

{Kondisi Untuk Mencapai Dead lock}

1

Mutual exclusion

Merupakan proses yang telah menggunakan suatu resource, maka tidak boleh ada proses lain yang menggunakan resource tersebut. Hanya satu proses yang dapat menggunakan sebuah resource pada satu waktu.

2

Hold & Wait

Merupakan suatu proses yang sedang mengakses suatu resource, proses tersebut dapat meminta lagi untuk mengakses resource lain yang dipunyai oleh proses lain.

4

Circular Wait Condition

Merupakan suatu proses P_0 sedang mengakses Resource R_1 dan meminta lagi untuk mengakses resource R_2 dan meminta izin untuk mengakses Resource R_2 ; dan pada saat yang bersamaan P_2 sedang mengakses Resource R_2 dan meminta izin untuk mengakses R_1 .

3

No Preemption

Merupakan suatu proses meminta izin untuk mengakses resource, sementara resource tersebut tidak tersedia maka permintaan izin tidak dapat dibatalkan.

Penanganan Deadlock

- 1 Mengabaikan permasalahan
Mengabaikan permasalahan atau yang bisa juga disebut sebagai The ostrich algorithm merupakan strategi mengabaikan masalah itu sendiri yang mungkin sangat jarang terjadi untuk itu muncul istilah seperti seolah kita di pesta dan berpura-pura bahwa tidak ada masalah.

Cara pengaplikasian Algoritma Ostrich:

- o) Jangan lakukan apapun, cukup restart sistem

2 Deteksi dan pemulihan

- o) Metode ini dapat juga disebut sebagai Detection and Recovery. Pada metode ini yang merupakan pemulihan deadlock biasanya untuk menghilangkan deadlock dari sistem sehingga sistem beroperasi kembali, dapat terlepas dari deadlock.
- o) Metode ini juga bertujuan untuk memastikan apakah telah terjadi deadlock dan menentukan proses-proses dan sumberdaya yang terlibat deadlock secara presisi. Dengan begitu dapat dilakukan sistem pemulihan dari deadlock dengan metode pemulihan.

3

Pencegahan dengan menghindari salah satu dari empat kondisi deadlock

Agar dapat mencegah terjadinya deadlock, terdapat 4 strategi yang dapat dilakukan, antara lain :

1). Menghindakan Mutual Exclusion

Yaitu dengan melakukan 'spooling program' yang harus disediakan sebagai proses. Dengan spooling + penitikan? yang dapat diartikan di hardware. Setiap job di antrean spooler akan dilayani satu persatu.

2). Menghindakan Hold and Wait

=> Mengalokasikan semua sumber daya atas tidak sama sekali

=> Hold and Release

3). Menghindakan No - preemption

4). Menghindakan Menunggu Struktur

=> Proses hanya dipbolehkan menggunakan satu sumber daya

=> Penomoran global semua sumber daya

4

Pengalokasian Sumber Daya yang Efisien

Mempakai suatu sumber daya yang dapat digunakan dengan aman dengan memuat satu proses adanya. Sehingga sumber daya yang ada di dalamnya akan lebih efisien.