

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

RIZQI AMALIYAH M.

21083010663

10 Kondisi untuk Mencapai Deadlock

① Mutual Exclusion

↳ Suatu cara yang menjamin jika ada sebuah proses yang menggunakan variabel atau berkas yang sama, maka proses lain akan dikeluarkan dari pekerjaan yang sama. Jadi, mutual exclusion terjadi setiap sumber daya diberikan pada tepat satu proses pada waktu tertentu.

② Kondisi Genggam dan Tunggu

↳ Proses yang sedang memegang sebuah sumber daya boleh meminta sumber daya yang lain, maksudnya menunggu hingga benar-benar sumber daya yang diminta tidak dipakai oleh proses lain, hal ini dapat menyebabkan kelaparan sumber daya sebab dapat saja sebuah proses tidak mendapat sumber daya dalam waktu yang lama.

③ Kondisi Non - Preemption

↳ Sumber daya yang sedang digunakan oleh sebuah proses tidak dapat diambil secara paksa, tetapi harus dilepas secara eksplisit oleh proses yang sedang menggunakannya.

④ Kondisi Menunggu Secara Sirkuler

↳ Harus ada sebuah rantai melingkar dari dua atau lebih proses yang dijalankan oleh proses yang lain dalam rantai tersebut.

10 Penanganan Deadlock

① Mengabaikan Permasalahan

↳ Metode ini lebih dikenal dengan Algoritma Ostrich. Dalam algoritma ini dikatakan bahwa untuk menghadapi deadlock salah dengan berpura-pura bahwa tidak ada masalah apapun. Hal ini sekiranya akan melakukan suatu hal yang fatal, tetapi sistem operasi Windows dan UNIX menanggulangi deadlock dengan cara ini dengan tidak mendeteksi deadlock dan membiarkannya secara otomatis mematikan program.

② Deteksi dan Pemulihan

↳ Caranya ialah dengan cara mendeteksi terjadi deadlock pada suatu proses maka dideteksi sistem mana yang terlibat di dalamnya. Setelah diketahui sistem mana saja yang terlibat maka diadakan proses untuk memperbaiki dan mengadakan sistem berjalan kembali.

③ Pencegahan dengan Meniadakan Salah Satu dari Empat Kondisi

↳ Ada 2 cara, yaitu:

- Jangan memulai proses apapun jika proses tersebut akan membawa kita pada kondisi deadlock, sehingga tidak mungkin terjadi deadlock karena ketika akan menuju deadlock sudah dicegah.
- Jangan memberi kesempatan pada suatu proses untuk meminta sumber daya lagi jika penambahan ini akan membawa pada suatu keadaan deadlock.

Jadi, diadakan dua kali pengagaan yaitu saat pengalokasian awal, dijaga agar tidak deadlock dan ditambah dengan pengagaan kedua saat suatu proses meminta sumber daya, dijaga agar jangan sampai terjadi deadlock.

④ Pengalokasian Sumber Daya yang Efisien

↳ Sumber daya yang dapat digunakan dengan aman oleh suatu proses pada suatu saat.