

# DEADLOCK.

## o> Apa itu Dead lock ?

Dead lock adalah jalan buntu yang dapat terjadi ketika dua atau lebih transaksi masing-masing menunggu lock yang sedang dipegang oleh transaksi lainnya untuk dilepas.

## • > Kondisi untuk mencapai Deadlock

- A. Mutual exclusion (mutual exclusion condition)
- B. Kondisi genggam dan Tunggu (hold and wait)
- C. Kondisi non-preemption (non-preemption condition)
- D. Kondisi menunggu secara sirkuler (circular wait condition)

## A. Mutual exclusion (mutual exclusion condition)

Merupakan objek program yang mencegah akses simultan ke sumber daya bersama. Dimana ada jepotang kode mengakses sumber daya bersama. Hanya satu mutex dengan nama unik, harus mengunci mutex dari atas lain.

## B. Kondisi genggam dan Tunggu (hold and wait)

Suatu proses membawa sedikitnya satu sumber daya menunggu mendapatkan tambahan sumber daya baru yang dibawa oleh proses.

## C. Kondisi non-preemption (non-preemption condition)

Kondisi dimana sebuah sumber daya dapat dibebaskan dengan sukarela oleh proses yang memegangnya setelah proses menyelesaikan task.

## D. Kondisi menunggu secara sirkuler (circular wait condition)

Kondisi ketika terdapat sekumpulan proses  $\{P_1, P_2, \dots, P_n\}$  yang menunggu sumber daya dimana  $P_1$  menunggu sumber daya yang dibawa  $P_2$ ,  $P_2$  menunggu sumber daya yang dibawa  $P_3$  dan seterusnya,  $P_{n-1}$  menunggu sumber daya yang dibawa oleh  $P_n$  dan  $P_n$  menunggu sumber daya yang dibawa  $P_1$ .

## •> Penanganan Deadlock

- A. Mengabaikan Permasalahan (The Ostrich Algorithm)
- B. Deteksi dan Pemulihan (recovery)
- C. Pencegahan, dengan meniadakan salah satu dari empat kondisi deadlock
- D. Pengalokasian sumber daya yang efisien.

## A. Mengabaikan Permasalahan (The Ostrich Algorithm)

Mengabaikan permasalahan seakan-akan deadlock tidak pernah terjadi pada sistem. Model ini yang banyak digunakan pada sistem operasi termask UNIX

## B. Deteksi dan Pemulihan (recovery)

Deteksi digunakan pada sistem yang memungkinkan terjadinya deadlock, dengan memeriksa apakah terjadi deadlock dan menentukan proses dan sumber daya yang terlibat deadlock secara presisi, begitu selesai, sistem dipulihkan sehingga beroperasi kembali.

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

Pencegahan merupakan ~~kondisi~~ solusi yang berhasil dipandang dari sudut tercegahnya deadlock. Untuk mencegah deadlock dilakukan meniadakan salah satu dari empat kondisi deadlock yaitu:

- Mencegah mutual Exclusion
- Mencegah hold and wait
- Mencegah Non preemption
- Mencegah kondisi Menunggu sirkular.

D. Pengalokasian Sumber daya yang efisien

Mengatakan Jumlah kebutuhan Sumber daya maksimum sebelum eksekusi.

Begitu eksekusi dimulai, tiap proses meminta sumber daya saat diperlukan sampai batas maksimum yang dinyatakan di awal. Proses-proses yang menyatakan kebutuhan sumber daya melebihi kapasitas total sistem tidak dapat dieksekusi.