وزارة التعليم العالي والبحث العلمي

جامعة نينوى

كلية تكنلوجيا المعلومات

**Real-Time System**

**LLF Scheduling**

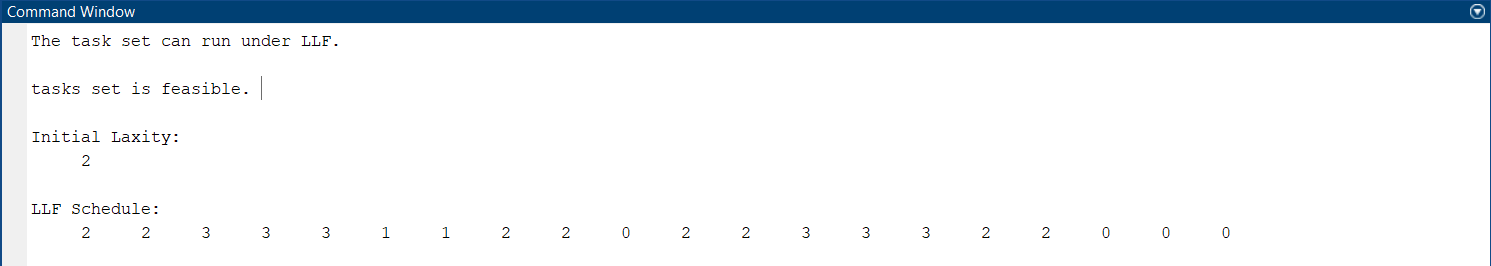
**Task**

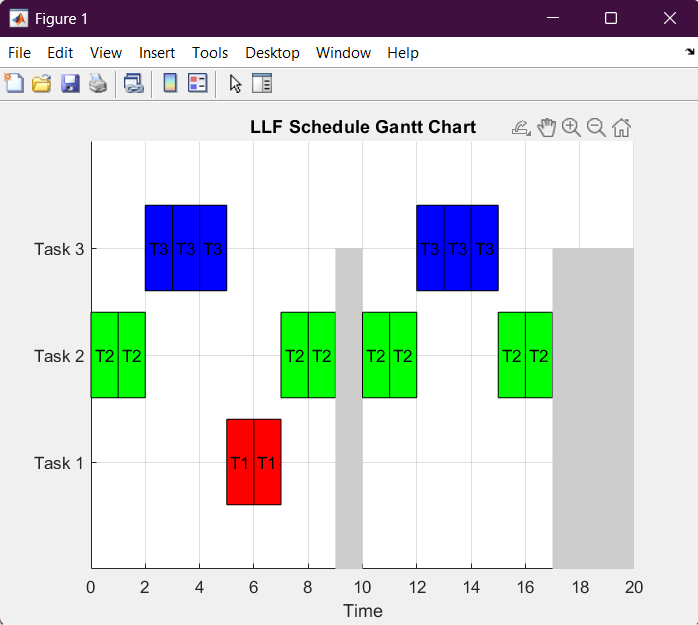
**Raheed Fadi & Hanan Salih**

**LLF Scheduling algorithm code:**

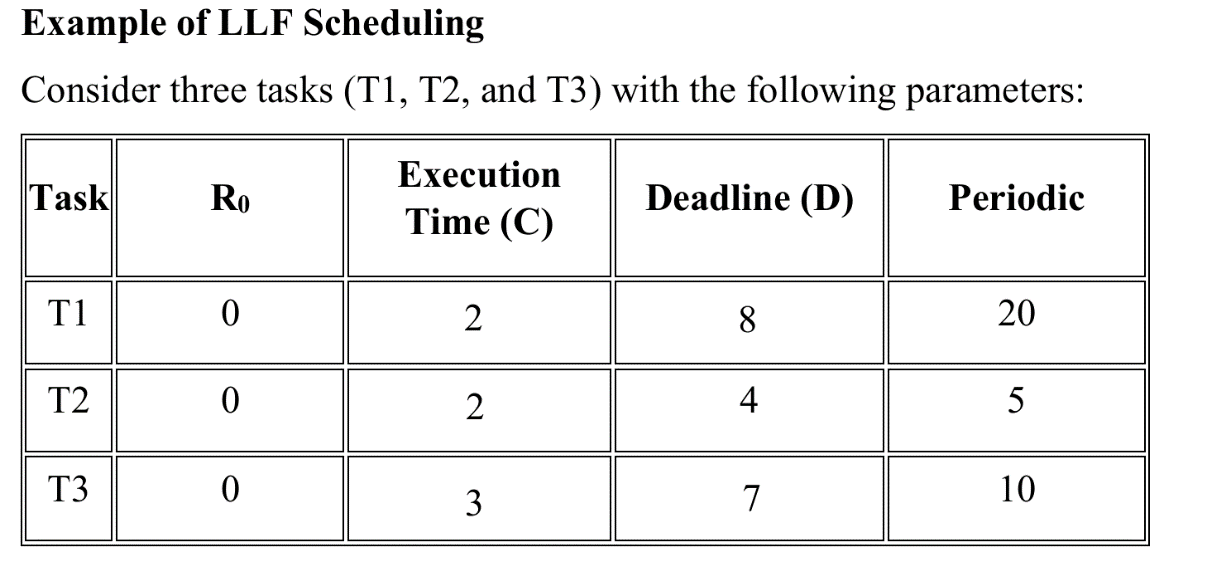
|  |
| --- |
| clc;  close all;  % Release-Time, Execution-Time, Deadline, Period  tasks = [**0**, **2**, **8**, **20**; % T1  **0**, **2**, **4**, **5**; % T2  **0**, **3**, **7**, **10**]; % T3  numTasks = size(tasks, **1**);  hyperperiod = lcm(lcm(tasks(**1**,**4**), tasks(**2**,**4**)), tasks(**3**,**4**));  schedule = zeros(**1**, hyperperiod);  remainingExecution = tasks(:, **2**);  nextRelease = tasks(:, **1**);  nextDeadline = tasks(:, **3**);  **for** t = **0**:hyperperiod-**1**  **for** i = **1**:numTasks  **if** t >= nextRelease(i) && remainingExecution(i) == **0**  nextRelease(i) = nextRelease(i) + tasks(i, **4**);  nextDeadline(i) = nextDeadline(i) + tasks(i, **4**);  remainingExecution(i) = tasks(i, **2**);  **end**  **end**  laxities = inf(numTasks, **1**);  **for** i = **1**:numTasks  **if** t >= nextRelease(i) && remainingExecution(i) > **0**  laxities(i) = nextDeadline(i) - (t + **1**);  **end**  **end**  [minLaxity, taskIndex] = min(laxities);  **if** minLaxity < inf  schedule(t + **1**) = taskIndex;  remainingExecution(taskIndex) = remainingExecution(taskIndex) - **1**;  **if** remainingExecution(taskIndex) == **0**  nextRelease(taskIndex) = nextRelease(taskIndex) + tasks(taskIndex, **4**);  nextDeadline(taskIndex) = nextDeadline(taskIndex) + tasks(taskIndex, **4**);  remainingExecution(taskIndex) = tasks(taskIndex, **2**);  **end**  **else**  schedule(t + **1**) = **0**;  **end**  **end**  figure;  hold on;  colors = ['r', 'g', 'b', 'c', 'm', 'y'];  **for** t = **1**:hyperperiod  task = schedule(t);  **if** task > **0**  rectangle('Position', [t-**1**, task-**0.4**, **1**, **0.8**], ...  'FaceColor', colors(mod(task-**1**, length(colors))+**1**), ...  'EdgeColor', 'k');  text(t-**0.5**, task, sprintf('T%d', task), ...  'HorizontalAlignment', 'center', 'VerticalAlignment', 'middle');  **else**  rectangle('Position', [t-**1**, **0**, **1**, numTasks], ...  'FaceColor', [**0.8** **0.8** **0.8**], 'EdgeColor', 'none');  **end**  **end**  ylim([**0** numTasks+**1**]);  xlim([**0** hyperperiod]);  yticks(**1**:numTasks);  yticklabels(arrayfun(@(x) sprintf('Task %d', x), **1**:numTasks, 'UniformOutput', false));  xlabel('Time');  title('LLF Schedule Gantt Chart');  grid on;  utilization = sum(tasks(:, **2**) ./ tasks(:, **4**));  **if** utilization <= **1**  fprintf('The task set can run under LLF. \n\n');  fprintf('tasks set is feasible. \n\n');  **else**  fprintf("The task set can't run under LLF.\n\n");  fprintf("tasks set isn't feasible. \n\n");  **end**  disp('Initial Laxity:'), disp(schedule(**1**));  disp('LLF Schedule:'), disp(schedule); |

**The output of the code:**

****

****

**Mathematical solution:**

****

