

Kondisi untuk mencapai Deadlock

1. Mutual Exclusion

Mutual Exclusion adalah keadaan atau kondisi dimana hanya ada satu proses yang dapat mengakses/menggunakan sumber daya.

Contoh: hanya ada satu proses pada satu waktu yang diperbolehkan untuk mengirimkan perintah kepada printer

2. Kondisi genggam dan tunggu

Proses - proses yang sedang menggenggam sumber daya, menunggu sumber daya - sumber daya baru

3. Kondisi non - Preemption

Sumber daya - sumber daya yang sebelumnya diberikan tidak dapat diambil paksa dari proses itu.

Sumber daya harus secara eksplisit dilepaskan dari proses yang menggenggamnya.

4. Kondisi menunggu secara sirkuler

Terdapat sekumpulan proses ($P_0, P_1 \dots P_n$) yang menunggu R, dimana P_0 menunggu R yang dibawa P_1 , P_1 menunggu R yang dibawa P_2 dst $\rightarrow P_{n-1}$ menunggu R yang dibawa P_n

Model Deadlock

Deadlock dapat digambarkan sebagai graph

- misal: 2 proses P_0 dan P_1
- 2 sumber daya R_0 dan R_1
- P_0 meminta sumber daya R_0
- Sumber daya R_1 dialokasikan ke P_1 .

Penanganan Deadlock

1. Mengabaikan Permasalahan

dalam ilmu komputer, algoritme ostrich adalah strategi mengabaikan masalah yang mungkin terjadi atas dasar bahwa masalah itu mungkin sangat jarang terjadi.

2. Deteksi dan Pemulihan

- metode deteksi digunakan pada sistem yg memungkinkan terjadinya deadlock. Tujuan metode ini adalah memeriksa apakah telah terjadi deadlock dan menentukan Proses - Proses dan sumber daya ? yg terlibat deadlock.
- metode Pemulihan dari deadlock berupaya untuk menghilangkan deadlock dari sistem sehingga sistem beroperasi kembali, bebas dari deadlock.

3. Pencegahan, dengan meniadakan salah satu dari empat kondisi deadlock

Pengkondisian sistem agar menghilangkan kemungkinan terjadinya deadlock. Pencegahan merupakan solusi yang bersih dipandang dari sudut terpegahnya deadlock.

4. Pengalokasian sumber daya yang efisien.

Situasi ketika sumber daya dialokasikan pada penggunaan nilai tertinggi mereka dapat digunakan dengan aman oleh satu proses pada suatu saat.