Operating Systems ClassWork 4 – Resource Allocation Page 1 of 3

Week \_\_\_\_\_\_\_\_\_\_\_ Student Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_ \_

|  |
| --- |
| **Resources** |
| 1. Identify 2 resources of a computer system that are shareable (can be used by more than one process at the same time)  **File that is only being read**  **Memory** |
| 2. Identify 2 resources of a computer system that are not shareable ( can be used by only one process at a time)  **File that is being written**  **CPU** |
| **Policies v Mechanisms** |
|  |
| 3. What is the **mechanism** for delivery of tea and coffee in the college restaurant?  **get a cup, put cup in place, push the button take the cup, pay the money** |
| 4. What is the **policy** for delivery of tea and coffee in the college restaurant?  **You must pay for it**  **It is only available during opening hours**  **Anybody can buy it** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Deadlock avoidance strategy**  A system has resources R1, R2 , R3, R4  Process A holds R1 and may eventually need R2 and R3.  Process B holds R2 and may eventually need R4.  Process C holds R3 and may eventually need R4  The current situation is represented in the following table:   |  |  |  | | --- | --- | --- | | Process | Holds currently | Possible Future Claims | | A | R1 | R2,R3 | | B | R2 | R4 | | C | R3 | R4 | |  |  |  | |
| 5. Is the system currently in a safe state? (Is there a sequence in which the processes could finish even if all of them need all their possible future resources?).  If yes show a sequence of process terminations that would be guaranteed possible.  If no show why not.  HINT: use the resource allocation graph  **Yes it is safe.**  **R4 is free and may be acquired by C at any time. When C completes R3 and R4 become free. Process B can now acquire R4 and finish freeing up R2. When R2 becomes free A can get R2 and R3 and so finish.**  **Possible finishing sequence is C followed by B followed by A**. |
| **6.** For the problem from Question 5, a new process D arrives which needs R4 to get started and may eventually need R1.  Should process D be allowed to start (and therefore to acquire R4). Justify your answer.    **Giving R4 to process D would mean that no resources are left available and no process could get the rest of what it requires and finish.**  **D would wait for R1 to become free**  **A has R1 would wait for R2 and R3 to come free**  **B has R2 and would wait for R4 to come free**  **C has R3 and would wait for R4 to come free**  **D would have R4 and would wait for R1 to come free……it’s a circular wait => deadlock.**  **So, it would not be safe to let process D get started by giving it R4 as deadlock might result.** |