Documentation

❖ Problem Statement:

- Task 1: Design your own I/O simulating input(read)/output(write) operations.
- Task 2: CFS Scheduling using RBTree in Nachos.

❖ Design/solution for each requirement

- Task 1:
  - To simulate IO read and write operation I have designed a class ioHandler in ioHandler.h file where the read write operations are simulated (a string is read from a buffer for read operation and for write operation a string is written into a buffer).
  - Inorder to generate IO requests the class ioRequest in the file ioRequest.h file is used. It responsible for generating IO requests (read/write). The structure of this request is defined in the ioRequestStructure.h file where the type of interrupt is set, and a random waiting time for the request is generated. Once a request is generated it is added to a list (reqList) present in the kernel.h file and an alarm is set for the randomly generated waiting time which simulates the request by invoking the callback method in the IOalarm.h class thereby triggering the simulation methods in ioHandler class which we discussed in the previous point.
  - Since this tasks tells us to create our own IO simulated interrupts, I have defined the two new IO interrupts in the file interrupt.h at line 51 in the enum as 'ReadInt' and 'WriteInt' for read and write interrupt respectively.

- Task 2:
  - For implementing CFS scheduler I have implemented the RBT. The implementation is adopted from a repo on GitHub. I have added couple of extra methods required for the implementation of CFS in the same code of RBTree.
  - The time quantum for each thread is the ratio of kernel->stats->totalTicks divided by the total number of threads available for execution in the RB tree. The value of totalTicks is reset from 100 to 1000. So now 1000 will be equally divided into n number of threads available for execution in RB tree. After the time slice for a process is calculated an interrupt is set.

- This interrupt is used to preempt the thread after its time slice is elapsed thus bringing the next thread from the RB tree for execution.
- After a thread has completed executing for its time slice the VRuntime for that thread is updated and the wait count is reset to 1.
- After each iteration/removal of thread/node from the RB tree we increment the wait time for each thread in the RBtree that is not executed (which means it is waiting to be executed. The wait time/decay is simply a count which is used to keep a track of how long has the process been sitting the tree waiting to get executed).
- There is no need to update Vruntime of every node in the tree after removal of one node. Only incrementing the wait time for every node will work since the order/structure of the tree would remain the same even after recalculating the VRuntime after updating the wait. Thus there is no need of the extra computation of VrunTime of all nodes at every iteration.
- The virtual rum time is calculated as: execution_time*weight

  Where,

  weight=1/wait time (thus larger the wait, time lower will be the weight thus making the Vruntime smaller and bringing it to the left side of the RB tree causing it to be invoked as soon as possible)
- The VRuntime for postal worker thread is set to 10000 since that thread does not have anything to execute so we can put that node to the right side of the tree.

❖ <u>Files with directory name that you modified or created for this assignment:</u>

- nachos/code/build.linux/Makefile
- nachos/code/machine/interrupt.h
- nachos/code/machine/stats.h
- nachos/code/machine/timer.h
- nachos/code/machine/timer.cc
- nachos/code/threads/IOalarm.h
- nachos/code/threads/IOalarm.cc
- nachos/code/threads/ioHandler.h
- nachos/code/threads/ioHandler.cc
- nachos/code/threads/ioRequest.h
- nachos/code/threads/ioRequest.cc
- nachos/code/threads/ioRequestStructure.h
- nachos/code/threads/ioRequestStructure.cc
- nachos/code/threads/kernel.h

- nachos/code/threads/kernel.cc
- nachos/code/threads/rbt.h
- nachos/code/threads/rbt.cc
- nachos/code/threads/scheduler.h
- nachos/code/threads/scheduler.cc
- nachos/code/threads/thread.h
- nachos/code/threads/thread.cc
- nachos/code/threads/threadtest.cc

❖ Test cases:

To execute the test cases uncomment the following lines in the mentioned files:

Nachos/code/machine/stats.cc: line 26

Nachos/code/threads/threadtest.cc : line 66

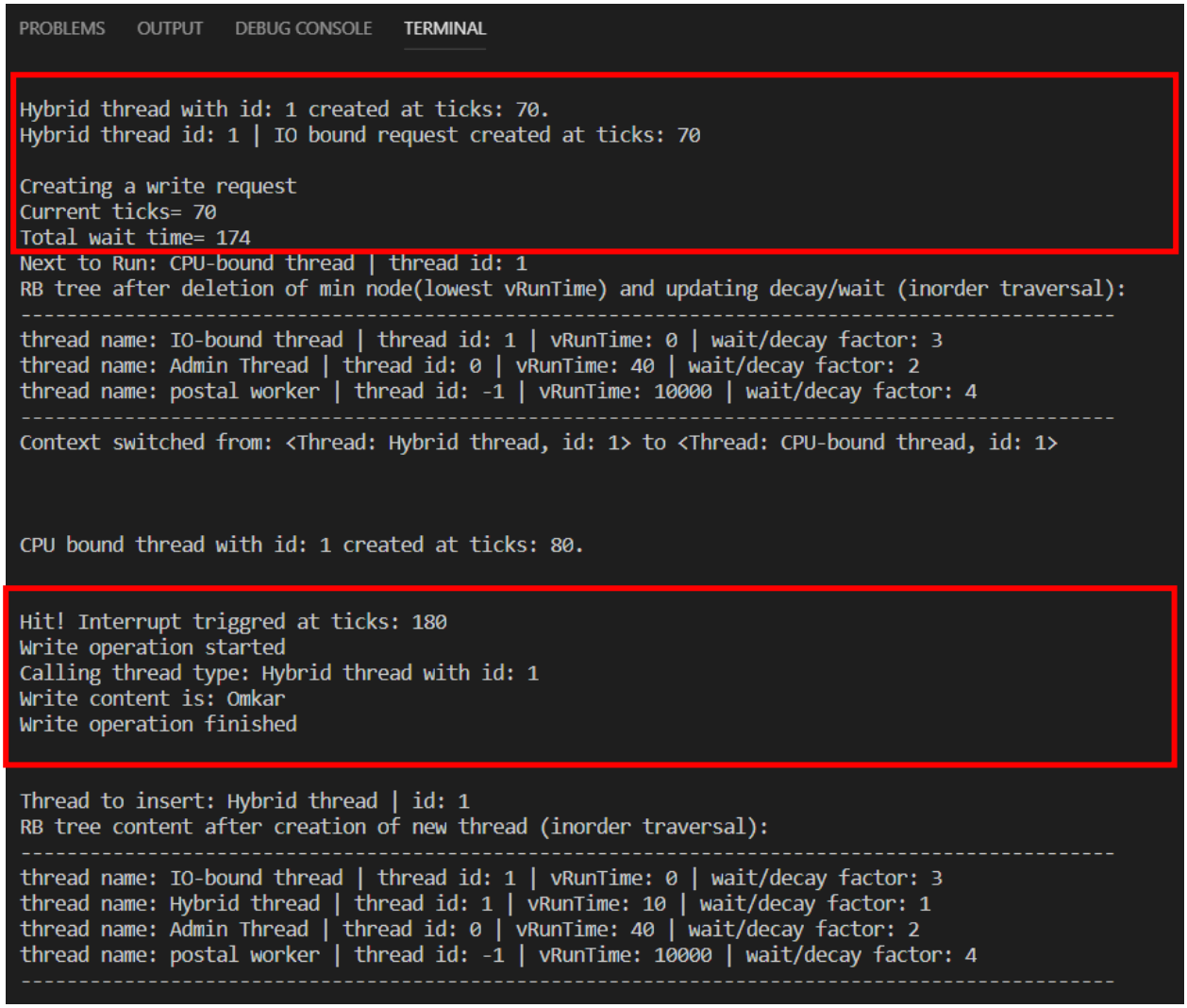
Following is the screenshots of the output for test cases. For test cases I have created 3 threads in the order- IO bound, CPU bound and Hybrid bound respectively. The vRunTime for postal worker is set to 10000 since the thread has nothing to execute it will have lowest VRunTime and will always be the first one to schedule. Since we don't need the postal worker thread for this simulation I have initialized it with VRunTime of 10000 thus making it the rightmost node of the tree.

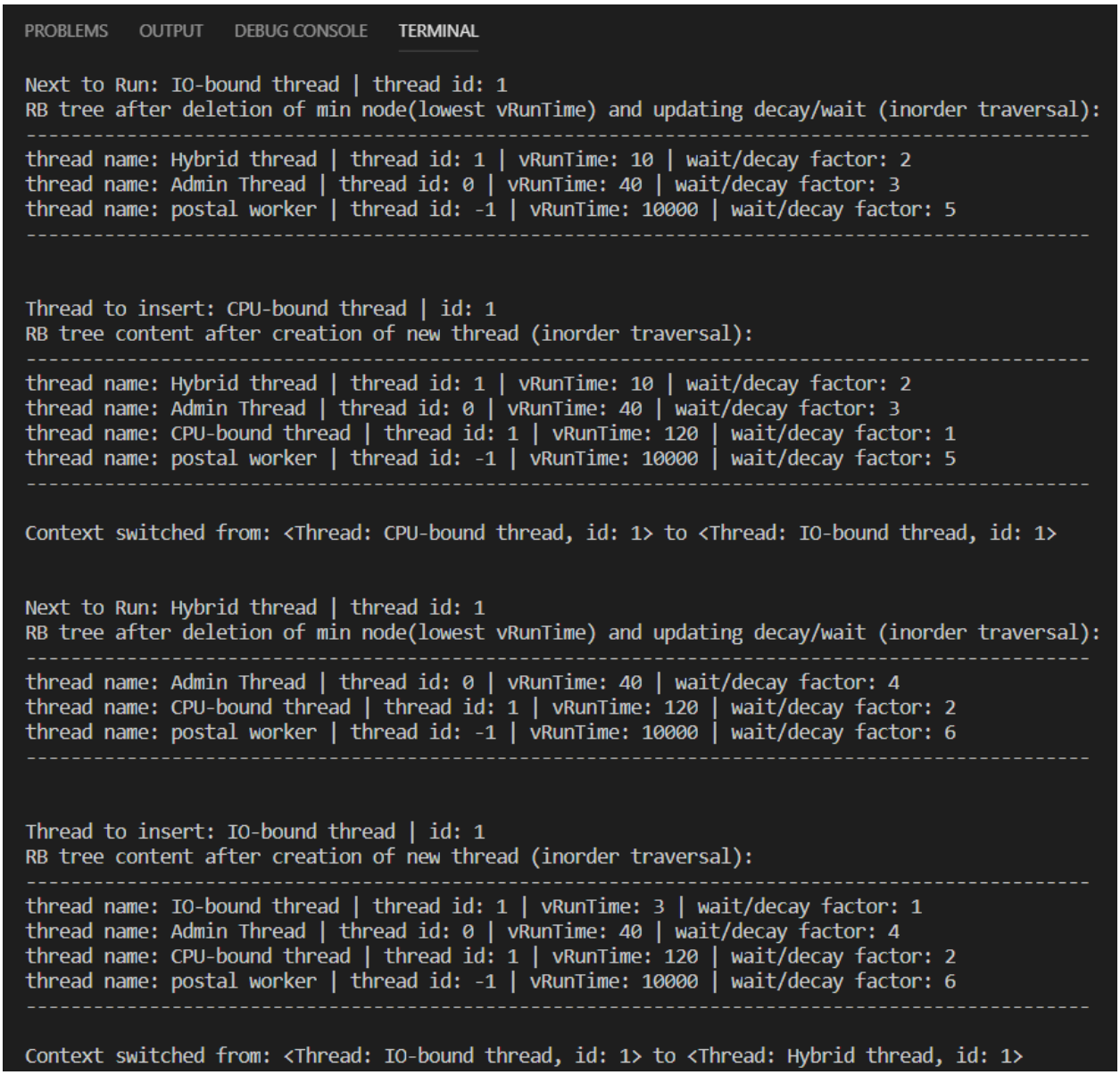
All 3 threads (Hybrid, CPU, IO) inserted into RB tree with vRunTime=0 and wait =1

Hybrid thread, leftmost node removed from RB tree and context switched with admin thread. Wait/decay for other threads is incremented by 1.

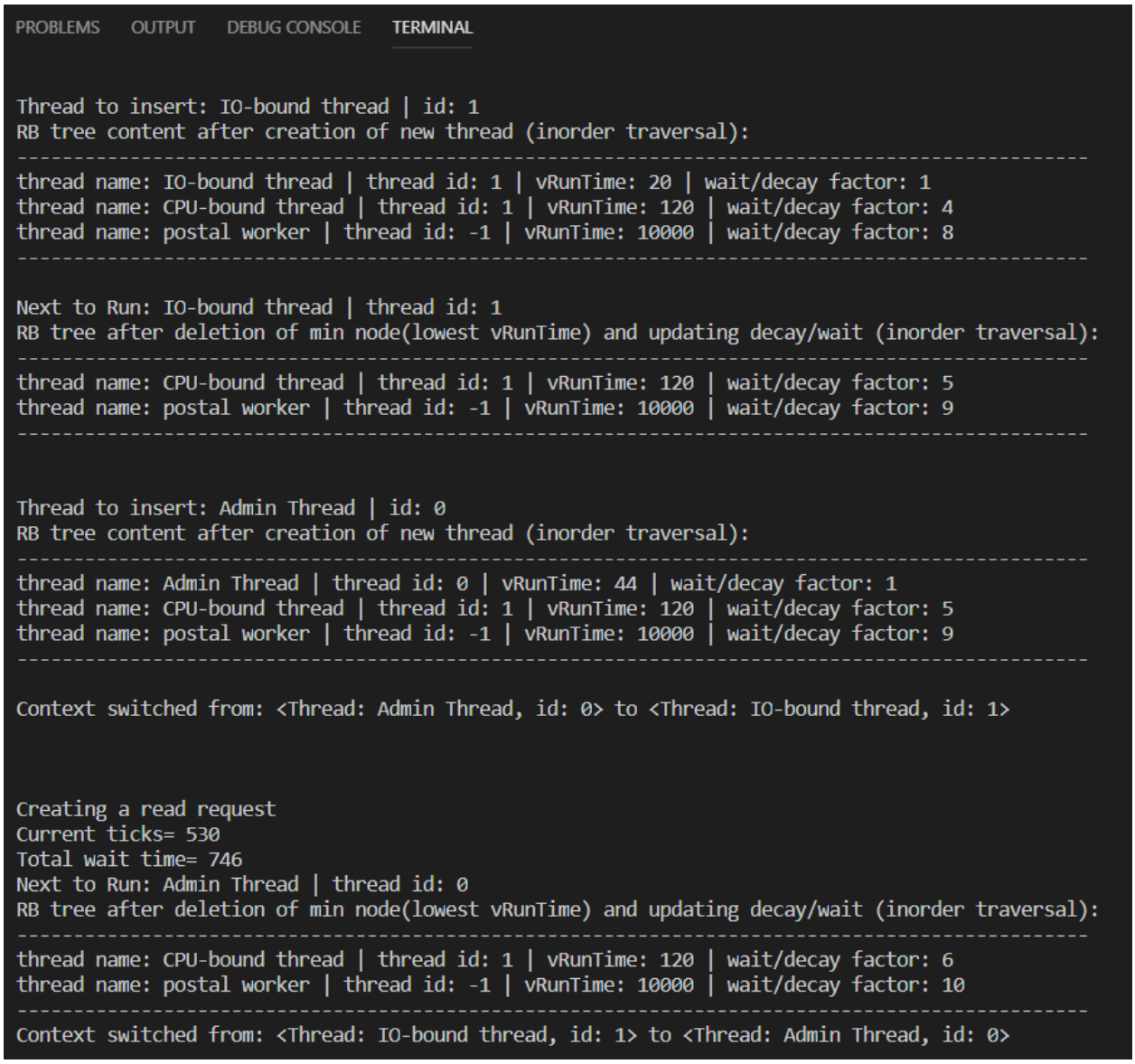
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Thread to insert: CPU-bound thread | id: 1
RB tree content after creation of new thread (inorder traversal):
-------------------------------------------------------------------------------------------
thread name: CPU-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 1
thread name: IO-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 1
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 2
-------------------------------------------------------------------------------------------



Thread to insert: Hybrid thread | id: 1
RB tree content after creation of new thread (inorder traversal):
-------------------------------------------------------------------------------------------
thread name: Hybrid thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 1
thread name: CPU-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 1
thread name: IO-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 1
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 2
-------------------------------------------------------------------------------------------

Next to Run: Hybrid thread | thread id: 1
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
-------------------------------------------------------------------------------------------
thread name: CPU-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 2
thread name: IO-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 2
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 3
-------------------------------------------------------------------------------------------



Thread to insert: Admin Thread | id: 0
RB tree content after creation of new thread (inorder traversal):
-------------------------------------------------------------------------------------------
thread name: CPU-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 2
thread name: IO-bound thread | thread id: 1 | vRunTime: 0 | wait/decay factor: 2
thread name: Admin Thread | thread id: 0 | vRunTime: 40 | wait/decay factor: 1
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 3
-------------------------------------------------------------------------------------------

Context switched from: <Thread: Admin Thread, id: 0> to <Thread: Hybrid thread, id: 1>
```

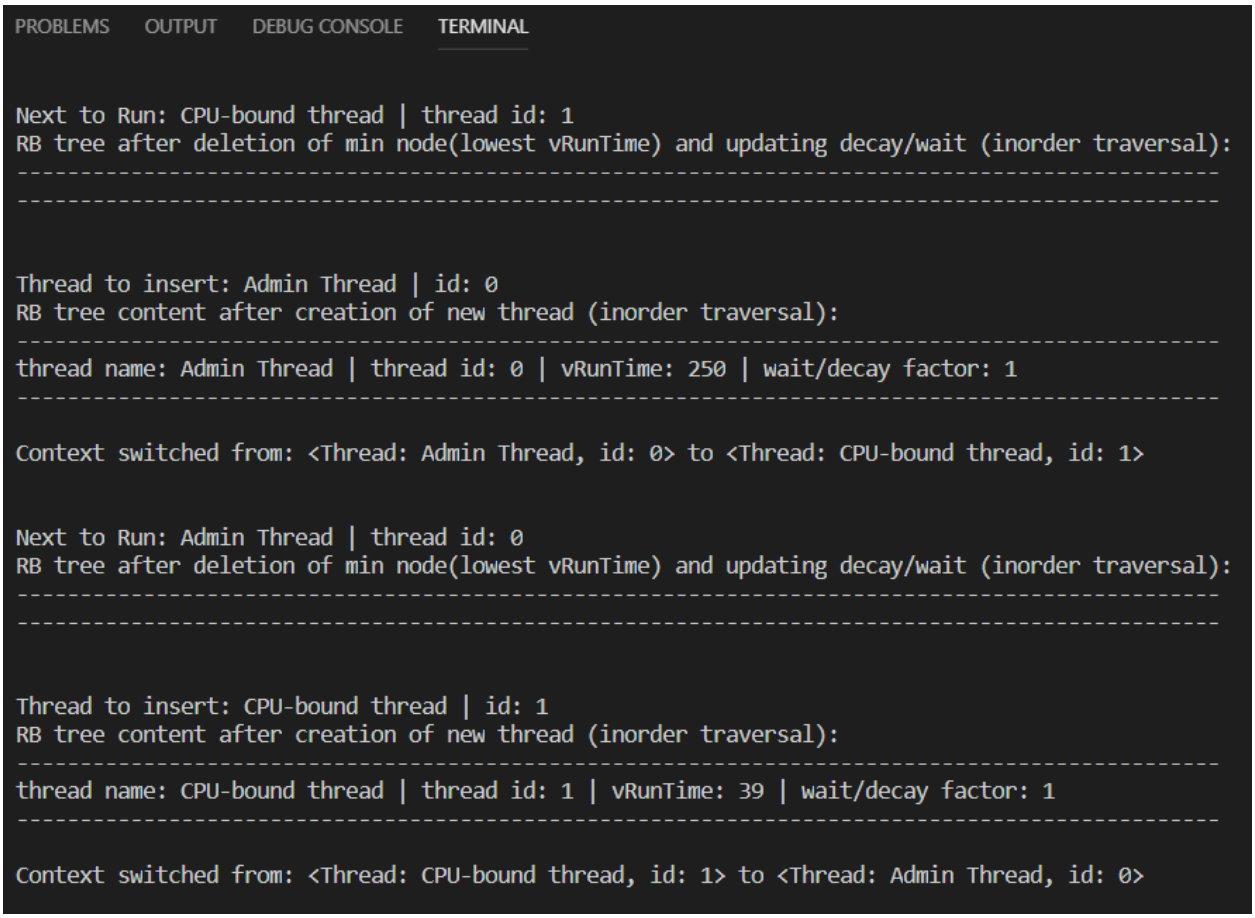
Hybrid thread, started execution and set an interrupt for time= 174 and the thread sleeps. The next thread CPU bound thread then starts with its execution after the context switch. When ticks =180 the interrupt is triggered and hybrid thread id 1 completes its write operation. Since the read operation is still remaining in the hybrid thread is inserted back to the RB tree. vRunTime of hybrid thread is also updated and wait is reset to 1.




PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Next to Run: IO-bound thread | thread id: 1
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
thread name: Hybrid thread | thread id: 1 | vRunTime: 10 | wait/decay factor: 2
thread name: Admin Thread | thread id: 0 | vRunTime: 40 | wait/decay factor: 3
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 5
Thread to insert: CPU-bound thread | id: 1
RB tree content after creation of new thread (inorder traversal):
thread name: Hybrid thread | thread id: 1 | vRunTime: 10 | wait/decay factor: 2
thread name: Admin Thread | thread id: 0 | vRunTime: 40 | wait/decay factor: 3
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 1
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 5
Context switched from: <Thread: CPU-bound thread, id: 1> to <Thread: IO-bound thread, id: 1>
Next to Run: Hybrid thread | thread id: 1
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
thread name: Admin Thread | thread id: 0 | vRunTime: 40 | wait/decay factor: 4
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 2
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 6
Thread to insert: IO-bound thread | id: 1
RB tree content after creation of new thread (inorder traversal):
thread name: IO-bound thread | thread id: 1 | vRunTime: 3 | wait/decay factor: 1
thread name: Admin Thread | thread id: 0 | vRunTime: 40 | wait/decay factor: 4
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 2
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 6
Context switched from: <Thread: IO-bound thread, id: 1> to <Thread: Hybrid thread, id: 1>

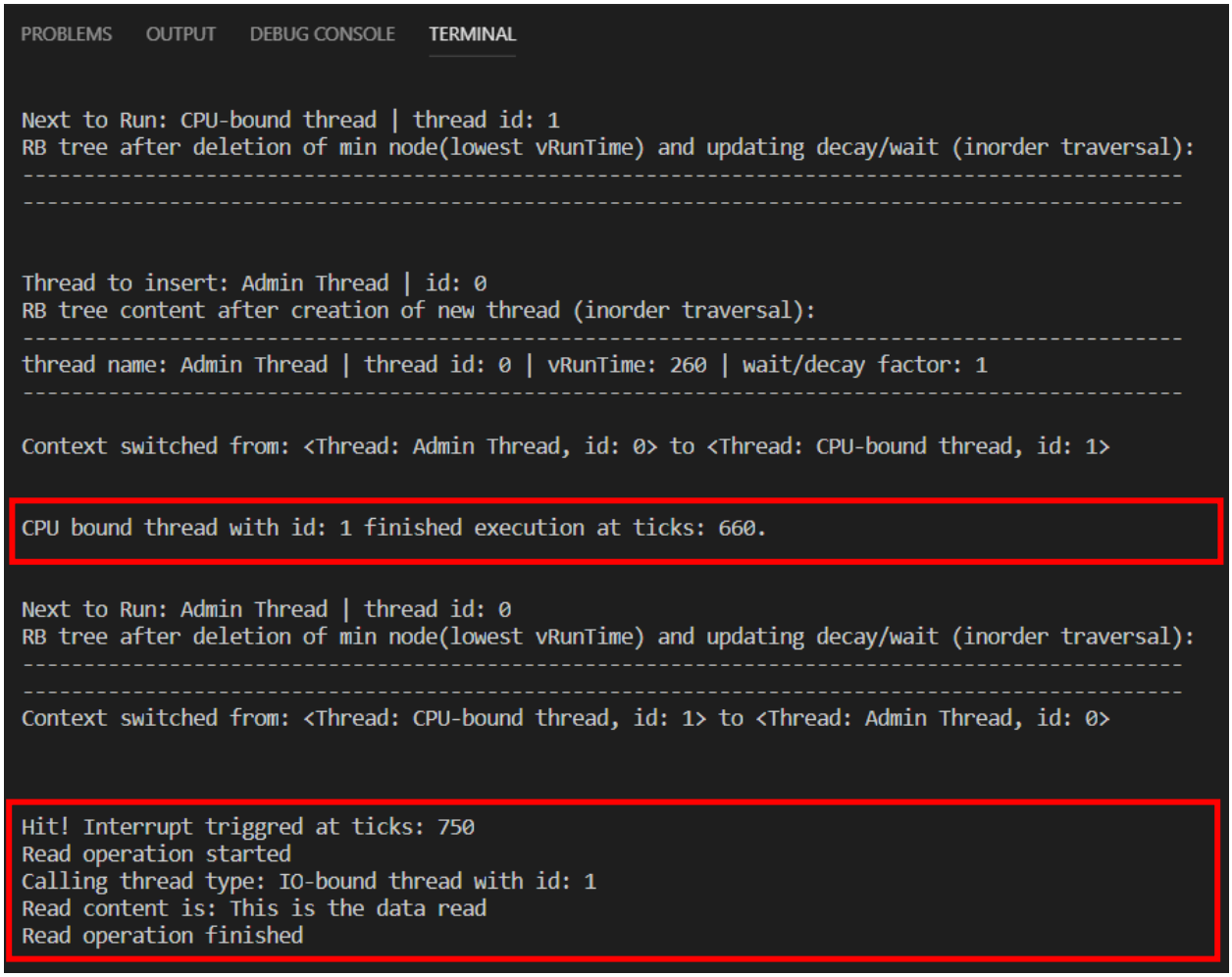
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Creating a read request
Current ticks= 330
Total wait time= 547
Next to Run: IO-bound thread | thread id: 1
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
------------------------------------------------------------------------------------------
thread name: Admin Thread | thread id: 0 | vRunTime: 40 | wait/decay factor: 5
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 3
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 7
------------------------------------------------------------------------------------------
Context switched from: <Thread: Hybrid thread, id: 1> to <Thread: IO-bound thread, id: 1>


IO bound thread with id: 1 created.
Creating a write request
Current ticks= 340
Total wait time= 448
Next to Run: Admin Thread | thread id: 0
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
------------------------------------------------------------------------------------------
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 4
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 8
------------------------------------------------------------------------------------------
Context switched from: <Thread: IO-bound thread, id: 1> to <Thread: Admin Thread, id: 0>



Hit! Interrupt triggred at ticks: 450
Write operation started
Calling thread type: IO-bound thread with id: 1
Write content is: Omkar
Write operation finished
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Thread to insert: IO-bound thread | id: 1
RB tree content after creation of new thread (inorder traversal):
thread name: IO-bound thread | thread id: 1 | vRunTime: 20 | wait/decay factor: 1
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 4
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 8
Next to Run: IO-bound thread | thread id: 1
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 5
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 9
Thread to insert: Admin Thread | id: 0
RB tree content after creation of new thread (inorder traversal):
thread name: Admin Thread | thread id: 0 | vRunTime: 44 | wait/decay factor: 1
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 5
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 9
Context switched from: <Thread: Admin Thread, id: 0> to <Thread: IO-bound thread, id: 1>
Creating a read request
Current ticks= 530
Total wait time= 746
Next to Run: Admin Thread | thread id: 0
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
thread name: CPU-bound thread | thread id: 1 | vRunTime: 120 | wait/decay factor: 6
thread name: postal worker | thread id: -1 | vRunTime: 10000 | wait/decay factor: 10
Context switched from: <Thread: IO-bound thread, id: 1> to <Thread: Admin Thread, id: 0>

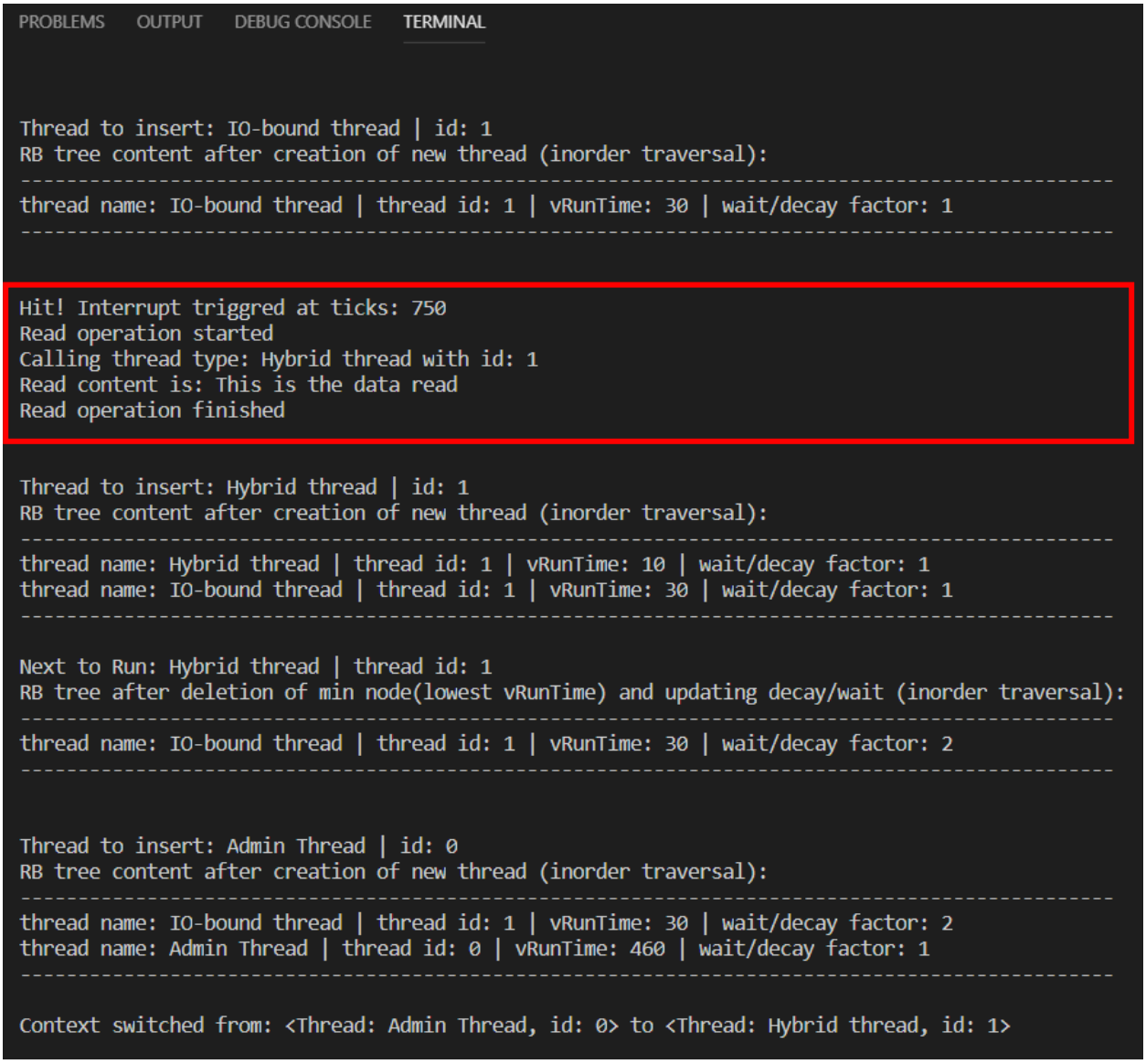

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Next to Run: CPU-bound thread | thread id: 1
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
Thread to insert: Admin Thread | id: 0
RB tree content after creation of new thread (inorder traversal):
thread name: Admin Thread | thread id: 0 | vRunTime: 250 | wait/decay factor: 1
Context switched from: <Thread: Admin Thread, id: 0> to <Thread: CPU-bound thread, id: 1>
Next to Run: Admin Thread | thread id: 0
RB tree after deletion of min node(lowest vRunTime) and updating decay/wait (inorder traversal):
Thread to insert: CPU-bound thread | id: 1
RB tree content after creation of new thread (inorder traversal):
thread name: CPU-bound thread | thread id: 1 | vRunTime: 39 | wait/decay factor: 1
Context switched from: <Thread: CPU-bound thread, id: 1> to <Thread: Admin Thread, id: 0>

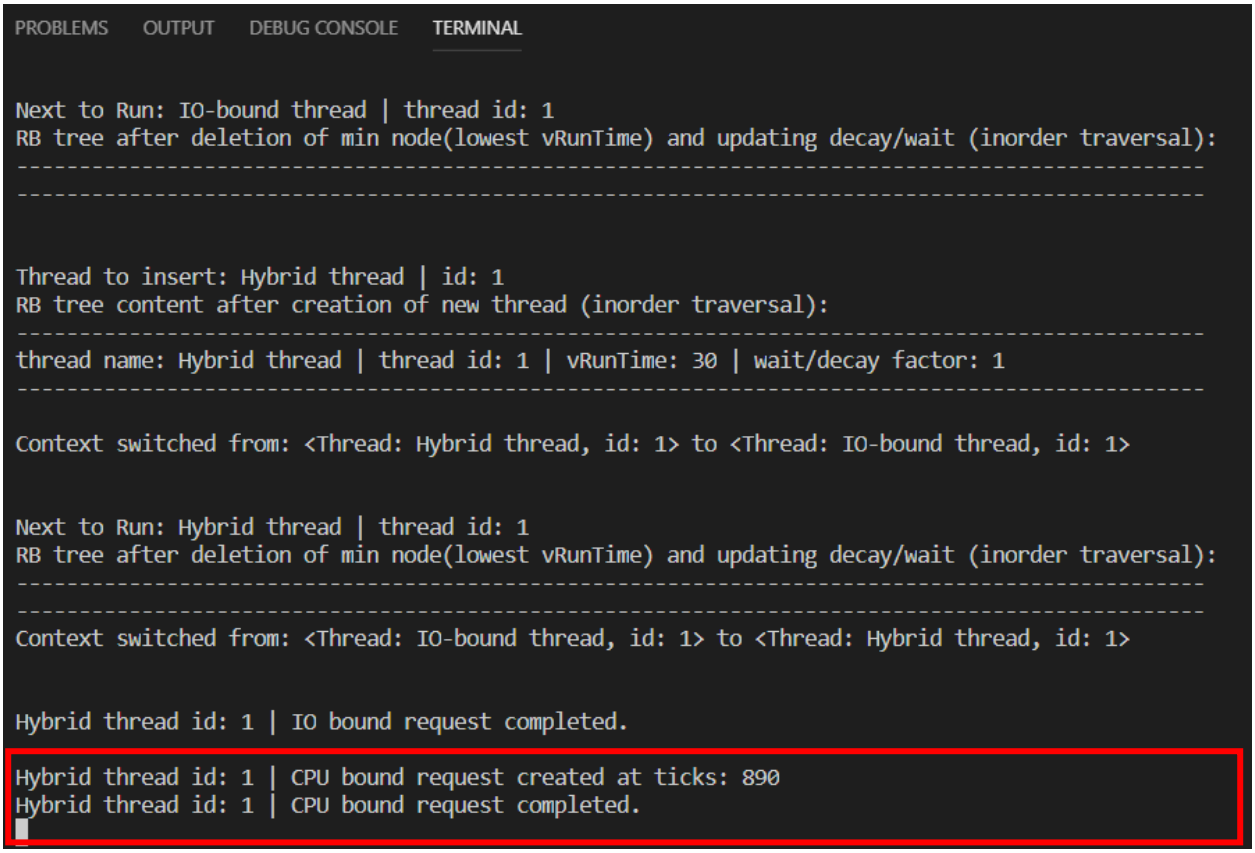
CPU bound thread with id 1 completed its execution.

Read interrupt triggered by IO bound thread with id 1 (this thread was put to sleep after setting an interrupt thus cannot be seen in the RB tree anymore). Here IO bound thread with id 1 completed its execution.



Hybrid thread id 1 completed IO operation (write+read).





Hybrid thread id 1 completed its CPU bound request.