

➤ Kondisi untuk mencapai deadlock

↳ Sekumpulan proses yang di blok, dimana setiap proses memegang satu resource dan menunggu resource lain dari proses dalam kumpulan proses yang sedang di blok tersebut, biasanya dari proses-proses atau resource yang non-preemptive.

1) Mutual Exclusion (mutual exclusion conditional)

↳ Jika suatu proses menggunakan suatu resource, tidak ada proses lain yang boleh menggunakan resource tersebut.

2) Kondisi genggam dan tunggu (hold and wait)

↳ Pada saat suatu proses mengakses suatu resource, proses tersebut dapat meminta ijin untuk mengakses resource lain.

3) Kondisi non-preemption (non-preemption condition)

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

4) Kondisi menunggu secara sirkuler (circular wait condition)

↳ Jika proses P_i sedang mengakses resource R_i , dan meminta ijin untuk mengakses resource R_j , dan pada saat bersamaan proses P_j sedang mengakses R_j dan minta ijin untuk mengakses resource R_i .

➤ Penanganan Deadlock

1) Mengabaikan permasalahan (The Ostrich Algorithm)

↳ Menghadapi deadlock dengan berpura-pura tidak ada masalah apapun. Digunakan jika yakin deadlock akan sangat jarang terjadi dan biaya untuk deteksi atau pemulihan deadlock tinggi.

2) Deteksi dan Pemulihan (Recovery)

↳ Deteksi : teknik untuk menentukan apakah deadlock terjadi serta mengidentifikasi proses-proses dan resource-resource yang terlibat deadlock. Biasanya beberapa proses akan kehilangan sebagian atau semua kerja yang telah dilakukan.

↳ Pemulihan : Preemption → mengambil sementara resource dari proses yang menggunakan. Melacak kembali.

3) Pencegahan dengan meniadakan salah satu dari empat kondisi deadlock

↳ Sedapat mungkin tidak mempergunakan mutual exclusion, jika proses sedang memegang resource tertentu, untuk permintaan berikutnya proses harus melepas dulu resource yang dipegang.

4) Pengalokasian Sumber daya yang efisien

↳ Mengelompokkan sumber daya menjadi sejumlah kelas, mengurutkan linear, dalam satu kelas sumber daya, gunakan algoritma yang cocok.