Problem 1. (12 Points) Complete the following fair share scheduling with TQ=60 ticks and Threshold=60. Assume that 5 processes in the system: {P1, P3} are belonging to group 1, while {P2, P4, P5} are belonging to group 2. Assume that both decay functions for “CPU Usage” and “Group CPU Usage” are using N=2. Starts at t=0 and stop at t=3+.

<table>
<thead>
<tr>
<th>GCPU1</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>GCPU2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU PRI</td>
<td>CPU PRI</td>
<td>CPU PRI</td>
<td>CPU PRI</td>
<td>CPU PRI</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>24</td>
<td>14</td>
<td>18</td>
<td>18</td>
<td>32</td>
<td>17</td>
</tr>
</tbody>
</table>

Problem 2. (30 Points) Assume that P_A, P_B and P_C are three distinctive programs. Assume that when P_A is executed, it forks new processes executing P_C at tick-marks 2, 4 and 7, and it has a burst time of 8. Assume that when P_B is executed, it forks new processes executing P_C at tick-marks 2 and 4, and it has a burst time of 6. Assume that P_C has a burst time of 4 and it requests an I/O at tick-mark 2. Assume that 3 processes {P_1, P_2, P_3} had arrived before t=0; and P_1 is executing P_A while P_2 and P_3 are executing P_B. Assume that each I/O request can be completed within 4 ticks and I/O requests are handled in a FCFS manner. Draw a Gantt chart illustrating the scheduling if a Round-Robin scheduling with a TQ=3 ticks is used. Mark on the chart when a process is arrived and finished. Note that you may need both a Ready Queue and an I/O Queue.

Problem 3. (18 Points) Assume that there are 4 processes {P_1, P_2, P_3, P_4} in the system before t=0. Assume that their burst times are given as {25, 37, 18, 29}. Assume that a monitoring signal is generated every 3 tick periodically starting at t=0. Assume that the signal is intercepted by the current running process and it should count how many signals it has received and reports the count before it complete its execution. Assume that a Round-Robin scheduling with a time quantum of 13 is used. Find out how many signals each process has encountered.
Problem 1. (30 Points) Assume that \( P_A, P_B \) and \( P_C \) are three distinguished programs. Assume processes that executing \( P_A \) has a priority of 2, processes that executing \( P_B \) has a priority of 3, and processes that executing \( P_C \) has a priority of 1. Assume in order to complete their execution, \( P_A \) needs 10 ticks, \( P_B \) needs 7 ticks, and \( P_C \) needs 5 ticks. Assume that both \( P_A \) and \( P_B \) will create new process every 3 ticks of execution, and \( P_A \) forks new processes executing \( P_B \) while \( P_B \) forks new processes executing \( P_C \). Assume that two processes \( P_1 \) and \( P_2 \) have arrived before \( t=0 \) and \( P_1 \) is executing program \( P_A \) and \( P_2 \) is executing \( P_B \). Draw a Gantt chart illustrating the scheduling of these processes if a priority with round-robin scheduling is used. Assume that a time quantum of 4 is used.

Problem 2. (12 Points) Assume that there are 4 processes \( \{ P_1, P_2, P_3, P_4 \} \) in the system before \( t=0 \). Assume that process \( P_i \) has a burst time of \( 5*i \). Assume that a hardware failure will generates periodical signals and sent to the current running process every 3 ticks. Assume that a round-robin scheduling system with a time quantum of 4 is used. Find out how many signals each process has received.

Problem 3. (18 Points) Assume that the following two command lines were executed under an UNIX shell:

\[
\begin{align*}
\% & \ p | q | r < \text{data1} > \text{out1} & \\
\% & \ u | v < \text{data2} > \text{out2} \\
\end{align*}
\]

where | denotes pipeline, < denotes input (or stdin), and > denotes output (or stdout). So, there are total 5 individual processes will be created after these two commands were executed. You need to assign proper PIDs to them according to the sequence of execution. Assume that the internal procedure for all process can be represented as the following pseudo codes:

\[
\begin{align*}
\text{while true} & \\
& \text{read data} \\
& \text{process data} \\
& \text{write result} \\
\text{end}
\end{align*}
\]

Assume that it takes 2 ticks to perform each loop and assume that processes that executing \( p, q \) or \( u \) need 4 loops to complete and processes that executing \( r \) or \( v \) need 3 loops to complete. Draw a Gantt chart illustrating the scheduling of these processes if a round-robin scheduling with a time quantum of 3 is used.
Problem 1. (35 Points) Assume that P_A, P_B and P_C are three distinctive programs. Assume that when P_A is executed, it forks new processes executing P_B at tick-marks 2 and 5, it does an I/O request at tick-mark 4, and its burst time is 8. Assume that when P_B is executed, it forks new processes executing P_C at tick-marks 2 and does an I/O request at tick-mark 3, and its burst time is 6. Assume that P_C has a burst time of 4 and it does an I/O request at tick-mark 2. Assume that two processes P_1 and P_2 have arrived before t=0; and P_1 is executing P_A while P_2 is executing P_B. Assume that each I/O request can be completed within 3 ticks and I/O requests are handled in a FCFS manner. Draw a Gantt chart illustrating the scheduling if a FCFS scheduling is used. Mark on the chart when a process is arrived and finished.

Problem 2. (30 Points) Redo the scheduling for Problem 1 but this time we are using the HRRN (Highest Response Rate Next) scheduling. Note that

Response Ratio for P_i = (WT_i + BT_i)/BT_i.

Problem 3. (20 Points) Redo the scheduling for Problem 1 but this time we are using the MFQ (Multilevel Feedback Queue) scheduling. Assume that both the first and the second queues use the same time quantum of 3, and the last queue uses FCFS scheduling and assume that no I/O requests for all processes.

Problem 4. (15 Points) Assume that the execution of each process is divided into a sequence of CPU bursts. Assume that P1, P2 and P3 are three processes in the system and they all arrived before t=0. Their sequences of CPU bursts are specified as the following:

P1: {1, 3, 2}, P2: {2, 1, 2}, and P3: {1, 1, 4}.

Use the following formula to evaluate the (n+1)-th CPU burst time:

\[ \tau_{n+1} = \alpha \tau_n + (1 - \alpha) \tau \]

where \( \alpha = 0.5, \tau_0 = 0 \) and \( \tau = 5 \) for all processes. Draw a Gantt chart to illustrate the scheduling of these processes if a shortest-job-first schedule algorithm is used and only up to n=6.
Problem 1. (30 Points) Assume that P_A, P_B and P_C are three distinctive programs. Assume that when P_A is executed, it forks new processes executing P_B at tick-marks 2 and 5, and its burst time is 8. Assume that when P_B is executed, it forks new processes executing P_C at tick-marks 2 and 5, and its burst time is 6. Assume that P_C has a burst time of 3 and needs to do I/O after 2 ticks of execution. Assume that two processes P_1 and P_2 have arrived before t=0; and P_1 is executing P_A while P_2 is executing P_B. Draw a Gantt chart illustrating the scheduling if a round robin scheduling with a time quantum of 3 is used. Assume that each I/O needs 4 ticks to complete and I/O requests are handled in a FCFS strategy. Mark on the chart when a process is arrived and finished.

Define Multilevel Feedback Queue Scheduling (MFQS) which is shown as the diagram below. Assume that queue 1 and 2 follow round robin with the same time quantum 3 and queue 3 follows FCFS. The scheduling of MFQS is given as below:

1. New processes first enter queue 1.
2. In a general rule, if a process does not complete in a time quantum than it is shifted to the lower priority queue.
3. In the last queue, processes are scheduled in a FCFS manner.
4. A process in lower priority queue can only execute only when higher priority queues are empty.

Problem 2. (30 Points) Redo the scheduling for Problem 1 but this time we are using the MFQS scheduling and no I/O requests for processes which are executing P_C.
Problem 1. (35 Points) Assume that $P_A$ and $P_B$ are two distinctive programs. Assume that when $P_A$ is executed, it forks new processes executing $P_B$ at tick-marks 2, 4 and 7; and the execution of $P_A$ will be divided into a sequence of CPU bursts of $\{3, 2, 3\}$ by interrupts. Assume that $P_B$ has a burst time of 5 and it does I/O at tick-marks 2 and 4. Assume that each I/O can be completed within 3 ticks and all I/O requests are handled in a FCFS manner. Assume that two processes $P_1$ and $P_2$ have arrived before $t=0$; and both are executing $P_A$. Draw a Gantt chart illustrating the scheduling if a FCFS scheduling is used. Mark on the chart when a new process is arrived.

Problem 2. (25 Points) Assume that there are three processes had arrived before $t=0$ in the order of $\{P_1, P_2, P_3\}$. Assume that each process was divided into a sequence of CPU bursts by random interrupts and the sequence of CPU bursts for given processes are specified as:

$P_1$: $\{3, 2, 2\}$, $P_2$: $\{1, 4\}$, and $P_3$: $\{1, 1, 5\}$.

Assume that the following formula is used to evaluate $\tau_{n+1}$, the $(n+1)$th estimated burst time:

$$\tau_{n+1} = \alpha \ \tau_n + (1 - \alpha) \ \tau_{n-1}$$

where $\alpha=0.5$, $\tau_0=0$ and $\tau_{-1}=5$ for all processes. The process with the shortest estimated burst time will be scheduled to use the CPU. Draw a Gantt chart illustrating this scheduling.

Problem 3. (5 Points) In the previous question, assume that there are no interrupts caught by any of the three processes of $\{P_1, P_2, P_3\}$. Then what’s happening to the scheduling?