

## Deadlock

### 1. Deadlock

Merupakan suatu keadaan 2 proses atau lebih yang saling membutuhkan lebih dari satu sumber daya (resource) dan masing-masing dari proses tersebut telah mendapat akses resource dan tidak ingin melepaskannya sebelum mendapat resource lain dan menyelesaikan prosesnya.

### 2. Kondisi untuk Mencapai Deadlock

#### a. Mutual exclusion (mutual exclusion conditional)

Suatu kondisi dimana hanya ada 1 proses yang boleh memakai sumber daya (resources) dan proses lainnya yang hendak memakai resources tadi ditanamkan menunggu hingga resources tersebut dilepaskan oleh proses sebelumnya atau sudah tidak ada lagi proses yang menggunakan resources tersebut.

#### b. Kondisi Genggam dan Tunggu (Hold and Wait)

Suatu kondisi di mana suatu proses yang sedang memakai sumber daya (resources) membutuhkan tambahan resource untuk dapat menjalankan prosesnya dan jika resource tambahan yang dibutuhkan sedang digunakan oleh proses lain, maka proses yang membutuhkan tambahan resource ditanamkan menunggu sampai resource tersebut dilepaskan.

#### c. Kondisi non-preemption

Suatu kondisi di mana sumber daya (resource) yang sedang dipakai oleh suatu proses tidak dapat diambil secara paksa oleh proses lainnya, tetapi harus dilepas secara eksplisit oleh proses yang sedang menggunakannya.

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

Suatu kondisi di mana proses menunggu dalam model sirkuler, yaitu proses yang membutuhkan sumber daya (resource) yang dimiliki oleh proses berikutnya.

### 3. Pencegahan Deadlock

#### a. Mengabaikan Permasalahan (The Ostrich Algorithm)

Merupakan penanganan dengan burpura-pura (mengabaikan) tidak terjadi masalah apapun. Sistem operasi tidak akan mendeteksi deadlock dan dibiarkan secara otomatis memalukan program.



## b. Deteksi dan Pemulihan (Recovery)

Merupakan penanganan dengan dilakukan deteksi terhadap sistem yang mengizinkan terjadinya deadlock untuk memeriksa adanya keadaan deadlock. Kemudian dilakukan penghentian proses yang menyebabkan deadlock sehingga sistem dapat beroperasi kembali.

## c. Pencegahan dengan meniadakan salah satu dari 4 kondisi deadlock

Merupakan penanganan dengan dilakukan pengendalian sistem dengan cara pencegahan agar tidak menyebabkan deadlock, meliputi:

- \* Mutual exclusion: Membuat file spool untuk resource yang digunakan bersama
- \* Kondisi Gergam dan tunggu: Memaksa sebuah proses untuk melepaskan resources yang dimiliki ketika meminta resource baru.
- \* Circular wait condition: Memberi penamaan resource berdasarkan urutan / level.
- \* Non-preemption: Memperbolehkan adanya preemption.

## d. Pengalokasian sumber daya yang efisien

Merupakan penanganan dengan mengalokasikan sumber daya (resource) yang lebih efisien sehingga akan mengurangi resiko terjadinya deadlock.