Nama : Zidan Zulkhairyan Mandalim

Kelas : D

NIM : 1800018208

***asasasa***

***Deadlock*** adalah suatu kondisi dimana dua proses atau lebih saling menunggu proses yang lain untuk melepaskan resource yang sedang dipakai. Karena beberapa proses itu saling menunggu, maka tidak terjadi kemajuan dalam kerja proses-proses tersebut. Deadlock dalam arti sebenarnya adalah kebuntuan. Kebuntuan yang dimaksud dalam sistem operasi adalah kebuntuan proses. Deadlock adalah masalah yang biasa terjadi ketika banyak proses yang membagi sebuah resource yang hanya boleh diubah oleh satu proses saja dalam satu waktu. Jadi deadlock adalah suatu kondisi dimana proses tidak berjalan lagi atau pun tidak ada komunikasi lagi antar proses.

4 syarat terjadinya deadlock :

a. Mutual Exclusion, suatu kondisi dimana setiap sumber daya diberikan tepat pada satu proses pada satu waktu.

b. Hold and Wait, kondisi yang menyatakan proses-proses yang sedang memakai suatu sumber daya dapat meminta sumber daya yang lain.

c. Non-preemptive, kondisi dimana suatu sumber daya yang sedang berada pada suatu proses tidak dapat diambil secara paksa dari proses tersebut, sampai proses itu melepaskannya.

d. Circular Wait, kondisi yang menyatakan bahwa adanya rantai saling meminta sumber daya yang dimiliki oleh suatu proses oleh proses lainnya.

STRATEGI MENGHADAPI DEADLOCK

Dibagi menjadi 3 pendekatan :

a. Mengabaikan adanya deadlock.

b. Memastikan bahwa deadlock tidak akan pernah ada, baik dengan metode pencegahan, dengan mencegah empat kondisi deadlock agar tidak akan pernah terjadi. Metode menghindari deadlock, yaitu mengizinkan empat kondisi deadlock, tetapi menghentikan setiap proses yang kemungkinan mencapai deadlock.

c. Membiarkan deadlock untuk terjadi, pendekatan ini membutuhkan dua metode yang saling mendukung, yaitu :

1. Pendeteksian deadlock, untuk mengidentifikasi ketika deadlock terjadi.

2. Pemulihan deadlock, mengembalikan kembali sumber daya yang dibutuhkan pada proses yang memintanya.

CONTOH METODE MENGHADAPI DEADLOCK

1.STRATEGI OSTRICH

Pendekatan yang paling sederhana adalah dengan menggunakan strategi burung unta: masukkan kepala dalam pasir dan seolah-olah tidak pernah ada masalah sama sekali. Beragam pendapat muncul berkaitan dengan strategi ini. Menurut para ahli Matematika, cara ini sama sekali tidak dapat diterima dan semua keadaan deadlock harus ditangani. Sementara menurut para ahli Teknik, jika komputer lebih sering mengalami kerusakkan disebabkan oleh kegagalan hardware, error pada kompilator atau bugs pada sistem operasi.

Maka ongkos yang dibayar untuk melakukan penanganan deadlock sangatlah besar dan lebih baik mengabaikan keadaan deadlock tersebut. Metode ini diterapkan pada sistem operasi UNIX dan MINIX.

2. MENCEGAH DEADLOCK

Metode ini merupakan metode yang paling sering digunakan. Metode Pencegahan dianggap sebagai solusi yang bersih dipandang dari sudut tercegahnya deadlock. Tetapi pencegahan akan mengakibatkan kinerja utilisasi sumber daya yang buruk.

Metode pencegahan menggunakan pendekatan dengan cara meniadakan empat syarat yang dapat menyebabkan deadlock terjadi pada saat eksekusi Coffman (1971).

Syarat pertama yang akan dapat ditiadakan adalah Mutual Exclusion, jika tidak ada sumber daya yang secara khusus diperuntukkan bagi suatu proses maka tidak akan pernah terjadi deadlock. Namun jika membiarkan ada dua atau lebih proses mengakses sebuah sumber daya yang sama akan menyebabkan chaos. Langkah yang digunakan adalah dengan spooling sumber daya, yaitu dengan mengantrikan job-job pada antrian dan akan dilayani satu-satu.

Masalah yang mungkin terjadi:

1.Sulitnya mengetahui berapa sumber daya yang dibutuhkan pada awal proses

2.Tidak optimalnya penggunaan sumber daya jika ada sumber daya yang digunakan hanya beberapa waktu dan tidak digunakan tapi tetap dimiliki oleh suatu proses yang telah memintanya dari awal.

Meniadakan syarat ketiga non preemptive ternyata tidak lebih menjanjikan dari meniadakan syarat kedua, karena dengan meniadakan syarat ketiga maka suatu proses dapat dihentikan ditengah jalan. Hal ini tidak dimungkinkan karena hasil dari suatu proses yang dihentikan menjadi tidak baik.

Cara terakhir adalah dengan meniadakan syarat keempat circular wait. Terdapat dua pendekatan, yaitu:

1.Mengatur agar setiap proses hanya dapat menggunakan sebuah sumber daya pada suatu waktu, jika menginginkan sumber daya lain maka sumber daya yang dimiliki harus dilepas.

2.Membuat penomoran pada proses-proses yang mengakses sumber daya. Suatu proses dimungkinkan untuk dapat meminta sumber daya kapanpun, tetapi permintaannya harus dibuat terurut.

Masalah yang mungkin terjadi dengan mengatur bahwa setiap proses hanya dapat memiliki satu proses adalah bahwa tidak semua proses hanya membutuhkan satu sumber daya, untuk suatu proses yang kompleks dibutuhkan banyak sumber daya pada saat yang bersamaan. Sedangkan dengan penomoran masalah yang dihadapi adalah tidak terdapatnya suatu penomoran yang dapat memuaskan semua pihak.

**TABEL DEADLOCK**



3. MENGHINDARI DEADLOCK

Pendekatan metode ini adalah dengan hanya memberi kesempatan ke permintaan sumber daya yang tidak mungkin akan menyebabkan deadlock. Metode ini memeriksa dampak pemberian akses pada suatu proses, jika pemberian akses tidak mungkin menuju kepada deadlock, maka sumber daya akan diberikan pada proses yang meminta. Jika tidak aman, proses yang meminta akan di-suspend sampai suatu waktu permintaannya aman untuk diberikan. Kondisi ini terjadi ketika setelah sumber daya yang sebelumnya dipegang oleh proses lain telah dilepaskan. •Kondisi aman yang dimaksudkan selanjutnya disebut sebagai safe-state, sedangkan keadaan yang tidak memungkinkan untuk diberikan sumber daya yang diminta disebut unsafe-state.

**Kondisi Aman (Safe state)**

Suatu keadaan dapat dinyatakan sebagai safe state jika tidak terjadi deadlock dan terdapat cara untuk memenuhi semua permintaan sumber daya yang ditunda tanpa menghasilkan deadlock. Dengan cara mengikuti urutan tertentu.

**Kondisi Tak Aman (Unsafe state)**

Suatu state dinyatakan sebagai state tak selamat (unsafe state) jika tidak terdapat cara untuk memenuhi semua permintaan yang saat ini ditunda dengan menjalankan proses-proses dengan suatu urutan.



Mendeteksi Deadlock Dan Memulihkan Deadlock

**Terminasi Proses**

Metode ini akan menghapus proses-proses yang terlibat pada kondisi deadlock dengan mengacu pada beberapa syarat. Beberapa syarat yang termasuk dalam metode ini adalah, sebagai berikut:

1.Menghapus semua proses yang terlibat dalam kondisi deadlock(solusi ini terlalu mahal).

2.Menghapus satu persatu proses yang terlibat, sampai kondisi deadlock dapat diatasi (memakan banyak waktu).

3.Menghapus proses berdasarkan prioritas, waktu eksekusi, waktu untuk selesai, dan kedalaman dari rollback.

**Resources Preemption**

Metode ini lebih menekankan kepada bagaimana menghambat suatu proses dan sumber daya, agar tidak terjebak pada unsafe condition. Beberapa langkahnya, yaitu:

1.Pilih salah satu -proses dan sumber daya yang akan di-preempt.

2.Rollback Ke safe state yang sebelumnya telah terjadi.

3.Mencegah suatu proses agar tidak terjebak pada starvation karena metode ini.

**Contoh Deadlock Pada Laptop saya**

Ini dalam kondisi tidak terjadi deadlock :



R2 menunjuk ke P1 artinya R2 dialokasikan di P1

R2 menunjuk ke P2 artinya R2 dialokasikan di P2

P1 menunjuk ke R1 artinya P1 meminta R1 untuk menjalankan prosesnya

R4 tidak menunjuk ke siapapun maka R4 tidak digunakan

dan seterusnya..

Ini Kondisi Saat Terjadi Deadlock :



Gambar diatas hampir sama dengan gambar sebelumnya, hanya yang berbeda adalah P3 meminta resource di R2. Padahal sudah kita ketahui bahwa R2 saja sudah dialokasikan ke P1 dan P2, maka dari itu akan memutar-mutar terus menerus dan menimbulkan DEADLOCK.