

☐ Kondisi untuk mencapai deadlock

☐

☐ - deadlock adalah keadaan dimana sejumlah permintaan yang tidak dijalankan oleh Scheduler karena permintaan-permintaan tersebut saling terus-menerus menunggu

☐

☐ 1. Mutual exclusion

☐

☐ - mutual exclusion adalah keadaan atau kondisi dimana hanya ada satu proses yang dapat mengakses/menggunakan sumber daya. Contoh: hanya ada satu proses pada satu waktu yang diperbolehkan untuk mengirimkan perintah kepada printer

☐

☐ 2. Genggam dan tunggu

☐

☐ - Suatu proses membawa setidaknya satu sumber daya menunggu mendapatkan sumber daya baru yang dibawa oleh proses

☐

☐ 3. Kondisi non-preemption

☐

☐ - sebuah sumber daya dapat dibebaskan dengan sukarela oleh proses yang memegangnya sebelum proses membebaskannya lagi

☐

☐ 4. Kondisi menunggu secara sirkuler

☐

☐ Terdapat sekumpulan proses (P_0, P_1, \dots, P_n) yang menunggu sumber daya dimana P_0 menunggu sumber daya yang dibawa oleh P_1 , P_1 menunggu sumber daya yang dibawa P_2 dan seterusnya

☐

☐

☐

☐

Penanganan Deadlock

1. Mengabaikan permasalahan (Ostetric algorithm)
Suatu algoritma penanganan deadlock dengan cara mengabaikan setiap permasalahan yang muncul, dengan asumsi bahwa permasalahan akan sangat jarang terjadi. Algoritma Ostetric dibagi menjadi 2 pendekatan yaitu trade off, jika kondisi belum teridentifikasi dan pendekatan hybrid menentukan bahwa deadlock sangat jarang

2. Deteksi dan pemulihan (recovery)

mendeteksi jika terjadi deadlock pada state proses, setelah diketahui sistem mana saja yang terlibat maka diadakan proses untuk memperbaiki dan menjalankan sistem berjalan kembali

3. Pencegahan, dengan meniadakan salah satu dari empat kondisi deadlock

- meniadakan mutual exclusion (langkah yang digunakan adalah dengan mensantrikan Job-Job pada antrian dan akan dilayani satu-satu)
- meniadakan hold and wait (langkah yang digunakan adalah dengan membuat proses agar meminta sumber daya yang mereka butuhkan pada awal proses sehingga dapat dialokasikan sumber daya yang dibutuhkan)
- meniadakan non preemptive tidak lebih menjanjikan dari meniadakan syarat kedua karena dengan meniadakan syarat kedua maka satu proses dapat dihentikan di tengah jalan
- meniadakan circular wait, mengatur agar setiap proses hanya dapat menggunakan sebuah sumber daya pada suatu waktu dan membuat penomoran pada proses-proses yang mengakses sumber daya

