

Nama: Yosua Satria Boru Harmoni

No. _____

NPM: 21083010029

Date: _____

Kondisi untuk Mencapai Deadlock

1.

Mutual Exclusion Conditional

Apabila proses telah menggunakan suatu resource, maka tidak boleh ada proses lain yang menggunakan resource tersebut. Hanya 1 proses yang dapat menggunakan sebuah resource pada satu waktu.

2.

Hold and wait

Jika suatu proses sedang mengakses suatu resource, proses tersebut dapat meminta izin untuk mengakses resource lain yang dipakai oleh proses lain.

3.

No preemption

Jika suatu proses meminta izin untuk mengakses resource, sementara resource tersebut tidak tersedia, maka permintaan izin tidak dapat dibatalkan.

4.

Circular wait Condition

Jika proses P_0 sedang mengakses Resource R_1 dan meminta izin untuk mengakses resource R_2 kemudian meminta izin untuk mengakses resource R_2 , dan pada saat yang sama P_1 sedang mengakses resource R_2 dan meminta izin untuk mengakses R_1 .

Penanganan Deadlock

1. Mengabaikan permasalahan (the ostrich Algorithm)
 Dalam computer science, the ostrich algorithm merupakan strategi mengabaikan masalah yang mungkin terjadi atas dasar pada masalah yang sendiri yang mungkin sangat jarang terjadi. Algoritma tersebut bisa digunakan untuk menangani jika terjadi deadlock pada pemrograman concurren.

Gambaran Algoritma Ostrich:

1. Jangan lakukan apapun cukup restart system
2. Dilakukan jika deadlock sangat jarang terjadi, algoritma deadlock lainnya biayanya lebih tinggi.

2. Deteksi dan pemulihan (recovery)

Metode deteksi digunakan pada system yang mengizinkan terjadinya deadlock. Tujuan metode ini adalah memeriksa apakah telah terjadi deadlock dan menentukan proses-proses dan sumberdaya-sumberdaya yang terlibat deadlock secara presisi. Begitu telah dapat ditentukan, sistem dipulihkan dari deadlock dengan metode pemulihan. Metode pemulihan dari deadlock dari sistem sehingga sistem beroperasi kembali, bebas dari deadlock. Proses-proses yang terlibat deadlock mungkin dapat menyelesaikan eksekusi dan membebaskan sumberdaya-sumberdayanya.

Penanganan Deadlock

1. Mengabaikan permasalahan (the ostrich Algorithm)
- Dalam computer science, the ostrich algorithm merupakan strategi mengabaikan masalah yang mungkin terjadi atas dasar pada masalah itu sendiri yang mungkin sangat jarang terjadi. Algoritma tersebut bisa digunakan untuk menangani jika terjadi deadlock pada pemrograman concurren.

Gambaran Algoritma Ostrich:

1. Jangan lakukan apapun, cukup restart system
2. Dilakukan jika deadlock sangat jarang terjadi, algoritma deadlock lainnya biayanya lebih tinggi.

2. Deteksi dan pemulihan (recovery)

Metode deteksi digunakan pada system yang mengizinkan terjadinya deadlock. Tujuan metode ini adalah memeriksa apakah telah terjadi deadlock dan menentukan proses-proses dan sumberdaya-sumberdaya yang terlibat deadlock secara presisi. Begitu telah dapat ditentukan, sistem dipulihkan dari deadlock dengan metode pemulihan. Metode pemulihan dari deadlock dari sistem sehingga sistem beroperasi kembali, bebas dari deadlock. Proses-proses yang terlibat deadlock mungkin dapat menyelesaikan eksekusi dan membebaskan sumberdaya-sumberdayanya.

3. Pencegahan dengan meniadakan salah satu dari empat kondisi deadlock.
- Empat kondisi deadlock:
- a. Meniadakan mutual Exclusion
melakukan spooling perangkat-perangkat yang harus dideklasikan ke suatu proses. Dengan spooling, permintaan-permintaan diantrekan di hard disk. Setiap job di antrean spooler akan dilayani satu per satu.
 - b. Meniadakan hold and wait.
 - Mengalokasikan semua sumber daya atau tidak sama sekali.
 - Hold and Release
 - c. Meniadakan NN-preemption
 - d. Meniadakan menunggu sirkular.
 - Proses hanya diperbolehkan memegang satu sumber daya.
 - Penomoran global semua sumber daya.

4. Pengalokasian sumber daya yang efisien.
- Sumber daya yang efisien merupakan suatu sumber daya yang dapat digunakan dengan aman dengan memuat satu proses a daya, sehingga sumber daya yang didalamnya akan lebih efisien.