

DEADLOCK

A. Kondisi untuk mencapai Deadlock

1. Mutual exclusion (mutual exclusion conditional): hanya satu proses yang mengakses sumber daya pada satu waktu interval tertentu.
2. Kondisi genggam dan tunggu (hold and wait): Suatu proses yang sedang mengakses suatu resource, proses tersebut dapat meminta izin untuk mengakses resource lain yang dipakai oleh proses lain
3. Kondisi non-preemption (non-preemption condition): Suatu proses meminta izin untuk mengakses resource, sementara resource tsb tidak tersedia, maka permintaan izin tidak dapat dibatalkan
4. Kondisi menunggu secara sirkuler (circular wait condition): Terdapat sekumpulan proses $\{P_0, P_1, \dots, P_n\}$ yang menunggu sumber daya dimana P_0 menunggu sumber daya yang dibawa P_1 , P_1 menunggu sumber daya yang dibawa P_2 , dst.

B. Penanganan Deadlock

1. Mengabaikan permasalahan (the ostrich algorithm): teknik yg paling sering digunakan dengan mengabaikan permasalahan & seakan-akan tidak pernah terjadi. Teknik ini dilakukan dg asumsi bahwa lebih efektif untuk memungkinkan masalah itu terjadi dibandingkan upaya pencegahannya.
2. Deteksi & pemulihan (recovery): metode deteksi digunakan pada sistem yang mengijinkan terjadinya deadlock untuk menentukan proses & sumber daya yang terlibat deadlock scr presisi. Setelah dapat ditentukan, sistem dipulihkan dari deadlock dengan metode pemulihan. Metode pemulihan berupaya untuk menghilangkan deadlock sehingga sistem dapat beroperasi kembali
3. Pencegahan dengan meniadakan salah satu dari 4 kondisi deadlock: pengondisian sistem agar menghilangkan salah satu dari 4 kondisi deadlock
4. Pengalokasian sumber daya yang efisien: menentukan sumber daya yang aman digunakan pada suatu proses

*tambahan poin ketiga untuk penanganan deadlock, cara menghilangkan satu dari empat kondisi deadlock

- a. Mutual exclusion : spooling sumber daya
- b. Hold and wait : meminta sumber daya di awal
- c. No pre-emptive : mengambil sumber daya di tengah proses
- d. Circular wait : penomoran permintaan sumber daya.