An example of a design your mileage may vary, just to follow curCpu and stats

Suggest you run main loop with only 1 process loaded till it terminates. For test of start and end of program.

Current time in CPU

Two reasons to leave CPU

curCpu == cpu so time to move to I/O add curCpu to cpuTotal and reset to 0

curCpu >= time quantum so move to ready/wait queue the curCpu must not change

curCpu only set to 0 on load process and when moving to IO

Current time in ready queue

If need to increase priority do not change time in ready, that total needed for min/max later

Increase priority when mod by max time == 0

Load and terminate

Have var activeCount for # of process 0 for none main loop test > 0

Load first process before main loop so activeCount is 1

Adding process for first 48 time clicks loading process

Have variable for nextLoad pointing to next index in array of processes 0 being first

Increment activeCount

Set stats to 0 and time in ready and curCpu to 0

Enqueue

cpuToIo

add curCpu to total then reset to 0

set current time in i/o to zero

checkCpu

increment curCpu

time for i/o cpuToIo

time quantum ToReady leave curCpu alone

if moved set CPU to empty ? 48

toReady

set time waiting to 0

set curPrior to original priority

enqueue (increment readyCount here)

checkReady

if cpu empty readyToCpu

each process

increment time in ready

test for need to upgrade priority

sort by curPrior

readyToCpu

dequeue and get process id

do not change curCpu

update stats for time in ready/wait

if first time sum==0 so set sum, min and max to time in wait else add/compare

ioToReady

remove from io get id

reset curio to 0

toReady (id)

remove from io returns process id

get process id to temp save index

move array members up to fill hole

decrement ioCount after move needed for loop

dequeue

returns top of queue process id

move array elements up to fill hole

decrement readyCount

enqueue

put on bottom of array increment count