

Deadlock : jalan buntu yang terjadi ketika ada satu atau lebih transaksi.

- Kondisi :
- mutual exclusion
 - hold and wait
 - non-preemption condition.
 - circular wait condition.

- Penangan Deadlock :
- Mengabaikan permasalahan
 - recovery
 - pencegahan
 - pengalokasian sumber daya yang efisien.

Penjelasan :

Deadlock adalah keadaan dimana sejumlah permintaan yang tidak bisa dijalankan oleh scheduler karena permintaan-permintaan tersebut saling tunggu-menunggu. Deadlock adalah masalah utama dalam penggunaan data yang digunakan secara bersama-sama.

Kondisi yang menyebabkan deadlock :

1. Mutual exclusion :

Mutual exclusion adalah suatu kondisi yang menjamin hanya satu proses saja yang berjalan dalam suatu critical region / section. Critical section adalah sebuah segmen kode di mana sebuah proses yang menggunakan sumber daya bersama diakses. Jadi, mutual exclusion adalah suatu kondisi dimana terdapat sumber daya yang tidak dapat dipakai bersama.

2. Kondisi gangguan dan tunggu (hold and wait)

Kondisi hold and wait adalah sebuah kondisi di mana proses harus dihentikan dari menggenggam satu atau beberapa proses di saat menunggu proses lainnya. Kondisi ini terjadi ketika proses yang memegang sumber daya masih bisa meminta sumber daya lain.

3. Non pre-emption condition

Kondisi ketika sumber daya yang sedang digunakan oleh suatu proses tidak bisa sembarangan diambil dari proses tersebut. Melainkan harus dilepaskan secara sendirinya oleh proses.

4. Circular wait condition

Kondisi ketika setiap proses menunggu sumber daya dari proses berikutnya yang sedang dipakai oleh proses lain.

Penanganan Deadlock.

1) Mengabaikan permasalahan (The Ostrich Algorithm)

Algoritma ini adalah strategi mengabaikan masalah yang mungkin terjadi atas dasar bahwa masalah itu mungkin sangat jarang terjadi. Dengan mengasumsikan bahwa lebih efektif untuk memungkinkan masalah itu terjadi dibandingkan upaya pencegahannya.

2) Peteksi & Pemulihan (Recovery)

Mendeteksi adanya deadlock dengan menganalisis / mengetahui di saat / kapan deadlock sering terjadi. Recovery atau pemulihan dilakukan dengan menggagalkan semua proses deadlock, memback-up dan merestart semua proses yang deadlock, dsb.

3) Pencegahan.

Dilakukan dengan meniadakan salah satu dari empat kondisi deadlock. Havender mengemukakan bahwa jika dari keempat penyebab tidak terpenuhi maka tidak akan terjadi deadlock.

4) Pengalokasian sumber daya yang efisien

Membuat proses hanya dilayani permintaannya bila semua sumber daya yang diperlukan tersedia. Jika sumber daya tersedia, proses dialokasikan pada yang diperlukan dan berjalan hingga selesai, dan sebaliknya.