

Laporan Praktikum

Sistem Operasi

Modul 5



Nama : Asep haryana saputra

NIM : 20230810043

Kelas : TINFC-2023-04

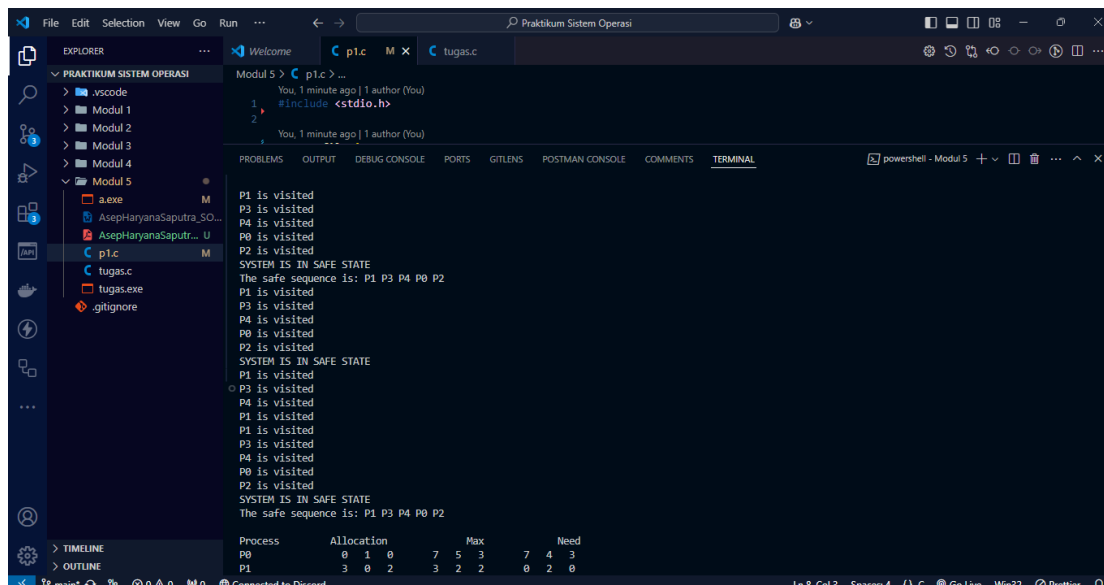
Teknik Informatika
Fakultas Ilmu Komputer
Universitas Kuningan

Pretest

1. Tujuan utama Algoritma Bankir adalah mencegah deadlock dengan memastikan sistem tetap berada dalam kondisi aman (*safe state*). Algoritma ini bekerja dengan memeriksa apakah permintaan sumber daya oleh suatu proses dapat dipenuhi tanpa menyebabkan potensi deadlock. Sebelum alokasi dilakukan, algoritma memprediksi dampak dari pemberian sumber daya terhadap ketersediaan dan keamanan sistem. Jika sistem tetap aman setelah permintaan dipenuhi, alokasi dilakukan; jika tidak, permintaan ditolak atau ditunda hingga sumber daya mencukupi. Dengan pendekatan ini, algoritma memastikan alokasi sumber daya dilakukan secara efisien tanpa mengorbankan kestabilan sistem.
2. Algoritma Bankir menentukan keadaan aman (*safe state*) dengan mensimulasikan alokasi sumber daya untuk memastikan semua proses dapat menyelesaikan eksekusinya secara berurutan tanpa deadlock. Algoritma memeriksa apakah sumber daya yang tersedia cukup untuk memenuhi kebutuhan proses tertentu, lalu secara hipotetis mengalokasikannya. Setelah proses selesai, sumber daya dikembalikan ke sistem untuk digunakan oleh proses lain. Jika ditemukan urutan di mana semua proses dapat selesai, sistem dianggap aman; jika tidak, sistem berada dalam kondisi tidak aman (*unsafe state*).
3. Dalam Algoritma Bankir, **Need** adalah jumlah sumber daya tambahan yang dibutuhkan oleh suatu proses untuk menyelesaikan eksekusinya. Nilai ini dihitung dengan mengurangi jumlah sumber daya yang telah dialokasikan dari jumlah maksimum yang diperlukan oleh proses tersebut.

Praktikum

1.



```
Modul 5 > C p1.c > ...
You, 1 minute ago | 1 author (You)
1 #include <stdio.h>
2
You, 1 minute ago | 1 author (You)

PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN CONSOLE COMMENTS TERMINAL
powershell - Modul 5 + v ... x

P1 is visited
P3 is visited
P4 is visited
P0 is visited
P2 is visited
SYSTEM IS IN SAFE STATE
The safe sequence is: P1 P3 P4 P0 P2
P1 is visited
P3 is visited
P4 is visited
P0 is visited
P2 is visited
SYSTEM IS IN SAFE STATE
P1 is visited
P3 is visited
P4 is visited
P0 is visited
P2 is visited
SYSTEM IS IN SAFE STATE
The safe sequence is: P1 P3 P4 P0 P2

Process Allocation Max Need
P0 0 1 0 7 5 3 7 4 3
P1 3 0 2 3 2 2 0 2 0
```

Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%205/p1.c>

PostTest

1

Tujuan utama penerapan Algoritma Bankir dalam program ini adalah memastikan sistem berada dalam keadaan aman (*safe state*) guna mencegah deadlock. Algoritma ini mengevaluasi kebutuhan sumber daya setiap proses dan memeriksa apakah sumber daya yang tersedia cukup untuk memenuhinya. Dengan demikian, algoritma memastikan semua proses dapat diselesaikan tanpa menyebabkan deadlock, menjaga urutan eksekusi tetap aman.

2

Untuk menentukan apakah sistem dalam keadaan aman (*safe state*), program mencari urutan proses yang dapat dieksekusi berdasarkan sumber daya yang tersedia. Jika semua proses dapat dijalankan tanpa ada yang terjebak menunggu, sistem dianggap aman. Sebaliknya, jika tidak ditemukan urutan tersebut, sistem berada dalam keadaan tidak aman, yang berpotensi menyebabkan deadlock.

3

Program menghitung kebutuhan (*need*) setiap proses dengan mengurangi jumlah sumber daya yang telah dialokasikan dari jumlah maksimum yang diperlukan. Sebagai contoh, jika sebuah proses membutuhkan 5 unit sumber daya dan saat ini telah dialokasikan 3 unit, maka sisa kebutuhannya adalah 2 unit. Rumusnya adalah **Need = Max - Allocation**.

4.

Jika terjadi deadlock, program akan menampilkan pesan bahwa permintaan sumber daya tidak dapat dipenuhi karena sistem berada dalam keadaan tidak aman. Deadlock terjadi ketika proses-proses dalam sistem saling menunggu sumber daya yang tidak tersedia, sehingga tidak ada proses yang dapat melanjutkan eksekusi.

Tugas

1

Algoritma Bankir lebih fleksibel karena memeriksa setiap permintaan sumber daya sebelum dialokasikan, memastikan sistem tetap dalam keadaan aman dan menghindari deadlock. Algoritma ini cocok untuk sistem yang dinamis dengan perubahan kebutuhan sumber daya. Sebaliknya, Grafik Alokasi lebih sederhana karena hanya mendeteksi deadlock jika ada siklus dalam grafik, tetapi kurang efektif untuk menangani sistem dengan banyak perubahan permintaan sumber daya.

2

Deadlock Avoidance mencegah deadlock dengan menganalisis setiap permintaan sumber daya secara hati-hati untuk memastikan tidak akan menimbulkan masalah di masa depan, menjaga sistem tetap aman. Sementara itu, Deadlock Prevention berfokus pada pengubahan aturan penggunaan sumber daya, seperti melarang kondisi yang dapat memicu deadlock, tetapi pendekatan ini sering kali mengurangi efisiensi sistem.

3.

A

Kesimpulan

Kesimpulan dari praktikum ini adalah bahwa penerapan Algoritma Bankir dalam sistem operasi sangat penting untuk mencegah terjadinya deadlock dan menjaga sistem tetap dalam kondisi aman (safe state). Algoritma ini bekerja dengan memprediksi dampak dari permintaan sumber daya sebelum alokasi dilakukan, memastikan bahwa setiap proses dapat diselesaikan tanpa terjebak dalam situasi deadlock. Melalui pendekatan ini, sistem dapat mengelola sumber daya secara efisien, terutama dalam situasi yang dinamis dengan perubahan kebutuhan sumber daya. Sementara itu, perbandingan antara Algoritma Bankir dan metode lain seperti Grafik Alokasi menunjukkan bahwa meskipun Grafik Alokasi lebih sederhana, namun Algoritma Bankir menawarkan fleksibilitas yang lebih besar dalam menangani berbagai jenis permintaan sumber daya. Dengan memeriksa kebutuhan sumber daya setiap proses dan memastikan ketersediaan sumber daya yang cukup, Algoritma Bankir berperan penting dalam menjaga stabilitas dan efisiensi sistem operasi.