Tugas Diskusi Kelompok Sistem Operasi



Disusun Oleh :

Risyad Pangestu 140810170003

Fahmi Auliya 140810170041

Rividya Permata Aluna 140810170047

Cryssa Aprilia Ermiyanda P. 140810170063

FAKULTAS MATEMATIKA DAN ILMU ALAM

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS PADJADJARAN

2018

1. Hal yang menyebabkan proses harus dihentikan.

Yaitu tindakan yang dilakukan sistem operasi pada saat sebuah proses telah selesai dilakukan atau dihentikan, sistem operasi menghapus struktur data yang digunakan untuk mengelola proses dan mengalokasikan ruang alamat ke proses itu. Secara umum, Destroyed Process (pemberhentian proses) dilakukan karena alasan-alasan sebagai berikut.

* Selesai secara normal. Proses mengeksekusi suatu panggilan layanan OS untuk mengindikasikan bahwa proses tersebut telah selesai beroperasi
* Batas waktu telah dilampaui. Proses telah berjalan lebih lama dari jumlah batas waktu yang telah ditentukan. Terdapat beberapa kemungkinan jenis waktu yang diukur. Jenis-jenis tersebut meliputi jumlah waktu yang berjalan (waktu jam dinding), jumlah waktu yang dibutuhkan untuk melakukan eksekusi, dan pada proses yang interaktif, jumlah waktu sejak pengguna terakhir diberi input
* Memori tidak tersedia. Proses memerlukan memori lebih banyak dari yang dapat disediakan oleh sistem
* Pelanggaran batas. Proses mencoba menakses lokasi memori yang tidak diijinkan untuk diases
* Error proteksi. Proses mencoba menggunakan  sumber daya atau file yang tidak diizinkan untuk digunakan, atau mencoba menggunakannya dengan cara yang salah, seperti melakukan penulisan ke file yang hanya dapat dibaca (read-only file)
* Error aritmetik. Proses mencoba untuk melakukan komputasi yang tidak diizinkan, seperti pembagian dengan nol, atau mencoba menyimpan bilangan yang besarnya diluar kemampuan perangkat keras. Kelebihan waktu proses telah menunggu event tertentu lebih lama dari waktu maksimum
* Kegagalan I/O. Error yang terjadi pada saat input atau output, seperti ketidakmampuan untuk menemukan file, kegagalan membaca atau menulis setelah beberapa kali mencoba (pada saat misalnya menemukan daerah yang telah rusak pada pita magnetik) atau operasi yang invalid (misalnya melakukan pembacaan dari printer)
* Intruksi yang invalid. Proses mencoba mengeksekusi intruksi yang tidak ada (seringkali akan mengakibatkan pencabangan ke suatu daerah data dan berusaha mengeksekusi data)
* Intruksi ber-privilage. Proses berusaha untuk menggunakan suatu intruksi yang dicadangkan bagi sistem operasi
* Pemakaian data yang salah. Potongan datanya memiliki jenis yang salah, atau tidak diinisilaisasi terlebih dahulu
* Intervensi Operator atau OS. Dengan sesuatu alasan, operator atau sistem operasi telah menghentikan proses (misalnya, telah terjadi deadlock)
* Penghentian Induk. Apabila induk berhenti, sistem operasi secara otomatis dapat menghentikan seluruh turunan  proses induk itu
* Permintaan induk. Proses induk umumnya memiliki otoritas untuk menghentikan semua turunannya

2. Thread

Thread adalah unit terkecil dalam suatu proses yang bisa dijadwalkan oleh sistem operasi. Merupakan sebuah status eksekusi (ready, running, suspend, block, queue, dll) Kadang disebut sebagai proses ringan (lightweight). Unit dasar dari dari sistem utilisasi pada processor (CPU). Dalam thread terdapat: ID Thread, Program Counter, Register dan Stack. Sebuah thread berbagi code section, data section dan resource sistem operasi dengan thread yang lain yang memiliki proses yang sama.

3. Komunikasi Antar Proses

Proses-proses yang ingin dikomunikasikan harus memiliki sebuah cara untuk memilih satu dengan yang lain. Mereka dapat menggunakan komunikasi langsung/ tidak langsung.

a. Komunikasi langsung

Setiap proses yang ingin berkomunikasi harus memiliki nama yang bersifat eksplisit baik penerimaan atau pengirim dari komunikasi tersebut. Dalam konteks ini, pengiriman dan penerimaan pesan secara primitive dapat dijabarkan sebagai:

* Send (P, message) – mengirim sebuah pesan ke proses P.
* Receive (Q, message) – menerima sebuah pesan dari proses Q.

Pembahasan ini memperlihatkan sebuah cara simetris dalam pemberian alamat. Oleh karena itu, baik keduanya yaitu pengirim dan penerima proses harus memberi nama bagi yang lain untuk berkomunikasi, hanya pengirim yang memberikan nama bagi penerima sedangkan penerima tidak menyediakan nama bagi pengirim. Dalam konteks ini, pengirim dan penerima secara sederhana dapat dijabarkan sebagai:

* Send (P, message) – mengirim sebuah pesan kepada proses P.
* Receive (id, message) – menerima sebuah pesan dari semua proses. Variabel id diatur sebagai nama dari proses dengan komunikasi.

b. Komunikasi tidak langsung

Dengan komunikasi tidak langsung, pesan akan dikirimkan pada dan diterima dari/ melalui mailbox (kotak surat) atau terminal-terminal, sebuah mailbox dapat dilihat secara abstrak sebagai sebuah objek didalam setiap pesan yang dapat ditempatkan dari proses dan dari setiap pesan yang bias dipindahkan. Setiap kotak surat memiliki sebuah identifikasi (identitas) yang unik, sebuah proses dapat berkomunikasi dengan beberapa proses lain melalui sebuah nomor dari mailbox yang berbeda.  
Dua proses dapat saling berkomunikasi apabila kedua proses tersebut sharing mailbox.  
Pengirim dan penerima dapat dijabarkan sebagai:

* Send (A, message) – mengirim pesan ke mailbox A.
* Receive (A, message) – menerima pesan dari mailbox A.

4. Buffering

Baik komunikasi itu langsung atau tak langsung, penukaran pesan oleh proses memerlukan antrian sementara. Pada dasarnya, terdapat tiga jalan dimana antrian tersebut diimplementasikan:

* Kapasitas nol: antrian mempunyai panjang maksimum 0, maka link tidak dapat mempunyai penungguan pesan (message waiting). Dalam kasus ini, pengirim harus memblok sampai penerima menerima pesan.
* Kapasitas terbatas: antrian mempunyai panjang yang telah ditentukan, paling banyak n pesan dapat dimasukkan. Jika antrian tidak penuh ketika pesan dikirimkan, pesan yang baru akan menimpa, dan pengirim pengirim dapat melanjutkan eksekusi tanpa menunggu. Link mempunyai kapasitas terbatas. Jika link penuh, pengirim harus memblok sampai terdapat ruang pada antrian.
* Kapasitas tak terbatas: antrian mempunyai panjang yang tak terhingga, maka, semua pesan dapat menunggu disini. Pengirim tidak akan pernah di blok.