Pertemuan 5

KONKURENSI
DAN
KEAMANAN SISTEM

A. Pengertian Konkurensi

Konkurensi merupakan landasan umum perancangan sistem operasi. Proses-proses disebut konkuren jika proses-proses (lebih dari satu proses) ada pada saat yang sama.

Proses-proses konkuren dapat sepenuhnya tidak bergantung dengan lainnya tapi dapat juga saling berinteraksi.

Proses-proses yang berinteraksi memerlukan sinkronisasi agar terkendali dengan baik.

B. Prinsip-prinsip Konkurensi

- Prinsip-prinsip konkurensi meliputi :
  1. Alokasi layanan pemroses untuk proses-proses
  2. Pemakaian bersama dan persaingan untuk mendapatkan sumberdaya
  3. Komunikasi antar proses
  4. Sinkronisasi aktivitas banyak proses

- Konkurensi dapat muncul pada konteks berbeda, yaitu :
  1. untuk banyak pemakai
  2. untuk strukturisasi dari aplikasi
  3. untuk strukturisasi dari satu proses
  4. untuk strukturisasi sistem operasi

Konteks Konkurensi

Konteks Konkurensi untuk Strukturisasi Satu Proses

Untuk peningkatan kinerja, maka satu proses dapat memiliki banyak *thread* yang independen. *Thread-thread* tersebut harus dapat bekerja sama untuk mencapai tujuan proses.

Konteks Konkurensi untuk Banyak Aplikasi

Sistem *multiprogramming* memungkinkan banyak aplikasi/proses yang sekaligus dijalankan di satu pemroses.

Konteks Konkurensi (Lanjut…)

Konteks Konkurensi untuk Strukturisasi Aplikasi

Perluasan prinsip perancangan modular dan pemrograman terstruktur adalah suatu aplikasi dapat secara efektif diimplementasikan sebagai sekumpulan proses, maka masing-masing proses menyediakan satu layanan spesifikasi tertentu.

Konteks Konkurensi u/ Strukturisasi Sistem Operasi

Keunggulan strukturisasi dapat diterapkan ke pemrograman sistem. Beberapa sistem operasi yang dipasarkan dan yang sedang dalam reset telah diimplementasikan sebagai sekumpulan proses.

C. Beberapa kesulitan yang ditimbulkan konkurensi

1. Pemakaian bersama sumber daya global
2. Pengelolaan alokasi sumber daya agar optimal
3. Pencarian kesalahan program (*Debuging*)

Pemakaian Bersama Sumber Daya Global

Jika dua proses menggunakan variabel global yang sama serta keduanya membaca dan menulis variabel itu, maka urutan terjadinya pembacaan dan penulisan terhadap variabel bersama menjadi kritis.

Beberapa kesulitan yang ditimbulkan konkurensi (Lanjut..)

Pengelolaan Alokasi Sumber Daya agar Optimal

Jika proses A meminta suatu kanal masukan/ keluaran tertentu dan dipenuhi, permintaan tersebut dapat ditunda (*suspend*) sebelum menggunakan kanal tersebut. Jika sistem operasi mempunyai kebijaksanaan mengunci kanal dan mencegah proses-proses lain menggunakan kanal itu, maka tindakan ini jelas hanya menghasilkan inefisiensi sistem komputer.

Pencarian Kesalahan Pemrograman

Pencarian kesalahan pada pemrograman konkuren lebih sulit dibanding pencarian kesalahan pada program-program sekuen

Proses-proses konkuren

Proses-proses konkuren mengharuskan hal-hal berikut ditangani sistem operasi, yaitu:

1. Mengetahui proses-proses yang aktif
2. Alokasi dan dealokasi beragam sumber daya untuk tiap proses aktif
3. Proteksi data dan sumber daya fisik proses
4. Hasil-hasil proses harus independen

D. Masalah-masalah konkuren

Masalah-masalah konkuren diantaranya :

1. *Mutual exclusion*
2. *Deadlock*
3. *Starvation*

1. Mutual Exclusion

Merupakan persoalan untuk menjamin hanya satu proses yang mengakses sumber daya pada suatu interval waktu tertentu. Pentingnya *mutual exclusion* dapat dilihat pada ilustrasi eksekusi *daemon printer*.

Mutual Exclusion (Lanjut…)

Contoh Ilustrasi Eksekusi *Daemon Printer*

Daemon untuk *printer* adalah proses penjadwalan dan pengendalian untuk mencetak berkas-berkas di printer sehingga seolah-olah *printer* dapat digunakan secara simultan oleh proses-proses. *Daemon* untuk *printer* mempunyai ruang penyimpanan di *harddisk* (disebut direktori *spooler*) untuk menyimpan berkas-berkas yang akan di cetak. Terdapat variabel ***in*** yang menunjuk ***slot*** bebas diruang *harddisk* yang dipakai untuk menyimpan berkas yang hendak di cetak.

Mutual Exclusion (Lanjut…)

Kriteria Penyelesaian *Mutual Exclusion*

Kemampuan menjamin *mutual exclusion* harus memenuhi kriteria-kriteria berikut:

a. *Mutual exclusion* harus dijamin
b. Hanya satu proses pada satu saat yang diizinkan masuk *critical section*

critical section: suatu bagian yang berisi sejumlah variabel yang akan di-*share* (dipengaruhi atau mempengaruhi) proses yang lain.

Mutual Exclusion (Lanjut…)

c. Proses yang berada di *noncritical section*, dilarang mem-block proses-proses yang ingin masuk *critical section.*
d. Harus dijamin proses yang ingin masuk *critical section* tidak menunggu selama waktu yang tidak berhingga.
e. Ketika tidak ada proses di *critical section*, maka proses yang ingin masuk *critical section* harus diizinkan segera masuk tanpa waktu tunda.
f. Tidak ada asumsi mengenai kecepatan relatif proses atau jumlah proses yang ada.

2. Deadlock

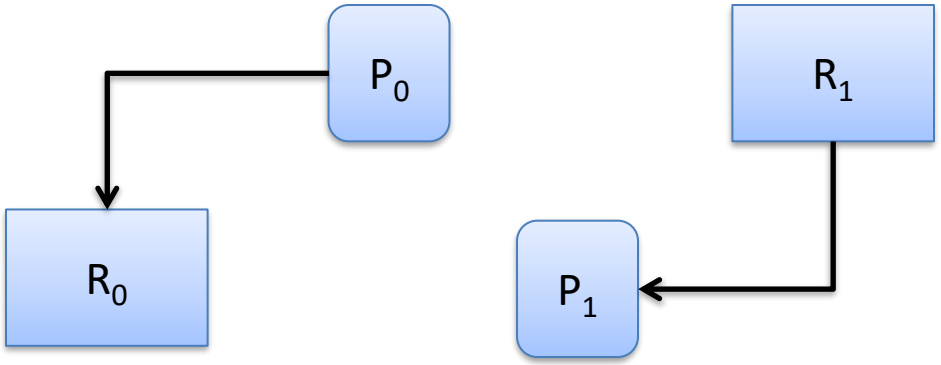
Deadlock terjadi ketika proses-proses mengakses sumber daya secara eksklusif. Semua *deadlock* yang terjadi melibatkan persaingan untuk memperoleh sumber data eksklusif oleh dua proses atau lebih.

Model *Deadlock*

Terjadi *deadlock* dapat digambarkan dengan menggunakan graph. Misal model *deadlock* dua proses dan dua sumber daya:

- Dua proses P_0 dan P_1
- Dua sumber daya R_0 dan R_1

Deadlock (Lanjut...)



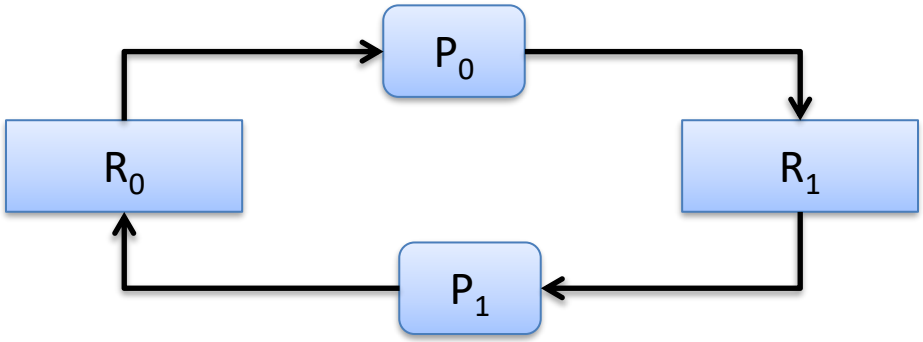
Gambar 7-1. Graph meminta sumber daya dan alokasi sumber daya

Keterangan:

(a) P_0 meminta sumber daya R_0, ditandai busur berarah dari proses P_0 ke sumber daya R_0.

(b) Sumber daya R_1 dialokasikan ke P_1, ditandai busur berarah dari sumber daya R_1 ke proses P_1.

Deadlock (Lanjut…)



Gambar 7-2. Graph *deadlock* dua proses dan dua sumber daya

Skenario yang Menimbulkan *Deadlock*:

- P_0 dialokasikan R_0, P_1 dialokasikan R_1. Kemudian P_0 sambil masih menggenggam R_0, meminta R_1 dan P_1 sambil masih menggenggam R_1, meminta R_0.
- Kejadian ini mengakibatkan *deadlock* karena sama-sama proses P_0 dan P_1 akan saling menunggu.
- Terjadinya *deadlock* ditandai munculnya graph melingkar

3. Startvation

Keadaan dimana pemberian akses bergantian terus-menerus, dan ada suatu proses yang tidak mendapatkan gilirannya.

Ilustasi *starvation*, misalnya :

- Terdapat tiga proses, yaitu P1, P2 dan P3.
- P1, P2 dan P3 memerlukan pengaksesan sumber daya R secara periodik.

Starvation (Lanjut…)

Skenario berikut terjadi:

- P1 sedang diberi sumber daya R sedangkan P2 dan P3 di*blocked* menunggu sumber daya R.
- Ketika P1 keluar dari *critical section*, maka P2 dan P3 diijinkan mengakses R.
- Asumsi P3 diberi hak akses, kemudian setelah selesai, hak akses kembali diberikan ke P1 yang saat itu kembali membutuhkan sumber daya R. Jika pemberian hak akses bergantian terus-menerus antara P1 dan P3, maka P2 tidak pernah memperoleh pengaksesan sumber daya R. Dalam kondisi ini memang tidak terjadi *deadlock*, hanya saja P2 mengalami *starvation* (tidak ada kesempatan untuk dilayani).

E. Pokok penyelesaian masalah konkurensi

Pada dasarnya penyelesaian masalah konkurensi terbagi menjadi dua, yaitu :

1. Mengasumsikan adanya memori yang digunakan bersama.
2. Tidak mengasumsikan adanya memori yang digunakan bersama

Adanya memori bersama lebih mempermudah dalam penyelesaian masalah konkurensi. Metode penyelesaian ini dapat dipakai untuk sistem singleprocessor ataupun multiprocessor yang mempunyai memori bersama

F. Definisi Keamanan

Keamanan sistem komputer adalah untuk menjamin sumber daya agar tidak digunakan atau dimodifikasi orang yang tidak diotorisasikan.

Keamanan Sistem terbagi menjadi tiga, yaitu:

1. Keamanan Eksternal (*external security*)
   Berkaitan dengan pengamanan fasilitas komputer dari penyusup dan bencana, seperti kebakaran dan kebanjiran.

Definisi Keamanan (Lanjut.....)

2. Keamanan Interface Pemakai (*user interface security*)
   Berkaitan dengan identifikasi pemakai sebelum pemakai diizinkan mengakses program dan data yang disimpan.

3. Keamanan Internal (*internal security*)
   Berkaitan dengan pengaman beragam kendali yang bangun pada perangkat keras dan sistem operasi yang menjamin operasi yang andal dan tak terkorupsi untuk menjaga integritas program dan data.

G. Masalah-masalah keamanan

Pada keamanan, terdapat dua masalah penting, yaitu:

1. Kehilangan data (*data lost*) disebabkan : bencana, kesalahan perangkat keras/lunak, kesalahan/ kelalaian manusia.
2. Penyusup (*intruder*), berupa penyusupan pasif dan penyusupan aktif

- **Penyusup pasif**
  Penyusup yang hanya membaca data yang tidak diotorisasikan
- **Penyusup Aktif**
  Penyusup yang mengubah data yang tidak diotorisasikan

H. Ancaman-ancaman keamanan

Kebutuhan keamanan sistem komputer meliputi tiga aspek, yaitu:

1. Kerahasiaan (*secrecy*)

   Adalah keterjaminan bahwa informasi di sistem komputer hanya dapat diakses oleh pihak-pihak yang diotorisasi, sehingga jika dimodifikasi tetap terjaga konsistensi dan keutuhan datanya.

Ancaman-ancaman keamanan (Lanjut...)

2. Integritas (*Intergrity*)

   Adalah keterjaminan bahwa sumber daya sistem komputer hanya dapat diakses oleh pihak-pihak yang diotorisasi.

3. Ketersediaan (*Availability*)

   Adalah keterjaminan bahwa sumber daya sistem komputer tersedia bagi pihak-pihak yang diotorisasi saat diperlukan.

I. Tipe-tipe Ancaman Keamanan

1. Interupsi

   Sumber daya sistem komputer dihancurkan atau menjadi tidak tersedia atau tidak berguna. Merupakan ancaman terhadap ketersediaan.

   cth: pemotongan kabel komunikasi, penghancuran bagian perangkat keras, seperti *harddisk*

2. Intersepsi

   Pihak tidak diotorisasi dapat mengakses sumber daya. Merupakan ancaman terhadap kerahasiaan. Pihak tidak diotorisasi dapat berupa orang atau program komputer.

   cth: penyadapan untuk mengambil data rahasia, mengkopi *file* tanpa diotorisasi

Tipe-tipe Ancaman Keamanan (Lanjut…)

3. Modifikasi

   Pihak tidak diotorisasi tidak hanya mengakses tapi juga merusak sumber daya. Merupakan ancaman terhadap integritas.

   cth: mengubah nilai-nilai *file* data, mengubah program sehingga bertindak secara berbeda, memodifikasi pesan-pesan yang ditransminikan pada jaringan.

4. Fabrikasi

   Pihak tidak diotorisasi menyisipkan/memasukkan objek-objek palsu ke sistem. Merupakan ancaman terhadap integritas

   cth: memasukkan pesan-pesan palsu ke jaringan, penambahan *record* ke *file*

J. Mekanisme Proteksi

Pada sistem komputer banyak objek yang perlu diproteksi, yaitu:

- Objek perangkat keras, antara lain: pemroses, segmen memori, terminal, disk drive, printer, dll.
- Objek perangkat lunak, antara lain: proses, file, basis data, dll.

Mekanisme proteksi dikembangkan berdasarkan konsep *domain*. *Domain* adalah himpunan pasangan (objek, hak).

K. Program-program Jahat

Bowles [BOW-92] memberikan taksonomi ancaman perangkat lunak atau klasifikasi program jahat (*malicious program*). Ancaman-ancaman itu dapat menjadi dua kategori, yaitu:

1. Program-program yang memerlukan *host program*
   - *Trapdoor*
   - *Logic Bomb*
   - *Trojan horse*
   - *Virus*
2. Program-program yang tidak memerlukan *host program* (independen).
   - *Bacteria*
   - *Worm*

Program-program Jahat (Lanjut…)

Bacteria

Adalah program yang mengkonsumsi sumber daya sistem dengan replikasi dirinya sendiri.

Logic Bomb

Adalah *logic* yang ditempelkan pada program komputer agar memeriksa kumpulan kondisi di sistem.

Trapdoor

Adalah titik masuk rahasia yang tidak terdokumentasi di satu program untuk memberikan akses tanpa metode otentifikasi normal.

Program-program Jahat (Lanjut…)

Trojan Horse

Adalah rutin tak terdokumentasi rahasia yang ditempelkan dalam satu program pengguna. Program-program tersebut jika terinfeksi, pasti terdapat kode tersembunyi dan ketika dijalankan, akan melakukan suatu fungsi yang tidak diinginkan.

Worm

Adalah program yang dapat mereplikasi dirinya dan mengirim kopian-kopian dari komputer ke komputer lewat hubungan jaringan.

L. *Virus* dan Anti *Virus*

Virus

Adalah kode yang ditempelkan dalam satu program yang menyebabkan pengopian dirinya ke satu program lain atau lebih. *Virus* biasanya melakukan fungsi yang tidak diinginkan.

Virus mengalami siklus hidup empat fase (tahap), yaitu:

1. Fase tidur (*dormant phase*)
2. Fase propagasi (*propagation phase*)
3. Fase pemicuan (*triggering phase*)
4. Fase eksekusi (*execution phase*)

Virus (Lanjut…)

Klasifikasi tipe *virus* adalah sebagai berikut:

- *Parasitic virus*
- *Memory-resident virus*
- *Boot sector virus*
- *Stealth virus*
- *Polymorphic virus*

Parasitic virus

Merupakan *virus* tradisional dan bentuk *virus* yang paling sering. Tipe ini menggantungkan diri ke *file exe.* Ketika program yang terinfeksi di eksekusi *Virus* mereplikasi dengan mencari *file-file exe* lain untuk diinfeksi.

Virus (Lanjut…)

Memory-resident virus

Virus memuatkan diri ke memori utama sebagai bagian program yang menetap. *Virus* menginfeksi setiap program yang dieksekusi.

Boot sector virus

Virus menginfeksi *master boot record* atau *boot record* dan menyebar saat sistem di-*boot* dari *disk* yang berisi *virus.*

Virus (Lanjut…)

Stealth virus

Virus yang bentuknya telah dirancang agar dapat menyembunyikan diri dari deteksi perangkat lunak *anti-virus*.

Polymorphic virus

Virus bermutasi setiap kali melakukan infeksi. Deteksi dengan "penandaan" *virus* tersebut tidak dimungkinkan.

Anti Virus

Solusi ideal terhadap ancaman *virus* adalah pencegahan. Pendekatan yang dilakukan setelah pencegahan terhadap masuknya *virus*,yaitu: Deteksi, Identifikasi dan Penghilangan

Perkembangan anti virus dapat diperiodekan menjadi 4 (empat) generasi, yaitu:

1. Generasi pertama : sekedar scanner sederhana
2. Generasi kedua : scanner yang pintar (*heuristic scanner*)
3. Generasi ketiga : jebakan-jebakan aktivitas *activity trap*)
4. Generasi keempat : proteksi penuh (*full-feature protection*)