

- DEADLOCK -

1. Apa itu Deadlock?

: Deadlock adalah keadaan dimana sejumlah permintaan yang tidak bisa dijalankan oleh scheduler karena permintaan-permintaan tersebut saling menunggu menunggu. Deadlock terjadi karena penggunaan data secara bersama-sama.

2. Kondisi untuk mencapai Deadlock

: a.) Mutual Exclusion (Mutual exclusion conditional)

→ kondisi ini terjadi apabila suatu proses telah menggunakan suatu resource, maka tidak boleh ada proses lain yang menggunakan resource tersebut. Hanya satu proses saja yang dapat menggunakan sebuah resource pada satu waktu.

b.) Kondisi genggam dan tunggu (Hold and wait)

→ Kondisi ini terjadi ketika suatu proses meminta izin untuk mengakses resource, proses tersebut dapat meminta izin untuk mengakses resource lain yang dipakai oleh proses lain.

c.) Kondisi No-preemption (No Preemption condition)

→ Kondisi ini terjadi jika suatu proses meminta izin untuk mengakses resource, sementara resource tersebut tidak tersedia, maka permintaan izin tidak dapat dibatalkan.

d.) Kondisi menunggu secara Sirkuler (Circular Wait Condition)

→ Kondisi ini terjadi jika proses P0 sedang mengakses Resource R1 dan minta izin untuk mengakses resource R1 dan minta izin untuk mengakses resource R2, dan pada saat yang bersamaan P1 sedang mengakses resource R2 dan minta izin untuk mengakses R1.

3. Penanganan Deadlock.

: a.) Mengabaikan permasalahan (The Ostrich Algorithm)

→ Untuk memastikan sistem tidak memasuki deadlock, sistem dapat menggunakan pencegahan deadlock atau penghindaran deadlock. Penghindaran deadlock membutuhkan informasi tentang sumber daya mana yang akan suatu proses meminta dan berapa lama akan digunakan. Dengan informasi tersebut dapat diputuskan apakah suatu proses harus menunggu atau tidak. Hal ini disebabkan oleh keberadaan sumber daya, apakah ia sedang digunakan oleh proses lain atau tidak.

b.) Deteksi dan Pemulihan (Recovery)

→ Pada suatu proses akan terdeteksi sistem mana yang terlibat dalam deadlock. Setelah tahu sistem mana yang terlibat, maka diadakan proses untuk memperbaiki dan menjadikan sistem berjalan kembali.

c.) Pencegahan, dengan meniadakan salah satu dari empat kondisi deadlock.

→ Dengan ini kita bisa menjalankan sistem secara satu persatu, tidak bersamaan.

d.) Pengalokasian sumber daya yang efisien

→ Pada suatu proses dimana kita memilih sistem yang akan dikorbankan atau diambil sumberdayanya untuk sementara waktu. Sehingga kita bisa memaksimalkan sistem yang utama.