

Nama : Zulfar Repe A.

NPM : 21083010122

Mata kuliah: sistem operasi b

### MENGERTI DEADLOCK

Deadlock adalah suatu kondisi dimana dua proses atau lebih tidak dapat meneruskan eksekusinya oleh pemroses. pada umumnya deadlock terjadi karena proses mengalami starvation, yaitu suatu job yang sedang direkrutasi dan eksekusi job tersebut tidak ada henti-hentinya.

#### ► Kondisi untuk mencapai Deadlock

##### 1. Mutual exclusion

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

##### 2. kondisi Genggam dan tunggu (Hold and wait)

pada suatu proses sedang mengakses suatu resource, proses tersebut dapat meminta izin untuk mengakses resource lain yang dipunyai proses lain.

##### 3. kondisi No preemption

Jika suatu proses meminta izin untuk mengakses resource, sementara resource tersebut terdapat ada, maka permintaan izin dibatalkan.

##### 4. kondisi menunggu secara sirkuler (Circular wait condition)

Jika proses P0 sedang mengakses Resource R1 dan meminta izin mengakses resource R2, dan pada saat bersamaan P1 sedang mengakses resource R2 dan mint izin untuk mengakses R1. Harus terdapat rantai sirkuler dari dua proses atau lebih, dan masing-masing proses menunggu sumber daya yang digenggam oleh proses berikutnya.

## ► Penanganan Deadlock

### 1. Mengabaikan permasalahan (The Ostrich Algorithm)

Algoritma penanganan Deadlock dengan cara mengabaikan setiap permasalahan yang muncul. Dengan asumsi permasalahan akan sangat jarang muncul. Algoritma Ostrich dibagi menjadi 2 pendekatan:

- Trade off, berarti bahwa jika kondisi belum teridentifikasi maka masalah yang sangat jarang terjadi dapat kembali terjadi.
- Pendekatan Hybrid, menentukan bahwa deadlock sangat jarang atau bahkan tidak pernah terjadi sama sekali.

### 2. Deteksi dan pemulihan (recovery)

Mendeteksi sistem mana yang mengalami deadlock. Setelah diketahui maka diadakan proses untuk memperbaiki dan menjadikan sistem berjalan kembali.

### 3. Pencegahan dengan meniadakan salah satu dari 4 kondisi

kondisi sistem menghilangkan kondisi penyebab deadlock. Meniadakan mutual exclusion, meniadakan hold and wait, meniadakan non preemption, meniadakan menunggu sirkular.

### 4. Pengalokasian sumber daya yang efisien.

sumber daya yang dapat digunakan dengan aman oleh satu proses pada suatu saat atau sumber daya yang dapat dibuat atau ditenyapkan.