Elfira Ratna Syaharani
2108301005G

No Rabu
Date 5 Oktober 2022

Kondisi untuk Mencapai Deadlock

1. Mutual Exclusion (Mutual Exclusion Conditional)
   ↳ Jika suatu proses menggunakan suatu resource, tidak ada proses lain yang boleh menggunakan resource tersebut.

   Pencegahan : Membuat resource shareable
2. Kondisi Genggam dan Tunggu (Hold and Wait)
   ↳ Jika pada suatu proses mengakses sebuah resource, proses tersebut dapat meminta ijin untuk mengakses resource lain

   Pencegahan : Proses harus melepas resource yang dibawanya sebelum meminta resource lainnya
3. Kondisi non-preemption (non-Preemption Condition)
   ↳ Jika suatu proses meminta ijin untuk mengakses resource, sementara resource tidak tersedia, maka permintaan tidak dapat dibatalkan

   Pencegahan :
   - Pembebasan semua resource yang dipegang suatu proses apabila proses ingin mengakses semua resource lain, dan tidak dapat langsung dipenuhi
   - Resource dengan preemption ditambahkan pada proses yang ingin mengakses resouree lain tsb.
   - Proses dimulai kembali apabila sudah mendapatkan kembali semua resource yang dilepaskan termasuk resource yang ingin diakses
4. Kondisi menunggu secara sirkuler (Circular wait condition)
   ↳ Jika proses Pi sedang mengakses resource Ri, dan meminta ijin untuk mengakses resource Rj, dan pada saat bersamaan proses Pj sedang mengakses Rj dan minta ijin untuk mengakses Resource Ri.

   Pencegahan : Memberi nomor pada setiap resource yang ada dan diakses secara berurutan.

Penanganan Deadlock

1. Mengabaikan Permasalahan (The Ostrich Algorithm)
   ↳ Deadlock diabaikan dan diasumsikan bahwa masalah tersebut tidak dapat diatasi.

→

No
Date

2. Deteksi dan pemulihan (Recovery)

Deteksi : • Single instance: jika resource allocation graph bersiklus
• Multiple instance: Request-i ≥ Available

Recovery : • Process termination
• Resource termination

3. Pencegahan

Terdapat 2 cara :

1. Tidak memulai proses apapun apabila membawanya pada kondisi deadlock
2. Tidak memberikan kesempatan pada proses yang meminta penambahan sumber daya apabila membawanya pada kondisi deadlock

4. Pengalokasian sumberdaya yang efisien

• Deadlock avoidance system untuk mendata INFORMASI yang tambahan tentang proses mana yang akan meminta dan menggunakan sumber daya