Makalah Sistem Operasi

Monitor

Disusun oleh :

Faturrahman	**5208100001**
Aris K	**5208100030**
Ach. Pramono	**5208100158**
Fais Nasrulloh	**5208100704**
Arief Anwar Shodiq	**5208100117**

Semester Ganjil 2008/2009

Jurusan Sistem Informasi

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya

KATA PENGANTAR

Segala puji kami panjatkan kepada Tuhan YME yang telah memberikan rahmat-Nya sehingga kami dapat menyelesaikan makalah ini.

Ucapan terima kasih kami sampaikan kepada semua pihak yang telah membantu dan memberi dorongan sehingga kami dapat menyelesaikan makalah ini.

Dalam pembuatan makalah ini tentunya sangat banyak sekali kekurangan. Untuk itu saran yang konstruktif selalu menjadi harapan kami, tidak lain untuk perubahan yang lebih baik dan demi kelancaran bersama.

Surabaya, 12 April 2009

Penyusun

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sejauh ini, kita telah mengenal semafor sebagai perangkat keras sinkronisasi yang handal, namun kita masih membutuhkan bahasa pemrograman untuk melakukan sinkronisasi. Alasannya adalah karena ternyata implementasi semafor masih memiliki beberapa kelemahan. Kelemahan yang pertama yaitu kenyataan bahwa semafor memerlukan implementasi di tingkat rendah, hal ini merupakan hal yang kompleks untuk dilakukan bagi kebanyakan orang. Bahasa pemrograman lebih mudah untuk dipelajari dan diimplementasikan.Kelemahan yang kedua dari semafor adalah kode semafor terdistribusi dalam seluruh program sehingga menyulitkan pemeliharaan. Sehingga jelas bahwa kita memerlukan konstruksi tingkat tinggi yang dapat mengatasi atau paling tidak mengurangi kelemahan – kelemahan dalam semafor. Salah satu hal yang dapat kita lakukan adalah dengan menggunakan metode monitor.

1.2 Rumusan Masalah

Dalam makalah ini disampaikan berbagai penjelasan mengenai monitor, antara lain :

a. Penjelasan tema

b. Flowchart penyelesaian masalah

c. Algoritma penyelesaian masalah

d. Contoh kasus yang dapat diselesaikan

BAB I I

PEMBAHASAN

2.1 Penjelasan Tentang Tema

Terdapat beberapa pengertian dari monitor , yaitu :

a) Konsep monitor diperkenalkan pertama kali oleh Hoare (1974) dan Brinch Hansen (1975) untuk mengatasi beberapa masalah yang timbul ketika memakai semafor. Monitor merupakan kumpulan dari prosedur, variabel, dan struktur data dalam satu modul. Dengan mempergunakan monitor, sebuah proses dapat memanggil prosedur di dalam monitor, tetapi tidak dapat mengakses struktur data (termasuk variabel- variabel) internal dalam monitor. Dengan karakteristik demikian, monitor dapat mengatasi manipulasi yang tidak sah terhadap variabel yang diakses bersama-sama karena variable lokal hanya dapat diakses oleh prosedur lokal. Monitor tetap mempertahankan kondisi mutual eksklusifnya dengan membatasi hanya satu proses yang dapat aktif dalam monitor dalam suatu waktu. Yang mengatur kondisi tersebut adalah kompiler dan bukan programmer. Ternyata, mempertahakankan kondisi mutual eksklusif saja tidaklah cukup. Diperlukan juga suatu cara untuk memblok suatu proses jika ia tidak dapat terus berjalan. Misalnya dalam masalah ***bounded buffer***, proses harus diblok ketika buffer telah penuh. Oleh karena itu, monitor harus diperlengkapi dengan mekanisme sinkronisasi yang lain. Salah satu mekanisme yang dipergunakan adalah variabel kondisi yang terdapat dalam monitor dan hanya dapat diakses oleh monitor.

b) Monitor adalah suatu tipe data abstrak yang dapat mengatur aktivitas serta penggunaan *resource* oleh beberapa *thread*. Ide monitor pertama kali diperkenalkan oleh C.A.R Hoare dan PerBrinch-Hansen pada awal 1970-an. Monitor terdiri atas data-data *private* dengan fungsi-fungsi *public* yang dapat mengakses data-data tersebut. *Method-method* dalam suatu monitor sudah dirancang sedemikian rupa agar hanya ada satu buah *method* yang dapat bekerja pada suatu saat. Hal ini bertujuan untuk menjaga agar semua operasi dalam monitor bersifat *mutual exclusion*.

c) Solusi sinkronisasi ini dikemukakan oleh Hoare pada tahun 1974. Monitor adalah kumpulan prosedur, variabel dan struktur data di satu modul atau paket khusus. Proses dapat memanggil prosedur-prosedur kapan pun diinginkan. Tapi proses tak dapat mengakses struktur data internal dalam monitor secara langsung. Hanya lewat prosedur-prosedur yang dideklarasikan minitor untuk mengakses struktur internal.

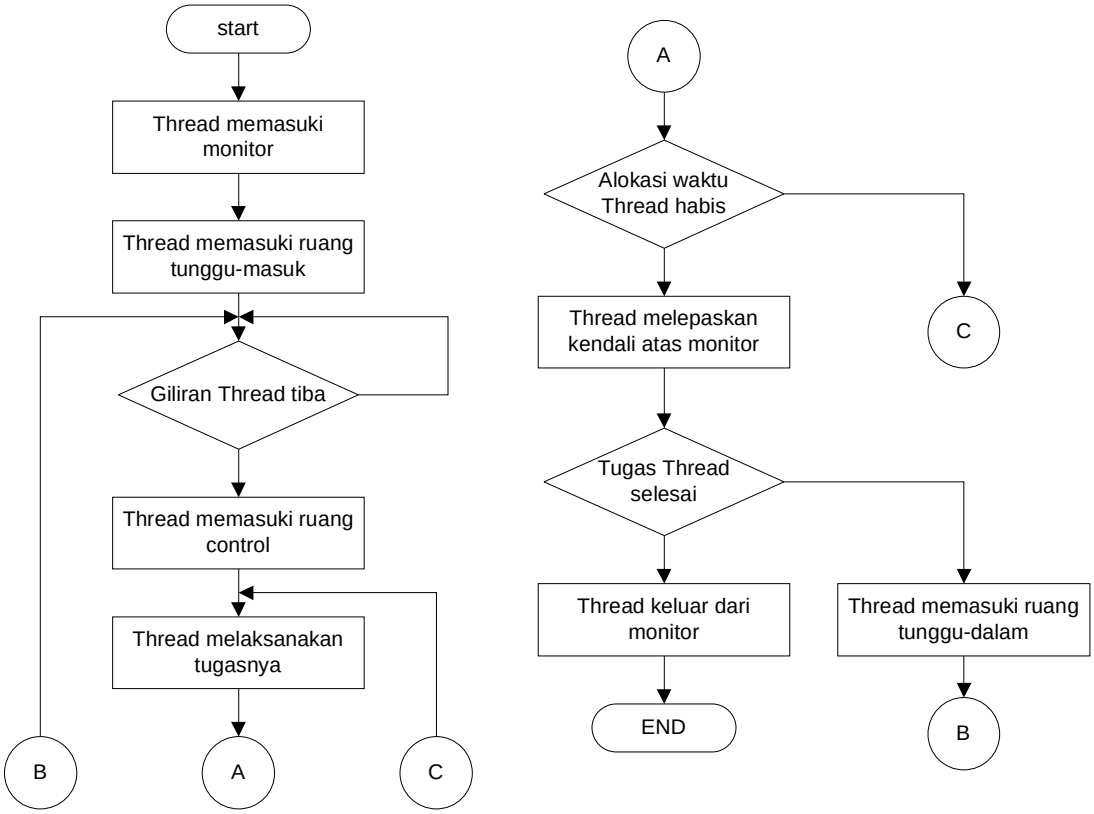
Properti-properti monitor adalah sebagai berikut:

- Variabel-variabel data lokal, hanya dapat diakses oleh prosedur-prosedur dala monitor dan tidak oleh prosedur di luar monitor.
- Hanya satu proses yang dapat aktif di monitor pada satu saat. Kompilator harus mengimplementasi ini(mutual exclusion).
- Terdapat cara agar proses yang tidak dapat berlangsung di-blocked. Menambahkan variabel-variabel kondisi, dengan dua operasi, yaitu Wait dan Signal.
- Wait: Ketika prosedur monitor tidak dapat berkanjut (misal producer menemui buffer penuh) menyebabkan proses pemanggil diblocked dan mengizinkan proses lain masuk monitor.
- Signal: Proses membangunkan partner-nya yang sedang diblocked dengan signal pada variabel kondisi yang sedang ditunggu partnernya.
- Versi Hoare: Setelah signal, membangunkan proses baru agar berjalan dan menunda proses lain.
- Versi Brinch Hansen: Setelah melakukan signal, proses segera keluar dari monitor.

Dengan memaksakan disiplin hanya satu proses pada satu saat yang berjalan pada monitor, monitor menyediakan fasilitas mutual exclusion. Variabel-variabel data dalam monitor hanya dapat diakses oleh satu proses pada satu saat. Struktur data bersama dapat dilindungi dengan menempatkannya dalam monitor. Jika data pada monitor merepresentasikan sumber daya, maka monitor menyediakan fasilitas mutual exclusion dalam mengakses sumber daya itu

d) Monitor dapat diibaratkan sebagai sebuah sekretariat dalam sebuah fakultas, dengan sekretariat sebagai suatu monitor dan mahasiswa,dosen sebagai proses, dan informasi akademik (jadwal kuliah,nilai,jadwal dosen, dll) sebagai variabel. Jika seorang mahasiswa akan mengambil transkrip nilainya dia akan meminta kepada petugas sekretariat daripada mengambilnya dan mencarinya sendiri.Sehingga hal ini akan meminimalkan kerusakan yang ditimbulkan dengan mengambilnya secara langsung.

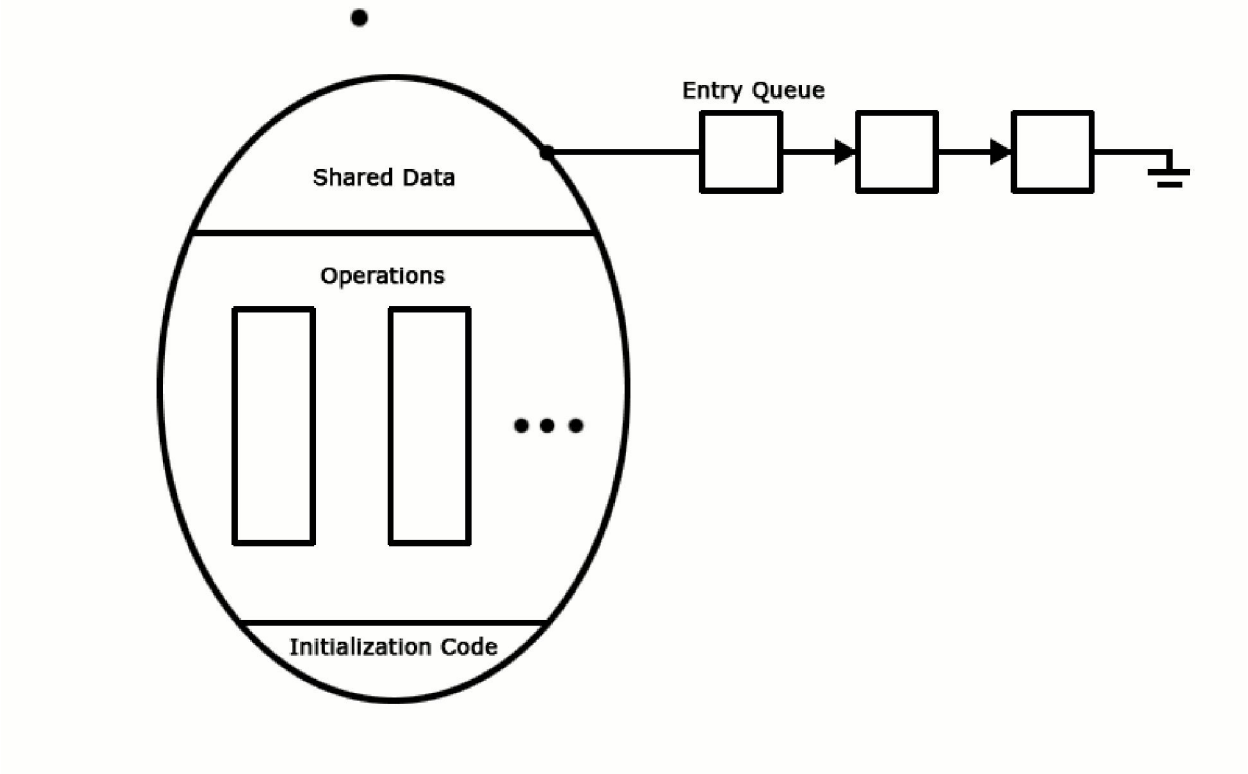
2.2 Flowchart Penyelesaian Masalah



2.3 Algoritma Penyelesaian Masalah

Monitor dapat dianalogikan sebagai sebuah bangunan dengan tiga buah ruangan yaitu satu buah ruangan kontrol, satu buah ruang-tunggu-masuk, satu buah ruang-tunggu-dalam. Ketika suatu *thread* memasuki monitor, ia memasuki ruang-tunggu-masuk (*enter*). Ketika gilirannya tiba, *thread* memasuki ruang kontrol (*acquire*), di sini *thread* menyelesaikan tugasnya dengan *shared resource* yang berada di ruang kontrol (*owning*). Jika tugas *thread* tersebut belum selesai tetapi alokasi waktu untuknya sudah habis atau *thread* tersebut menunggu pekerjaan *thread* lain selesai, *thread* melepaskan kendali atas monitor (*release*) dan dipindahkan ke ruang-tunggu-dalam (*waiting queue*). Ketika gilirannya tiba kembali, *thread* memasuki ruang kontrol lagi (*acquire*). Jika tugasnya selesai, ia keluar dari monitor (*release and exit*).

Gambar 20.1. Monitor



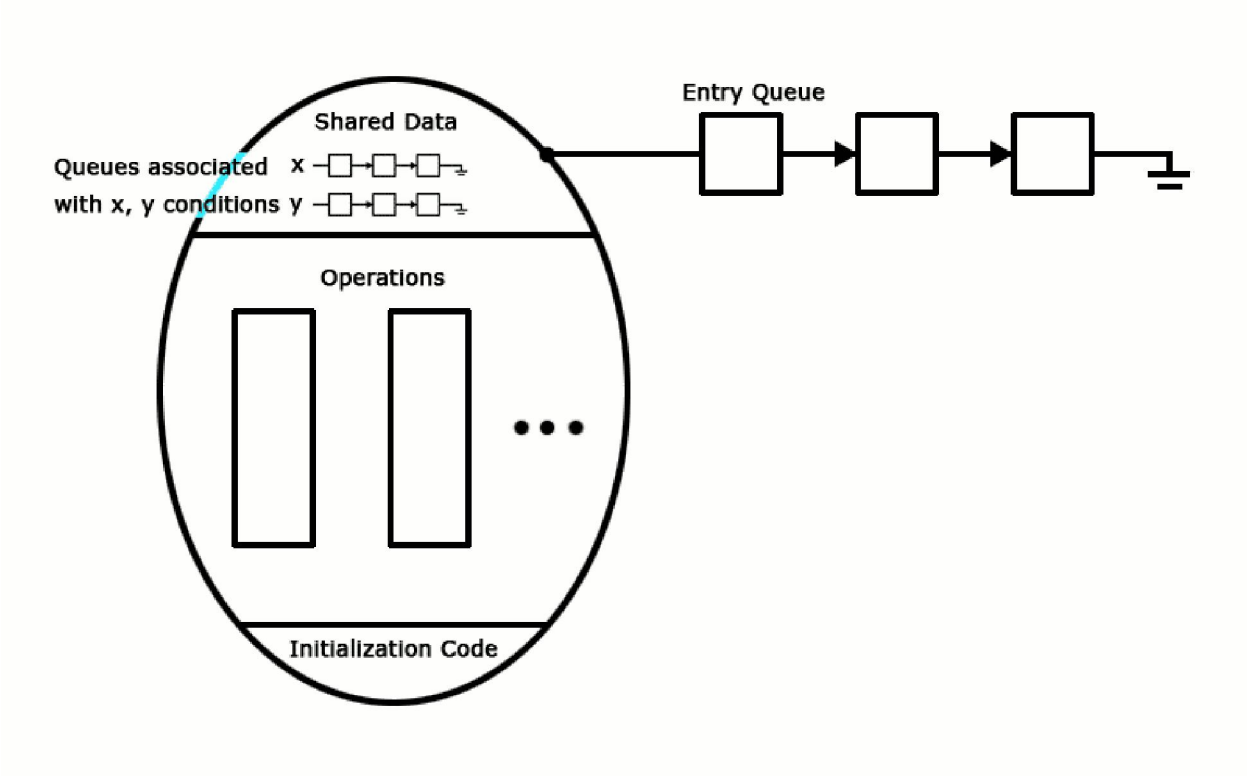
Bentuk dasar monitor adalah sebagai berikut :

```
monitor example
    i: intenger;
    c: condition;

    Procedure producer(x);
    .
    .
    End;

    Procedure consumer(x);
    .
    .
    End;
End monitor;
```

Gambar 20.2. Monitor dengan condition variable



Bayangkan jika pada suatu saat sebuah *thread* A memanggil fungsi signal pada *condition* x (x.signal()) dan ada sebuah *thread* B yang sedang menunggu operasi tersebut (B telahmemanggil fungsi x.wait() sebelumnya), ada dua kemungkinan keadaan *thread* A dan B setelah A mengeksekusi x.signal():

a. Signal-and-Wait, A menunggu sampai B keluar dari monitor atau menunggu *condition* lain yang dapat mengaktifkannya.
b. Signal-and-Continue, B menunggu sampai A keluar dari monitor atau menunggu *condition* lain yang dapat mengakifkannya.

2.3 Contoh Kasus yang dapat Diselesaikan :

Penyelesaian producer – consumer menggunakan monitor adalah sebagai berikut :

```
Monitor ProducerConsumer;
    condition full, empty;
    Integer count;

    Procedure Enter_Item;
    Begin
        if count = N then wait (full);
        enter_item;
        count:= count+1;
        if count=1 then signal (empty);
    End;

    Procedure Remove_Item;
    Begin
        if count = 0 then wait (empty);
        remover_intem;
        count:= count - 1;
        if count = N-1 then signal (full);
    End;
    count:= 0;
End;{Monitor}

Procedure Produce_Item;
{Menghasilkan item baru}

Procedure Consume_Item;
{Mengkonsumsi item}
```

```
Procedure producer
    Repeat
        Produce_Item;
        ProducerConsumer.enter;
    Forever
End;

Procedure consumer
    Repeat
        ProducerConsumer.remove;
        Consume_Item;
    Forever
End

Begin
    Parbegin
        Producer;
        Consumer;
    Parend
End.
```

BAB III

PENUTUP

Keunggulan monitor atas semafor adalah semua fungsi singkronisasi dilakukan terpusat di monitor. Dengan cara ini, lebih mudah memverifikasi kebenaran sinkronisasi dan mendeteksi kesalahan – kesalahan. Begitu monitor deprogram dengan benar, pengaksesan sumber daya yang dilindungi akan benar untuk seluruh pengaksesan yang dilakukan proses – proses. Pada semafor, pengaksesan sumber daya benar jika dan hanya jika semua proses yang mengakses sumber daya itu diprogram secara benar.

Metode hanya menyelesaikan mutual exclusion pada satu pemroses atau lebih yang mempunyai memori dipakai bersama (tightly coupled multiprocessors).

Metode tidak dapat diterapkan pada system terdistribusi berisi banyak pemroses dan memiliki memori sendiri (loosely coupled multyprocessors). Metode monitor dan semafor tidak dapat diterapkan karena tidak terdapat pemakaian memory bersama peletakan semafor atau monitor.

Untuk itu harus dicari metode komunikasi antar proses untuk mengatasi masalah – masalah kongkurensi (mutual exclusion, sinkronisasi, deadlock, dan startvation) pada system terdistribusi.

DAFTAR PUSTAKA

Bambang Hariyanto, Ir. *"Sistem Operasi"* CV Informatika, Bandung, 1997

Rahmat M. Samik – Ibrahim, *"Pengantar Sistem Operasi Komputer"*, Ardi Publishing, Yogyakarta,2004