Nama: Ahmad Wafi Fathurrahman
NIM: 260 8301 0011

No.
Date.

- Kondisi untuk mencapai Deadlock

1) Mutual exclusion (mutual exclusion conditional)
Apabila proses telah menggunakan suatu resource, maka tidak boleh ada proses lain yang menggunakan resource tersebut. Hanya 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 ijin untuk mengakses resource lain yang dipakai oleh proses lain

3.) Kondisi non-preemption (non-Preemption Condition)
Jika suatu proses meminta itin untuk mengakses resource, sementara resource tersebut tidak tersedia, maka permintaan itin tidak dapat dipatalkan.

4.) Kondisi menunggu secara sirkuler (Circular wait condition)
Jika proses P0 sedang mengakses Resource R1 dan meminta itin untuk mengakses resource R2, dan pada saat yang bersamaan P1 sedang mengakses resource R2 dan minta itin untuk mengakses R1.

No.

Date.

Penanganan Deadlock

1.) Mengabaikan Permasalahan (The Ostrich Algorithm)

Dalam ilmu komputer, algoritma ostrich merupakan strategi mengabaikan masalah yang mungkin terjadi atas dasar pada masalah itu sendiri yang mungkin sangat jarang terjadi. [illegible]

Algoritma tersebut bisa digunakan untuk menangani jika terjadi deadlock pada pemrograman concurren.

Gambaran Algoritma Ostrich.

- Jangan lakukan apapun, cukup restart sistem
- Dilakukan apabila :
  1.) Deadlock sangat jarang terjadi
  2.) Algoritma deadlock lainnya biayanya lebih tinggi

2.) Deteksi dan pemulihan (recovery).

Deteksi digunakan pada sistem yang mengijinkan terjadinya deadlock. Tujuannya adalah untuk memeriksa apakah telah terjadi deadlock dan menentukan proses- proses dan sumberdaya - sumberdaya yang terlibat deadlock secara presisi. Begitu telah ditentukan, sistem dipulihkan dari deadlock dengan metode pemulihan.

No.

Date.

Pemulihan dari deadlock berupaya untuk menghilangkan deadlock dari sistem sehingga sistem beroperasi kembali, bebas dari deadlock. Proses - proses yang terlibat deadlock mungkin dapat menyelesaikan eksekusi dan membebaskan sumberdaya - sumber dayanya.

3.) Pencegahan, dengan menradakan salah satu dari empat kondisi deadlock.

* Menradakan Mutual Exclusion

Melakukan spooling perangkat - perangkat yang haknya di dedikasikan ke suatu proses. Dengan spooling, permintaan - permintaan di antrikan di hard disk. setiap job di antrikan spooler akan dilayani satu per satu.

* Menradakan Hold and Wait

1) Mengalokasikan semua sumber daya atau tidak sama sekali:

2) Hold and release

* Menradakan Non-Preemption

* Menradakan Menunggu Sirkular

- Proses hanya di perbolehkan mengsenggam satu sumberdaya
- Penomoran global semua sumber daya.

a.) Pengalokasian sumber daya yang efisien.

Merupakan suatu sumber daya yang dapat digunakan dengan aman dengan memuat satu proses adanya, sehingga sumber daya yang ada di dalamnya akan lebih efisiens.