## 操作系统

## 第六次作业

**死锁**

**7.1  Consider the traffic deadlock depicted in Figure 7.10.**

**a. Show that the four necessary conditions for deadlock indeed hold in this example.**

**b. State a simple rule for avoiding deadlocks in this system.**

1. **互斥：**道路资源在某一时刻只能有一辆汽车通过。

**占有并等待：**正在通过某段道路的车辆占用了这段道路，并需要等待其前方的车辆离开前方的道路。

**不可抢占：**道路只有在车辆通过后会被释放，车辆不能抢占其他车辆的位置。

**循环等待：**每辆车都在等待其前方的道路资源，四条道路连接成环，车辆的等待也形成闭环。

此案例满足死锁发生的四个必要条件。

1. 规定车辆不能在十字路口处停留，若路口前方的道路无法通过，则停留在路口后方等待。在这种规则下，车辆可以有序通过路口，不会出现路口被占用导致的循环等待。

**7.11  Consider the following snapshot of a system:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Allocation***  **A B C D** | ***Max***  **A B C D** | ***Available***  **A B C D** |
| **P0** | **0012** | **0012** | **1520** |
| **P1** | **1000** | **1750** |  |
| **P2** | **1354** | **2356** |  |
| **P3** | **0632** | **0652** |  |
| **P4** | **0014** | **0656** |  |

**Answer the following questions using the banker’s algorithm:**

**a. What is the content of the matrix Need?**

**b. Is the system in a safe state?**

**c. If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?**

1. Need[i,j] = Max[i,j] – Allocation[i,j]

|  |  |
| --- | --- |
|  | ***Need***  **A B C D** |
| **P0** | **0000** |
| **P1** | **0750** |
| **P2** | **1002** |
| **P3** | **0020** |
| **P4** | **0642** |

1. 根据安全性算法，

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Allocation***  **A B C D** | ***Need***  **A B C D** | ***Work***  **A B C D** | ***Finish*** |
|  |  |  | **1 5 2 0** |  |
| **P0** | **0012** | **0000** | **1 5 3 2** | **True** |
| **P1** | **1000** | **0750** | **3 14 12 12** | **True** |
| **P2** | **1354** | **1002** | **2 8 8 6** | **True** |
| **P3** | **0632** | **0020** | **2 14 11 8** | **True** |
| **P4** | **0014** | **0642** | **2 14 12 12** | **True** |

存在安全序列 {P0, P2, P3, P4, P1}，系统处于安全状态。

1. 若P1发出请求0420，

（1）0420 < 0750，即Reqest[i,j] < Need[i,j]，转下一步

（2）0420 <= 1520，即Request[i,j] <= Available[j]，转下一步

（3）系统尝试将资源分配给进程P1，得到

|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Allocation***  **A B C D** | ***Need***  **A B C D** | ***Available***  **A B C D** |
| **P1** | **1420** | **0330** | **1100** |

（4）安全性算法：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Allocation***  **A B C D** | ***Need***  **A B C D** | ***Work***  **A B C D** | ***Finish*** |
|  |  |  | **1 1 0 0** |  |
| **P0** | **0012** | **0000** | **1 1 1 2** | **True** |
| **P1** | **1420** | **0330** | **3 14 12 12** | **True** |
| **P2** | **1354** | **1002** | **2 4 6 6** | **True** |
| **P3** | **0632** | **0020** | **2 10 9 8** | **True** |
| **P4** | **0014** | **0642** | **2 10 10 12** | **True** |

存在安全序列 {P0, P2, P3, P4, P1}，系统处于安全状态。

因此P1的请求会被立即同意。