Nama : FITRI INDAH SARI
Kelas : B
NPM : 21083010025

No.
Date

DEADLOCK

* Kondisi untuk mencapai deadlock

1. Mutual exclusion (mutual exclusion conditional)
   → suatu kondisi jika suatu proses menggunakan suatu resource, maka tidak ada proses lain yang boleh menggunakan resource tersebut.
2. Kondisi genggam dan tunggu (hold and wait)
   → suatu kondisi saat suatu proses mengakses suatu resource, proses tersebut dapat meminta ijin untuk mengakses resource lain.
3. Kondisi non-preemption (non-preemption condition)
   → suatu kondisi 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)
   → suatu kondisi jika proses P_1 sedang mengakses resource R_1, dan meminta ijin untuk mengakses resource R_2, dan pada saat bersamaan proses P_2 sedang mengakses resource R_2 dan minta ijin untuk mengakses resource R_1.

* Penanganan Deadlock

1. Mengabaikan permasalahan (The Ostrich Algorithm)
   → The ostrich algorithm adalah strategi mengabaikan masalah yang mungkin terjadi akas dasar bahwa masalah itu mungkin sangat jarang terjadi. Dengan mengasumsikan bahwa lebih efektif untuk memungkinkan masalah itu terjadi dibandingkan upaya pencegahannya. Strategi ini bisa dilakukan dalam menangani deadlock jika deadlock diyakini sangat jarang terjadi, dan jika biaya untuk mendeteksi atau pencegahan lebih tinggi.
2. Deteksi dan pemulihan (recovery)
   → dilakukan dengan mengijinkan sistem mengalami deadlock, namun kemudian harus segera dapat memperbaikinya.
3. Pencegahan, dengan meniadakan salah satu dari empat kondisi deadlock
   → ⇒ Mencegah mutual exclusion
   Mutual exclusion tidak dapat dihindari karena tidak ada sumber daya yang dapat digunakan bersama-sama, jadi sistem harus membawa sumber daya yang tidak dapat digunakan bersama-sama.
   ⇒ Mencegah hold and wait
   Sistem harus menjamin apabila suatu proses meminta sumber daya, maka proses tersebut tidak sedang memegang sumber daya yang lain.
   ⇒ Mencegah non preemption
   • jika suatu proses yang membawa beberapa sumber daya meminta sumber daya lain yang tidak dapat segera dipenuhi untuk dialokasikan pada proses tersebut, maka semua

No.

Date

sumber daya yang sedang dibawah proses tersebut harus dibebaskan.

- proses [redacted] yang sedang dalam keadaan menunggu, sumber daya yang dibawahnya ditunda dan ditambahkan pada daftar sumber daya.
- proses akan di ~~[redacted]~~ restart hanya jika dapat memperoleh sumber daya yang lama dan sumber daya yang baru diminta.

→ Mencegah kondisi menunggu sirkular

proses dapat meminta proses kapanpun, tetapi permintaan harus dibuat terurut secara numerik agar tidak menimbulkan siklus.

4. Pengalokasian sumber daya yang efisien

→ situasi ketika sumberdaya dialokasikan pada penggunaan nilai tertinggi, tidak ada alternatif untuk menggunakan sumber daya lebih lanjut tanpa membuat yang lain lebih buruk.