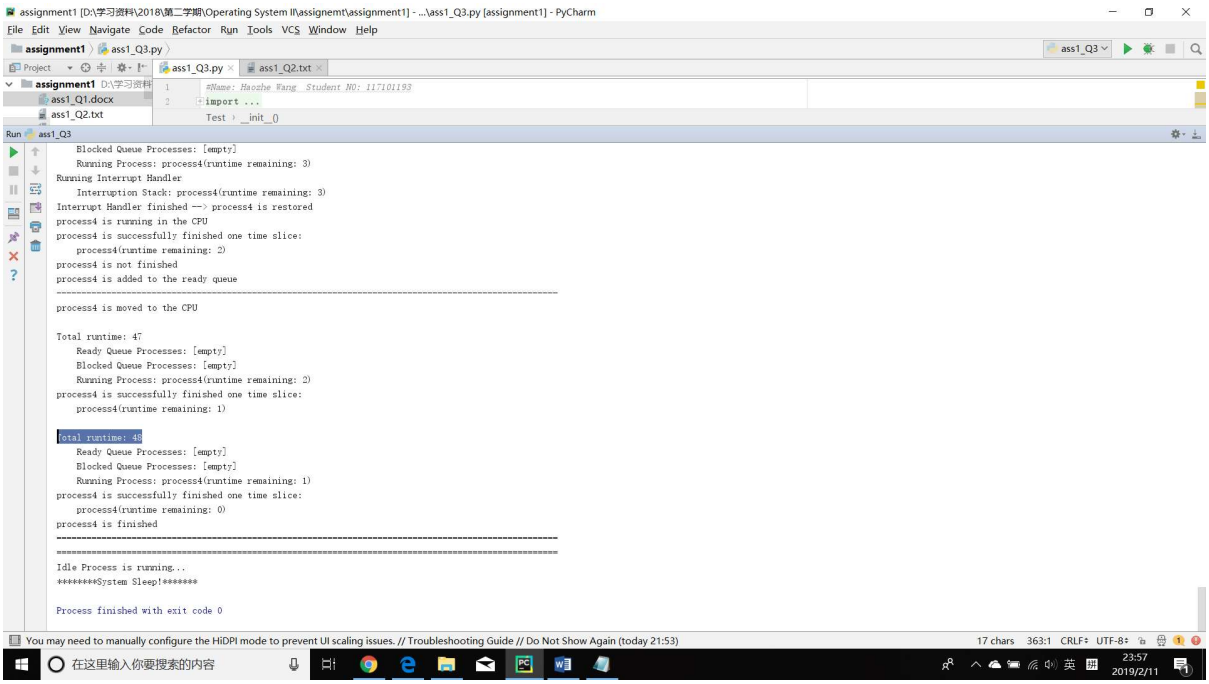

assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3.py ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
Running Process: process3(runtime remaining: 3)
process3 is successfully finished one time slice:
process3(runtime remaining: 2)
Total runtime: 24
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 18)
Running Process: process3(runtime remaining: 2)
process3 is successfully finished one time slice:
process3(runtime remaining: 1)
process3 is not finished
process3 is added to the ready queue
process3 is moved to the CPU
Total runtime: 25
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 17)
Running Process: process3(runtime remaining: 1)
process3 is successfully finished one time slice:
process3(runtime remaining: 0)
process3 is finished
Idle Process is running...
Total runtime: 26
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 16)
Running Process: Idle Process
Total runtime: 27
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 15)
Running Process: Idle Process
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
48 chars 198:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:56
2019/2/11


assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3.py ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
Running Process: Idle Process
Total runtime: 28
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 14)
Running Process: Idle Process
Idle Process is running...
Total runtime: 29
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 13)
Running Process: Idle Process
Total runtime: 30
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 12)
Running Process: Idle Process
Total runtime: 31
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 11)
Running Process: Idle Process
Idle Process is running...
Total runtime: 32
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 10)
Running Process: Idle Process
Total runtime: 33
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 9)
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
30 chars 231:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:56
2019/2/11


assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 ass1_Q3.py
ass1_Q3.py ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
#Name: Haozhe Wang Student NO: 117101193
import ...
Test _init_()
Run ass1_Q3
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 9)
Running Process: Idle Process
Total runtime: 34
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 8)
Running Process: Idle Process
Idle Process is running...
Total runtime: 35
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 7)
Running Process: Idle Process
Total runtime: 36
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 6)
Running Process: Idle Process
Total runtime: 37
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 5)
Running Process: Idle Process
Idle Process is running...
Total runtime: 38
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 4)
Running Process: Idle Process
Total runtime: 39
Ready Queue Processes: [empty]
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
82 chars 264:1 CRLF: UTF-8:
在这里输入你要搜索的内容
23:56
2019/2/11


assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
#Name: Haozhe Wang Student NO: 11710193
import ...
Test › __init__()
Run ass1_Q3
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 3)
Running Process: Idle Process
Total runtime: 40
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 2)
Running Process: Idle Process
Idle Process is running...
Total runtime: 41
Ready Queue Processes: [empty]
Blocked Queue Processes: process4(runtime remaining: 5 | Block time remaining: 1)
Running Process: Idle Process
process4 is added to the ready queue
process4 IO finished, moved to the ready queue
Total runtime: 42
Ready Queue Processes: process4(runtime remaining: 5)
Blocked Queue Processes: [empty]
Running Process: Idle Process
Total runtime: 43
Ready Queue Processes: process4(runtime remaining: 5)
Blocked Queue Processes: [empty]
Running Process: Idle Process
process4 is moved to the CPU
Total runtime: 44
Ready Queue Processes: [empty]
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 5)
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
31 chars 297:1 CRLF UTF-8
在这里输入你要搜索的内容
23:56
2019/2/11


assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
#Name: Haozhe Wang Student NO: 11710193
import ...
Test › __init__()
Run ass1_Q3
Running Process: process4(runtime remaining: 5)
process4 is successfully finished one time slice:
process4(runtime remaining: 4)
Total runtime: 45
Ready Queue Processes: [empty]
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 4)
process4 is successfully finished one time slice:
process4(runtime remaining: 3)
Total runtime: 46
Ready Queue Processes: [empty]
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 3)
Running Interrupt Handler
Interruption Stack: process4(runtime remaining: 3)
Interrupt Handler finished --> process4 is restored
process4 is running in the CPU
process4 is successfully finished one time slice:
process4(runtime remaining: 2)
process4 is not finished
process4 is added to the ready queue
process4 is moved to the CPU
Total runtime: 47
Ready Queue Processes: [empty]
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 2)
process4 is successfully finished one time slice:
process4(runtime remaining: 1)
Total runtime: 48
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
48 chars 330:1 CRLF UTF-8
在这里输入你要搜索的内容
23:56
2019/2/11


assignment1 [D:\学习资料\2018\第二学期\Operating System II\assignemt\assignment1] - ...\ass1_Q3.py [assignment1] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
assignment1 〉 ass1_Q3.py 〉
ass1_Q3
Project
ass1_Q3.py
ass1_Q2.txt
assignment1 D:\学习资料
ass1_Q1.docx
ass1_Q2.txt
1 #Name: Haozhe Wang Student NO: 117101193
2 import ...
Test 〉 __init__()
Run ass1_Q3
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 3)
Running Interrupt Handler
Interruption Stack: process4(runtime remaining: 3)
Interrupt Handler finished --> process4 is restored
process4 is running in the CPU
process4 is successfully finished one time slice:
process4(runtime remaining: 2)
process4 is not finished
process4 is added to the ready queue
process4 is moved to the CPU
Total runtime: 47
Ready Queue Processes: [empty]
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 2)
process4 is successfully finished one time slice:
process4(runtime remaining: 1)
Total runtime: 48
Ready Queue Processes: [empty]
Blocked Queue Processes: [empty]
Running Process: process4(runtime remaining: 1)
process4 is successfully finished one time slice:
process4(runtime remaining: 0)
process4 is finished
Idle Process is running...
*******System Sleep!*******
Process finished with exit code 0
You may need to manually configure the HiDPI mode to prevent UI scaling issues. // Troubleshooting Guide // Do Not Show Again (today 21:53)
17 chars 363:1 CRLF UTF-8
在这里输入你要搜索的内容
英
23:57
2019/2/11