|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deadlock | 系統中存在一組process彼此形成互相循環等待的關係，使得這些process皆無法繼續執行下去，導致throughput下降的情況 | Deadlock成立的  必要條件 | Mutual exclusion | (to resource)任何時間點，資源只能讓一個process持有，不可多個process同時共用 |
| Hold & wait | Process持有部分資源，仍在等待其他process持有的資源 |
| No preemption | Process不可搶奪其他process所持有的資源，必須等其釋放 |
| Circular waiting | 一群process形成彼此循環等待的關係 |

* 互斥：1.CPU 2.Memory 3.I/O device 沒有互斥：read only file

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Deadlock | 一組process形成circular waiting無法往下執行 | No preemptive成立 | 導致throughput降低 | 解決法：deadlock prevention/avoidance |
| Starvation | 長期無法取的完工所需要的資源形成indefinite blocking | 發生在不公平的環境preemptive | 無關 | 解決法：aging |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

* No cycle no deadlock；有cycle不一定deadlock；如果每一資源只有單一個有cycle一定有deadlock

|  |  |  |  |
| --- | --- | --- | --- |
| Deadlock的處理方式 | 名稱 | 優點 | 缺點 |
| deadlock prevention | 保證不會有deadlock | 1. resources utilization低；throughput低 2. 可能有starvation |
| deadlock avoidance | 保證不會有deadlock | 1. resources utilization低；throughput低 2. 可能有starvation |
| deadlock detection& recovery | resources utilization高；throughput高 | 1. 可能有deadlock 2. 成本高 |
| ignored |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Deadlock的處理方式 | deadlock prevention | 原則：  打破四個必要條件之一 | 原則 | | 做法 | 原因 / 缺點 |
| Mutual exclusion | | 做不到 | 資源本身特性 |
| Hold & wait | | 規定process要取得完工所需要的所有資源，才可以持有resource | 資源利用度低，可能有starvation |
| Process可以先持有resource，但要申請其他resource之前必須要放掉所有resource | 資源利用度低，可能有starvation |
| No preemption | | 改成 preemption |  |
| Circular waiting | | 1. 給每一種resource一個唯一的ID 2. Process必須依照resource ID**遞增**的方式提出申請 |  |
| deadlock avoidance | 當資源提出申請，執行banker’s algo判斷若同意申請是否處於safe狀態，若是責核准否則為unsafe否決，須等一段時間再申請 | | | |  |
| deadlock detection& recovery | 消除deadlock恢復正常 | | Kill process in deadlock | 1.kill all process | 成本高 |
| 2.kill one process each time：若deadlock仍存在repeat | 成本高 |
| Resource preemption | 1. 選victim process 2. 搶奪他們的resource 3. 恢復這些資源原先的狀態 | 成本高 |