SISTEM OPERASI

Tugas Konkurensi

Oleh:

Calvin Calfi Montolalu (140810200053)

UNIVERSITAS PADJADJARAN

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

Program Studi S-1 Teknik Informatika

2021

A. Apa yang dimaksud dengan konkurensi?

Konkurensi adalah kondisi kejadian dimana banyak proses berjalan di waktu yang bersamaan. Pada proses konkuren yang berinteraksi mempunyai beberapa masalah yang harus diselesaikan. Dengan begitu konkurensi akan berkaitan dengan bagaimana membuat struktur dan cara kerja, agar permasalahan dapat teratasi. Dikatakan sebagai landasan umum perancangan sistem operasi karena dalam menciptakan suatu sistem operasi, sistem operasi tersebut umumnya harus bisa menjalankan beberapa proses (lebih dari satu proses) pada saat yang bersamaan. Konkurensi merupakan kegiatan yang berhubungan dengan:

- Alokasi waktu pemroses untuk proses-proses yang aktif
- Pemakaian bersama dan persaingan untuk mendapat sumber daya
- Komunikasi antar proses
- Sinkronisasi Aktivitas banyak proses

Konkurensi muncul pada 4 konteks yang berbeda, yakni:

- Banyak aplikasi
- Strukturisasi sebuah aplikasi yang terdiri atas kumpulan proses
- Strukturisasi sistem operasi
- Strukturisasi sebuah proses

Di Dalam konkurensi terdapat beberapa kesulitan seperti berikut:

- Aktivitas proses lain
- Cara sistem operasi menghandle interupsi
- Kebijakan penjadwalan yang dilakukan oleh sistem operasi
- Pemakaian bersama sumber daya global
- Pengelolaan alokasi sumber daya agar optimal
- Pencarian kesalahan pemrograman

B. Beberapa istilah pada konkurensi

a. Critical Section

Critical section adalah dengan mendesain sebuah protokol di mana proses-proses dapat menggunakannya secara bersama-sama. Setiap proses harus ‘meminta izin’ untuk memasuki critical section-nya. Bagian dari kode yang mengimplementasikan izin ini disebut entry section. Akhir dari critical section itu disebut exit section. Bagian kode selanjutnya disebut remainder section.

Struktur umum dari proses Pi yang memiliki segmen critical section adalah :

```
do {
entry section
critical section
exit section
remainder section
} while(1);
```

Solusi dari masalah critical section harus memenuhi tiga syarat berikut:

1. Mutual Exclusion

   Jika suatu proses sedang menjalankan critical section-nya, maka proses-proses lain tidak dapat menjalankan critical section mereka. Dengan kata lain, tidak ada dua proses yang berada di critical section pada saat yang bersamaan.

2. Terjadi kemajuan (progress)

   Jika tidak ada proses yang sedang menjalankan critical section-nya dan ada proses-proses lain yang ingin masuk ke critical section, maka hanya proses-proses yang yang sedang berada dalam entry section saja yang dapat berkompetisi untuk mengerjakan critical section.

3. Ada batas waktu tunggu (bounded waiting)

   Jika seandainya ada proses yang sedang menjalankan critical section, maka proses lain memiliki waktu tunggu yang ada batasnya untuk menjalankan critical section -nya, sehingga dapat

dipastikan bahwa proses tersebut dapat mengakses critical section-nya (tidak mengalami starvation: proses seolah-olah berhenti, menunggu request akses ke critical section diperbolehkan).

b. Deadlock

Deadlock ini merupakan salah satu permasalahan ketika pemaksaan mutual exclusion. Deadlock adalah banyak proses yang saling menunggu hasil dari proses yang lain untuk dapat melanjutkan atau menyelesaikan tugasnya. Pada umumnya deadlock terjadi karena proses mengalami starvation, yakni suatu job yang sedang dieksekusi dan tidak ada hentinya, tidak diketahui kapan berhentinya atau bahkan bisa dikatakan job yang antri mempunyai status mati. Kondisi ini merupakan kondisi terparah karena banyak proses dapat terlibat dan semuanya tidak dapat mengakhiri prosesnya secara benar.

Metode mengendalikan deadlock:

1. Menggunakan suatu protokol untuk meyakinkan bahwa sistem tidak akan pernah mengalami deadlock
2. Mengizinkan sistem mengalami deadlock, namun kemudian harus segera dapat memperbaikinya
3. Mengabaikan semua masalah dan menganggap deadlock tidak akan pernah terjadi lagi dalam sistem.

Pencegahan Deadlock:

1. Meniadakan Mutual Exclusion

   Harus tetap menjaga resource-resource yang bersifat non-shareable. Yaitu, proses menahan sebuah resource, proses lain yang meminta resource tsb harus menunggu sampai proses melepaskannya. Jika terjadi pada perangkat I/O dan berkas, maka sulit untuk menghindari mutual exclusion pada sumber daya non shareable.

2. Meniadakan Syarat Hold & Wait

   Apabila suatu proses minta ijin untuk mengakses suatu resource, maka proses tersebut tidak boleh membawa resource

yang lainnya. Sebleum proses meminta resource, maka harus melepas semua resource yang dibawa.

3. Meniadakan Non Preemption

   Jika suatu proses minta ijin mengakses resource, sementara resource tersebut tidak dapat dipenuhi secepatnya, maka proses tersebut harus membebaskan semua resourcenya terlebih dahulu.

4. Meniadakan Circular Wait

   Memberi nomor pada setiap resource yang ada, dan setiap resource hanya boleh mengakses resource secara berurutan.

c. Mutual Exclusion

Mutual exclusion adalah jaminan hanya satu proses yang mengakses sumber daya pada satu interval waktu tertentu. Sumber daya yang tidak dapat dipakai bersama pada saat bersamaan. Kondisi demikian disebut sumber daya kritis, dan bagian program yang menggunakan sumber daya kritis disebut critical region / section. Hanya satu program pada satu saat yang diijinkan masuk ke critical region.

Sistem operasi menyediakan layanan (system call) yang bertujuan untuk mencegah proses lain masuk ke critical section yang sedang digunakan proses tertentu. Pemrograman harus menspesifikasikan bagian-bagian critical section, sehingga sistem operasi akan menjaganya. Pemaksaan atau pelanggaran mutual exclusion menimbulkan antara lain :

- Deadlock
- Starvation

Mutual exclusion adalah sebuah jaminan bahwa sebuah sumber daya komputer (ex. Hard Disk) hanya dipakai 1 proses pada suatu waktu. Misal ada sebuah jaringan komputer pada sebuah Bank, dimana server untuk penyimpanan data terpusat.

- Seorang nasabah dapat melakukan penyimpanan maupun penarikan dari cabang mana saja dan kapan saja termasuk pengambilan lewat ATM.

- Sehingga jika ingin data tepat, jika ada salah satu nasabah melakukan transaksi, record nasabah bersangkutan harus dikunci supaya record tidak dapat diakses oleh orang lain.
- Setelah transaksi selesai, record tempat data nasabah disimpan, record baru dibuka kembali. Ini adalah salah satu contoh penjaminan bahwa sumber daya hanya bisa dipakai oleh satu proses.

d. Race Condition

Race Condition adalah situasi di mana beberapa proses mengakses dan memanipulasi data bersama pada saat bersamaan. Nilai akhir dari data bersama tersebut tergantung pada proses yang terakhir selesai. Untuk mencegah race condition, proses-proses yang berjalan bersamaan harus disinkronisasi. Dalam beberapa sistem operasi, proses-proses yang berjalan bersamaan mungkin untuk membagi beberapa penyimpanan umum, masing-masing dapat melakukan proses baca (read) dan proses tulis (write). Penyimpanan bersama (shared storage) mungkin berada di memori utama atau berupa sebuah berkas bersama, lokasi dari memori bersama tidak merubah kealamian dari komunikasi atau masalah yang muncul. Untuk mengetahui bagaimana komunikasi antar proses bekerja, mari kita simak sebuah contoh sederhana, sebuah print spooler. Ketika sebuah proses ingin mencetak sebuah berkas, proses tersebut memasukkan nama berkas ke dalam sebuah spooler direktori yang khusus. Proses yang lain, printer daemon, secara periodik memeriksa untuk mengetahui jika ada banyak berkas yang akan dicetak, dan jika ada berkas yang sudah dicetak dihilangkan nama berkasnya dari direktori.

e. Starvation

Starvation adalah keadaan dimana pemberian akses bergantian terus menerus, dan ada suatu proses yang tidak mendapatkan gilirannya. Juga dapat dimaksudkan bahwa kondisi bila beberapa proses-proses menunggu alokasi sumber daya sampai tak berhingga, sementara proses-proses lain dapat memperoleh alokasi sumber daya.

Hal ini disebabkan bias pada kebijaksanaan atau strategi alokasi sumber daya. Kondisi seperti ini harus dihindari pada sistem operasi karena tidak adil, tapi dikehendaki penghindaran dilakukan seefisien mungkin. Penanganan ini merupakan persoalan yang sulit untuk menemukan kriteria yang benar, adil dan efisien dalam suatu strategi Sistem Operasi.

C. Sebutkan beberapa cara untuk mencapai mutual exclusion dan jelaskan secara singkat

Fasilitas atau kemampuan menyediakan dukungan mutual exclusion harus memenuhi kriteria-kriteria berikut:

1. Mutual exclusion harus dijamin. Hanya satu proses pada saat diizinkan masuk critical section. Proses-proses lain dilarang masuk critical section yang sama pada saat telah terdapat satu proses masuk critical section itu.
2. Proses yang berada di non critical section dilarang memblokir proses-proses lain yang ingin masuk critical section
3. Harus dijamin proses yang ingin masuk critical section tidak menunggu selama waktu yang tak terhingga. Atau tidak boleh terdapat deadlock atau starvation
4. Ketika tidak ada proses pada critical section maka proses yang ingin masuk critical section harus diizinkan masuk tanpa waktu tunda.
5. Tidak ada asumsi mengenai kecepatan relatif proses atau jumlah proses yang ada.
6. Proses hanya tinggal pada critical section selama satu waktu yang berhingga.