



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : DEADLOCK
NAMA : ROCHMANU PURNOMOHADI ERFITRA
NIM : 225150200111018
TANGGAL : 17/04/2022
ASISTEN : DEVIN JAYA NUGRAHA
FATAH ABDI PRAKOSO

6.4. Langkah Praktikum

1. Contoh jika terjadi deadlock

```
Maximum for resource 1 : 7
Allocation for resource 2 : 1
Maximum for resource 2 : 5
Allocation for resource 3 : 0
Maximum for resource 3 : 3

Process 2
Allocation for resource 1 : 2
Maximum for resource 1 : 3
Allocation for resource 2 : 0
Maximum for resource 2 : 0
Allocation for resource 3 : 0
Maximum for resource 3 : 2

Process 3
Allocation for resource 1 : 3
Maximum for resource 1 : 9
Allocation for resource 2 : 0
Maximum for resource 2 : 0
Allocation for resource 3 : 2
Maximum for resource 3 : 2

Process 4
Allocation for resource 1 : 2
Maximum for resource 1 : 2
Allocation for resource 2 : 1
Maximum for resource 2 : 2
Allocation for resource 3 : 1
Maximum for resource 3 : 2

Process 5
Allocation for resource 1 : 0
Maximum for resource 1 : 4
Allocation for resource 2 : 0
Maximum for resource 2 : 3
Allocation for resource 3 : 2
```

2. Contoh jika tidak terjadi deadlock

Screenshot

6.5. Pembahasan

Setelah anda berhasil melakukan kompilasi program lakukan percobaan dengan langkah-langkah berikut:

1. Masukkan variable berikut :

- a. Jumlah proses = 5
- b. Jumlah resource = 3

Dengan Matriks



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

	Allocation			Max			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	5	3	3	3	2
P1	2	0	0	3	2	2			
P2	3	0	2	9	0	2			
P3	2	1	1	2	2	2			
P4	0	0	2	4	3	3			

Jawab:

```
Maximum for resource 1 : 7
Allocation for resource 2 : 1
Maximum for resource 2 : 5
Allocation for resource 3 : 0
Maximum for resource 3 : 3

Process 2
Allocation for resource 1 : 2
Maximum for resource 1 : 3
Allocation for resource 2 : 0
Maximum for resource 2 : 2
Allocation for resource 3 : 0
Maximum for resource 3 : 2

Process 3
Allocation for resource 1 : 3
Maximum for resource 1 : 9
Allocation for resource 2 : 0
Maximum for resource 2 : 0
Allocation for resource 3 : 2
Maximum for resource 3 : 2

Process 4
Allocation for resource 1 : 2
Maximum for resource 1 : 2
Allocation for resource 2 : 1
Maximum for resource 2 : 2
Allocation for resource 3 : 1
Maximum for resource 3 : 2

Process 5
Allocation for resource 1 : 0
Maximum for resource 1 : 4
Allocation for resource 2 : 0
Maximum for resource 2 : 3
Allocation for resource 3 : 2
```

2. Amati hasilnya dan tuliskan outputnya!

Jawab:



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```
Process 5
Allocation for resource 1 : 0
Maximum for resource 1 : 4
Allocation for resource 2 : 0
Maximum for resource 2 : 3
Allocation for resource 3 : 2
Maximum for resource 3 : 3

Available resources :
Resource 1 : 3
Resource 2 : 3
Resource 3 : 2

Allocation Matrix
0 1 0
2 0 0
3 0 2
0 1 1
0 0 2

Maximum Requirement Matrix
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3

Need Matrix
7 4 3
1 2 2
6 0 0
0 1 1
4 3 1

P1 P3 P4 P0 P2
A safety sequence has been detected.

Do you want make an additional request ? (1=Yes|0=No)
```

Proses di atas merupakan contoh dari algoritma Banker's Algorithm yang digunakan dalam manajemen sumber daya sistem operasi. Algoritma ini digunakan untuk menghindari deadlock dalam sistem dengan memastikan bahwa setiap proses hanya akan meminta sumber daya yang tersedia.

Pada tahap awal, matriks alokasi (allocation matrix), matriks maksimum permintaan (maximum requirement matrix), dan matriks kebutuhan (need matrix) ditentukan. Matriks alokasi menunjukkan sumber daya yang telah dialokasikan untuk setiap proses, matriks maksimum permintaan menunjukkan jumlah maksimum sumber daya yang dibutuhkan oleh setiap proses, dan matriks kebutuhan menunjukkan sisa jumlah sumber daya yang dibutuhkan oleh setiap proses.

3. Apakah permintaan penambahan resource dari kondisi pada no 1 diijinkan apabila P4 melakukan permintaan tambahan alokasi sebanyak R1=3, R2=3 dan R3=0

Jawab:



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

```
Allocation for resource 1 : 2 2 1 2 1 2
Maximum for resource 1 : Allocation for resource 2 : Maximum for resource 2 : Allocation
Process 5
Allocation for resource 1 : 0 4 0 3 2 3
Maximum for resource 1 : Allocation for resource 2 : Maximum for resource 2 : Allocation
Available resources :
Resource 1 : 3 3 2
Resource 2 : Resource 3 :
Allocation Matrix
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Maximum Requirement Matrix
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Need Matrix
7 4 3
1 2 2
6 0 0
0 1 1
4 3 1
P1 P3 P4 P0 P2
A safety sequence has been detected.
Do you want make an additional request ? (1=Yes|0=No)1
Enter process no. : 5
Enter additional request :-
Request for resource 1 : 3 3 0
Deadlock has occurred.
```

Tidak diizinkan karena akan terjadi deadlock pada process yang dijalankan karena resource yang ada tidak cukup untuk menyelesaikan seluruh program meskipun telas memakai algoritma Banker's Algorithm

4. Amati hasilnya dan tuliskan outputnya untuk kondisi dari point 3.

Jawab:

```
Allocation for resource 1 : 0 4 0 3 2 3
Maximum for resource 1 : Allocation for resource 2 : Maximum for resource 2 : Allocation
Available resources :
Resource 1 : 3 3 2
Resource 2 : Resource 3 :
Allocation Matrix
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Maximum Requirement Matrix
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Need Matrix
7 4 3
1 2 2
6 0 0
0 1 1
4 3 1
P1 P3 P4 P0 P2
A safety sequence has been detected.
Do you want make an additional request ? (1=Yes|0=No)1
Enter process no. : 5
Enter additional request :-
Request for resource 1 : 3 3 0
Request for resource 2 : Request for resource 3 :
Deadlock has occurred.
```



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

Karena setelah process 1 dan process 3 diselesaikan, available resource yang tersisa tidak cukup untuk menyelesaikan process selanjutnya sehingga deadlock terjadi setelah process 3 diselesaikan.

5. Apakah permintaan penambahan resource dari kondisi pada no 1 diijinkan apabila P4 melakukan permintaan tambahan alokasi sebanyak R1=0, R2=2 dan R3=0

Jawab:

```
Process 5
Allocation for resource 1 : 0 4 0 3 2 3
Maximum for resource 1 : Allocation for resource 2 : Maximum for resource 2 : Allocation
Available resources :
Resource 1 : 3 3 2
Resource 2 : Resource 3 :
Allocation Matrix
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Maximum Requirement Matrix
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Need Matrix
7 4 3
1 2 2
6 0 0
0 1 1
4 3 1
P1 P3 P4 P0 P2
A safety sequence has been detected.
Do you want make an additional request ? (1=Yes|0=No)1
Enter process no. : 5
Enter additional request :-
Request for resource 1 : 0 2 0
Request for resource 2 : Request for resource 3 :
```

Diizinkan

6. Amati hasilnya dan tuliskan outputnya untuk kondisi pada point 5.

Jawab:



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

```
Available resources :
Resource 1 : 3 3 2
Resource 2 : Resource 3 :
Allocation Matrix
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Maximum Requirement Matrix
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Need Matrix
7 4 3
1 2 0
6 0 0
0 1 1
4 3 1
P1 P2 P3 P4 P0 P2
A safety sequence has been detected.
Do you want make an additional request ? (1=Yes|0=No)1
Enter process no. : 5
Enter additional request :-
Request for resource 1 : 0 2 0
Request for resource 2 : Request for resource 3 :
P3 P4 P1 P2 P0
A safety sequence has been detected.
rchmn@rchmn-comp:~/Desktop/Kampus/Sistem Operasi/6_Deadlock$
```

Penambahan pada nomor 5 diizinkan karena process tidak menemui deadlock selama meeksekusi process dan program dapat menyelesaikan semua process dengan safety

7. Pada bagian program terdapat Resource Request Alghoritm (baris 14 sampai 40) dan Safety Alghoritm (baris 42- 72). Jelaskan konsep algoritmanya dan penggunaannya!

Jawab:



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

```
C:\deadlock > g++ safetyint.cpp [1]10, int [1]10, int, int, int [1]
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void print(int x[1]10, int n, int m)
5 {
6     int i, j;
7     for (i = 0; i < n; i++)
8     {
9         printf("\n");
10        for (j = 0; j < m; j++)
11        {
12            printf("%d\t", x[i][j]);
13        }
14    }
15 }
16
17 // Resource Request algorithm
18 void res_request(int A[10][10], int N[10][10], int AV[10][10], int pid, int m)
19 {
20     int reqmat[1][10];
21     int i;
22     printf("\nEnter additional request: ");
23     for (i = 0; i < m; i++)
24     {
25         printf("Request for resource %d : ", i + 1);
26         scanf("%d", &reqmat[0][i]);
27     }
28     for (i = 0; i < m; i++)
29     {
30         if (reqmat[0][i] > N[pid][i])
31         {
32             printf("\n Error encountered.\n");
33             exit(0);
34         }
35         for (i = 0; i < m; i++)
36         {
37             if (reqmat[0][i] > AV[0][i])
38             {
39                 printf("\n Resource unavailable.\n");
40                 exit(0);
41             }
42         }
43     }
44 }
```

```
C:\deadlock > g++ safetyint.cpp [1]10, int [1]10, int, int, int [1]
41 {
42     AV[0][i] = reqmat[0][i];
43     A[pid][i] = reqmat[0][i];
44     N[pid][i] = reqmat[0][i];
45 }
46 }
47
48 // Safety algorithm
49 int safety(int A[1]10, int N[1]10, int AV[1]10, int n, int m, int a[])
50 {
51     int i, j, k, x = 0;
52     int F[10], W[1]10;
53     int pflag = 0, flag = 0;
54     for (i = 0; i < n; i++)
55     {
56         F[i] = 0;
57         W[0][i] = AV[0][i];
58         for (k = 0; k < m; k++)
59         {
60             for (i = 0; i < n; i++)
61             {
62                 if (F[i] == 0)
63                 {
64                     flag = 0;
65                     for (j = 0; j < m; j++)
66                     {
67                         if (N[i][j] > W[0][j])
68                         {
69                             flag = 1;
70                         }
71                     }
72                     if (flag == 0 && F[i] == 0)
73                     {
74                         for (j = 0; j < m; j++)
75                         {
76                             W[0][j] = A[i][j];
77                             F[i] = 1;
78                             pflag++;
79                             a[x++] = i;
80                         }
81                     }
82                 }
83             }
84         }
85     }
86 }
```

Resource Request Algorithm pada baris 14 sampai 40 adalah algoritma yang digunakan untuk menangani permintaan alokasi sumber daya tambahan dari suatu proses. Pertama-tama, program akan meminta input jumlah alokasi sumber daya tambahan yang diminta oleh proses dengan menggunakan fungsi scanf. Setelah itu, program akan melakukan pengecekan apakah permintaan sumber daya tambahan yang diminta oleh proses tidak melebihi jumlah sumber daya yang tersedia dalam matriks N. Jika permintaan sumber daya tambahan melebihi sumber daya yang tersedia dalam matriks AV, maka program akan menampilkan pesan "Resources unavailable" dan keluar dari program. Jika



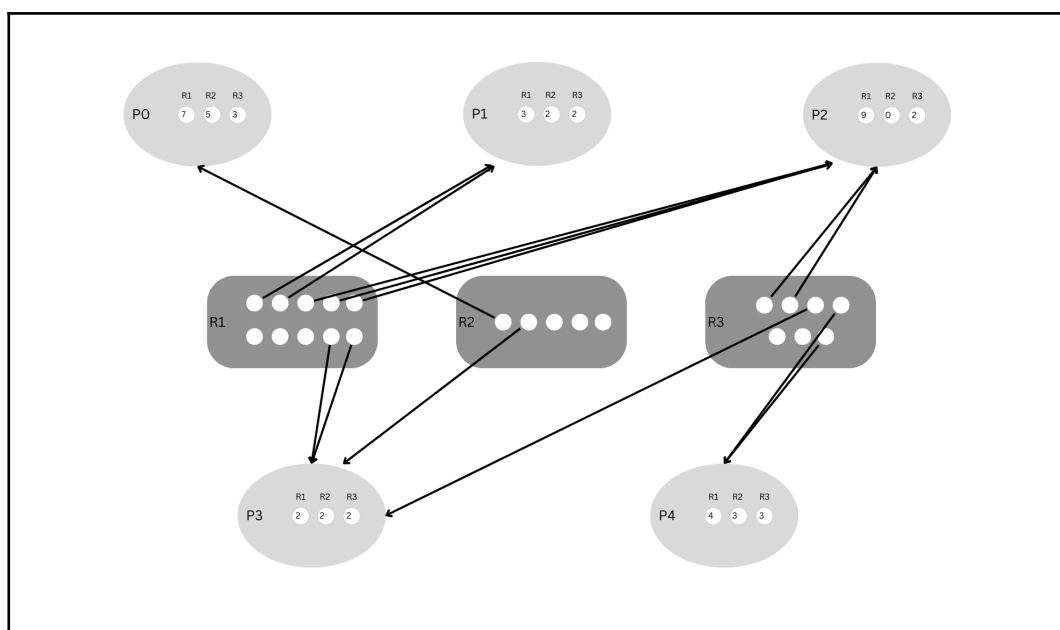
tidak, maka program akan mengalokasikan sumber daya tambahan ke dalam matriks A dan mengurangi jumlah sumber daya yang tersedia dalam matriks AV dan N

Safety Algorithm pada baris 42 sampai 72 adalah algoritma yang digunakan untuk menentukan apakah suatu sistem aman atau tidak aman. Pertama-tama, algoritma akan memeriksa apakah sumber daya yang tersedia (W) cukup untuk memenuhi permintaan sumber daya setiap proses yang belum selesai. Jika iya, maka algoritma akan menandai proses tersebut sebagai selesai dan menambahkan sumber daya yang digunakan oleh proses tersebut ke sumber daya yang tersedia. Selanjutnya, algoritma akan memeriksa lagi apakah sumber daya yang tersedia cukup untuk memenuhi permintaan sumber daya setiap proses yang belum selesai. Proses tersebut akan diulang hingga semua proses selesai atau tidak ada lagi proses yang dapat dijalankan.

Jika algoritma menemukan urutan proses yang dapat dilakukan sehingga tidak ada proses yang mengalami deadlock, maka algoritma akan mengeluarkan pesan bahwa sistem aman. Jika tidak, maka algoritma akan mengeluarkan pesan bahwa sistem tidak aman.

8. Dapatkan kondisi pada soal no 1, 3 dan 5 di gambarkan dengan menggunakan Resource Allocation Graph? Jelaskan!

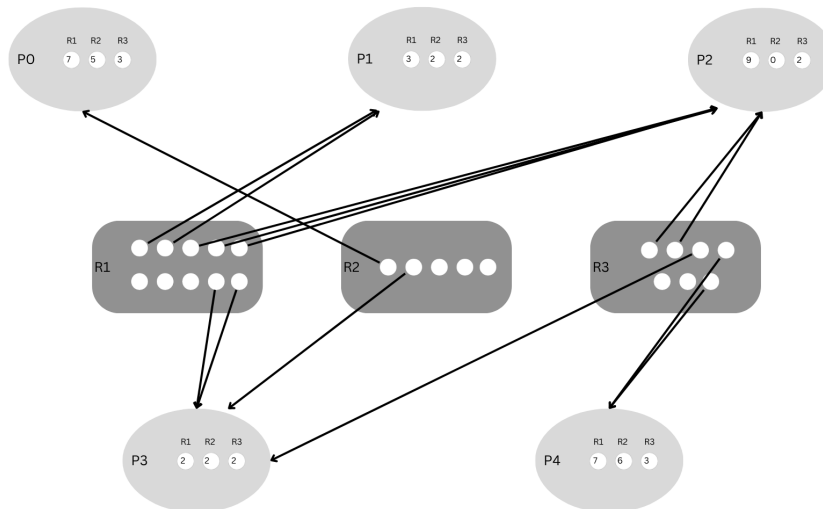
Jawab:



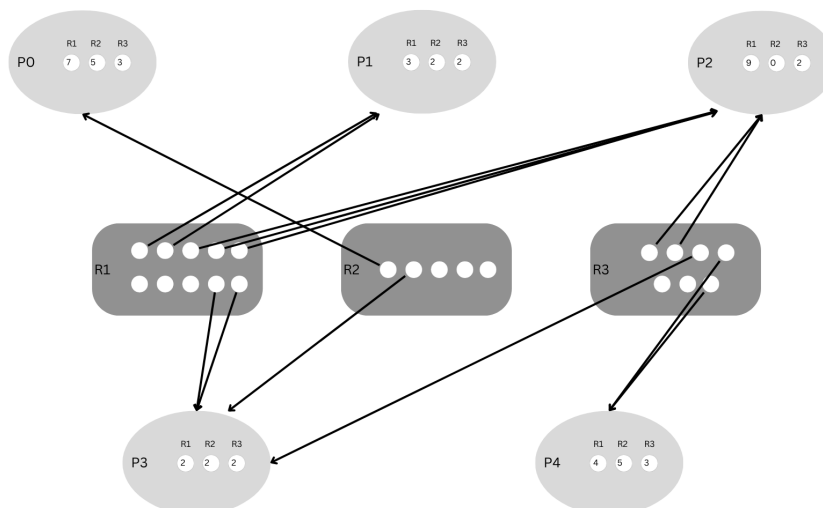


LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

Panah menunjukkan resource yang sedang dipakai dan angka pada bulatan process menunjukkan jumlah resource yang di request



Panah menunjukkan resource yang sedang dipakai dan angka pada bulatan process menunjukkan jumlah resource yang di request



Panah menunjukkan resource yang sedang dipakai dan angka pada bulatan process menunjukkan jumlah resource yang di request



6.6. Kesimpulan

Kita membahas mengenai program Banker's Algorithm yang digunakan untuk menangani masalah deadlock dalam sistem operasi. Program tersebut terdiri dari tiga bagian utama, yaitu Resource Request Algorithm, Safety Algorithm, dan Banker's Algorithm.

Resource Request Algorithm digunakan untuk memproses permintaan tambahan dari proses yang sedang berjalan terhadap sumber daya tertentu. Safety Algorithm digunakan untuk menentukan apakah urutan keamanan yang dapat diterapkan pada sistem tersebut. Sedangkan Banker's Algorithm digunakan untuk mengimplementasikan Resource Request Algorithm dan Safety Algorithm secara keseluruhan.