**-n 38.2**

Example 1: ./mlfq.py -n 3 -l 0,200,0 -c

Example 2: ./mlfq.py -n 3 -l 0,200,0:100,20,0 -c

Example 3: ./mlfq.py -n 3 -l 0,20,1:0,200,0 -c

**8.3**

./mlfq.py -n 1 0,200,0:0,200,0

Dadurch, dass es nur eine Queue gibt, wechselt der Scheduler stets zwischen den beiden identischen Jobs hin und her und verhält sich wie ein Round Robin scheduler.

**8.4**

./mlfq.py -n 3 -q 100 -I 0,500,0:0,500,99 -S -c

**8.5**

**200ms.**

100% / 5% = 20 20 \* 10ms = 200ms

**8.6**

In unserem Beispiel:

./mlfq.py -n 3 -l 0,200,0:0,20,1:0,30,1 -I -c

./mlfq.py -n 3 -l 0,200,0:0,20,1:0,30,1 -c

Ging die Turnaroundtime von Job 3 nach unten mit der -I Option. Sie ging um 54s runter. Durch die -I option wird ein I/O Task vorne in der Queue eingereiht und nicht hinten, dies führt dazu dass der task 2 früher angefangen wird, daher auch schneller fertig wird.