

## Operating Systems II - CS2506

Claire Foran 115379021

### Task 1

8 Multilevel Feedback Queues:

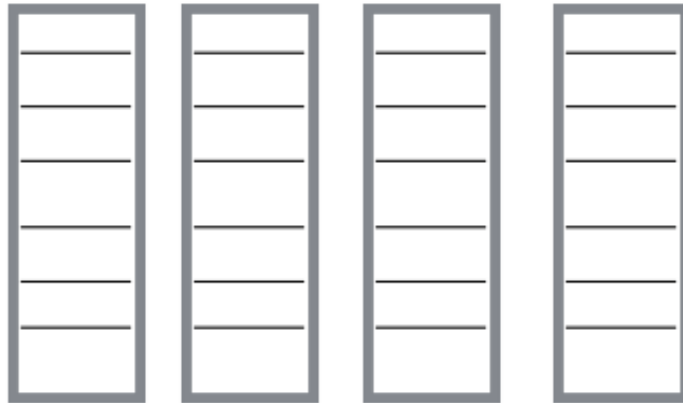
Process level:

**4**

**6**

**8**

**10**



Time Slice:

**$2^4$**

**$2^6$**

**$2^8$**

**$2^{10}$**

Process level:

**12**

**14**

**16**

**18**



Time Slice:

**$2^{12}$**

**$2^{14}$**

**$2^{16}$**

**$2^{18}$**

**Explain what happens when all user processes terminate and there is no other user process ready to execute.**

If all user processes have terminated, the CPU goes into idle mode

## Task 2:

for each queue in list of queues:

while there are processes in the queue

current process = top of queue

process time slice = process time slice - queue time slice

if process time slice  $< 0$ :

process is done

process time slice = 0

if I/O is required

add the process to the blocked queue

remove process from ready queue

wait for I/O to complete

if process priority  $> 4$

increase priority

add process back to queue of correct priority

runProcess()

else:

Remove the process from its queue

if process time slice  $> 0$ :

decrease its priority

add the process to queue of correct priority

else:

terminate the process

if all queues are empty

give idle process control of the CPU

## Task 3

```
# CPU class that holds the current process
class CPU():
    def __init__(self):
        self._process = None

# Process class that holds the process ID, the quanta, state and whether there is I/O
class Process():
    def __init__(self, pid, time, io, state, priority):
        self._pid = pid
        self._timeslice = time
        self._io = io
        self._state = state
        self._priority = priority

    def __str__(self):
        strg = "Process ID: %s, Time: %s, I/O: %s, State: %s, Priority: %i" % (self._pid, self._timeslice, self._io, self._state, self._priority)
        return strg

class Queue():
    def __init__(self, timeslice, priority_level):
        self._queue = []
        self._timeslice = timeslice
        self._head = 0
        self._priority = priority_level

    def __str__(self):
        output = "["
        for process in self._queue:
            output += "%s," % process._pid
        return output + "]"

    def getProcess(self):
        # Return process at the top of blocked queue
        if self._queue != []:
            return self._queue[0]

    def add(self, process):
        # Append process to the end of the queue
        self._queue += [process]

    def remove(self):
        # Remove the first process in the queue
        return self._queue.pop(0)

# Scheduler class that holds the queues, the process and the CPU
class Scheduler():
    def __init__(self):
        self._queuelist = [Queue(2**n+4, n+4) for n in range(0, 16, 2)]
        self._blockedQ = Queue(None, None)
        self._process = None
        self._cpu = CPU()
        self.run()

    def addProcess(self, process):
        # add process to ready queue
        for queue in self._queuelist:
            if process._priority == queue._priority:
                queue.add(process)
                return queue

        return False

    def removeProcess(self, queue):
        # remove the current process from ready queue
        return queue.remove()

    def addBlockedProcess(self, process):
        # add process to blocked queue
        self._blockedQ.add(process)

    def removeBlockedProcess(self):
        # remove process from blocked queue
        self._readyQ.remove()

    def getProcess(self):
        # return the current process
        return self._process
```

```

def checkQueues(self):
    # Check if all queues in the queue_list are empty.
    # If not, return False. Otherwise, return True
    for queue in self._queuelist:
        if len(queue._queue) > 0:
            return False
    return True

def runProcess(self):
    # Runs current process and checks for I/O
    # Moves process to Blocked queue and then to Ready queue if I/O
    # Reduces the process's quanta

    for queue in self._queuelist:
        print("***Queue Level:", queue._priority, "Timeslice:", queue._timeslice, "***")
        print(queue)
        while len(queue._queue) > 0:
            self._process = queue.getProcess()
            self._cpu._process = self._process
            print(self._process)

            self._process._timeslice -= queue._timeslice
            if self._process._timeslice < 0:
                self._process._timeslice = 0
            # if there is I/O required
            if self._process._io == True:
                self._cpu._process = None
                self.addBlockedProcess(self._process)
                process = self.removeProcess(queue)
                print("Waiting for I/O")
                self._process._state = "Blocked"
                if self._process._priority < 18 and self._process._priority > 4:
                    self._process._priority -= 2
                self._process._io = False
                self.addProcess(process)
                print("Process returned to queue")
                print("Re-run")
                self._process._state = "Ready"
                self.runProcess()
            else:
                process = self.removeProcess(queue)
                print("Process running")
                # if process is still unfinished
                if int(self._process._timeslice) > 0:
                    self._process._priority += 2

                self._process._priority -= 2
                result = self.addProcess(process)
                self._cpu._process = None
                print("Process returned to queue \n")
            else:
                print("Process terminated \n")

    if self.checkQueues() == False:
        self.runProcess()
    else:
        process = Process(0, None, "IDLE", None)
        self._addProcess(process)
        self._CPU._process = process

```

## Task 4

```
def run(self):
    self.addProcess(Process(1, 40, False, "ready", 4))
    self.addProcess(Process(2, 79, True, "ready", 8))
    self.addProcess(Process(3, 100, False, "ready", 14))
    self.addProcess(Process(4, 55, True, "ready", 10))
    self.addProcess(Process(5, 81, False, "ready", 18))
    self.addProcess(Process(6, 700, False, "ready", 10))
    self.runProcess()
```

Execution:

```
**Queue Level: 4 Timeslice: 5 **
[1,]
Process ID: 1, Time: 40, I/O: False, State: ready, Priority: 4
Process running
Process returned to queue

**Queue Level: 6 Timeslice: 8 **
[1,]
Process ID: 1, Time: 35, I/O: False, State: ready, Priority: 6
Process running
Process returned to queue

**Queue Level: 8 Timeslice: 20 **
[2,1,]
Process ID: 2, Time: 79, I/O: True, State: ready, Priority: 8
Waiting for I/O
Process returned to queue
Re-run
**Queue Level: 4 Timeslice: 5 **
[]
**Queue Level: 6 Timeslice: 8 **
[2,]
Process ID: 2, Time: 59, I/O: False, State: Ready, Priority: 6
Process running
Process returned to queue

**Queue Level: 8 Timeslice: 20 **
[1,2,]
Process ID: 1, Time: 27, I/O: False, State: ready, Priority: 8
Process running
Process returned to queue

Process ID: 2, Time: 51, I/O: False, State: Ready, Priority: 8
Process running
Process returned to queue
```

```
**Queue Level: 10 Timeslice: 68 **
[4,6,1,2,]
Process ID: 4, Time: 55, I/O: True, State: ready, Priority: 10
Waiting for I/O
Process returned to queue
Re-run
**Queue Level: 4 Timeslice: 5 **
□
**Queue Level: 6 Timeslice: 8 **
□
**Queue Level: 8 Timeslice: 20 **
[4,]
Process ID: 4, Time: 0, I/O: False, State: Ready, Priority: 8
Process running
Process terminated

**Queue Level: 10 Timeslice: 68 **
[6,1,2,]
Process ID: 6, Time: 700, I/O: False, State: ready, Priority: 10
Process running
Process returned to queue

Process ID: 1, Time: 7, I/O: False, State: ready, Priority: 10
Process running
Process terminated

Process ID: 2, Time: 31, I/O: False, State: Ready, Priority: 10
Process running
Process terminated

**Queue Level: 12 Timeslice: 260 **
[6,]
Process ID: 6, Time: 632, I/O: False, State: ready, Priority: 12
Process running
Process returned to queue

**Queue Level: 14 Timeslice: 1028 **
[3,6,]
Process ID: 3, Time: 100, I/O: False, State: ready, Priority: 14
Process running
Process terminated

Process ID: 6, Time: 372, I/O: False, State: ready, Priority: 14
Process running
Process terminated

**Queue Level: 16 Timeslice: 4100 **
□
**Queue Level: 18 Timeslice: 16388 **
[5,]
Process ID: 5, Time: 81, I/O: False, State: ready, Priority: 18
Process running
Process terminated
```