



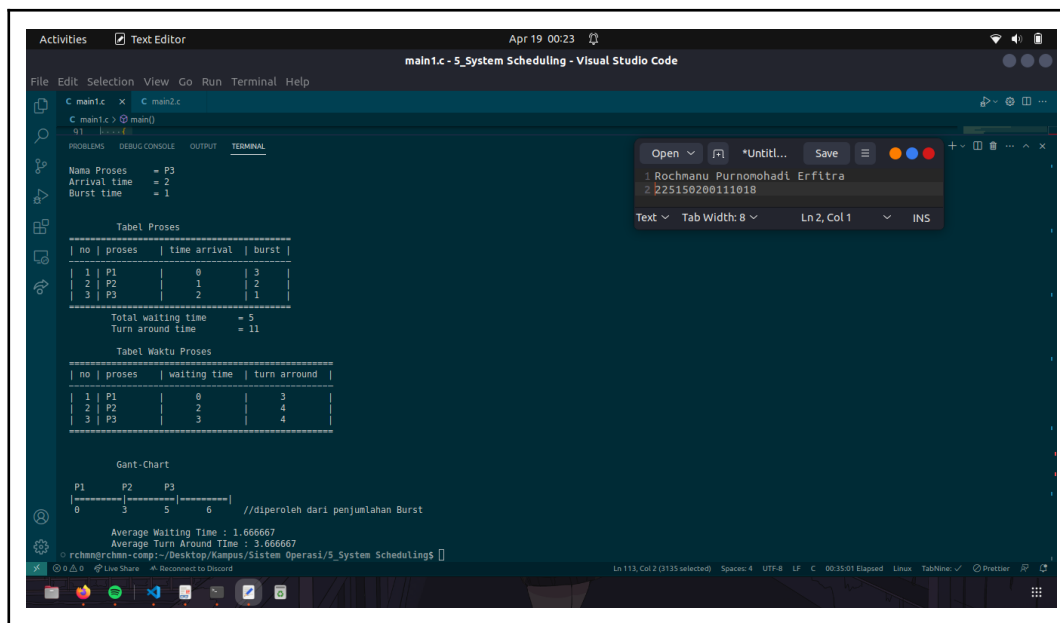
LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : CPU SCHEDULING
NAMA : ROCHMANU PURNOMOHADI ERFITRA
NIM : 225150200111018
TANGGAL : 11/04/2023
ASISTEN : ZHAFRAN RAMA AZMI
GIBRAN HAKIM

5.4. Langkah Praktikum

1. Masukkan variabel berikut dengan jumlah proses sebanyak 3. Capture / Snapshot output-nya dan simpan sebagai laporan.

Jawab:



2. Jalankan program tersebut beberapa kali dengan nilai variable masukan yang berbeda-beda. Amati output hasil percobaannya.

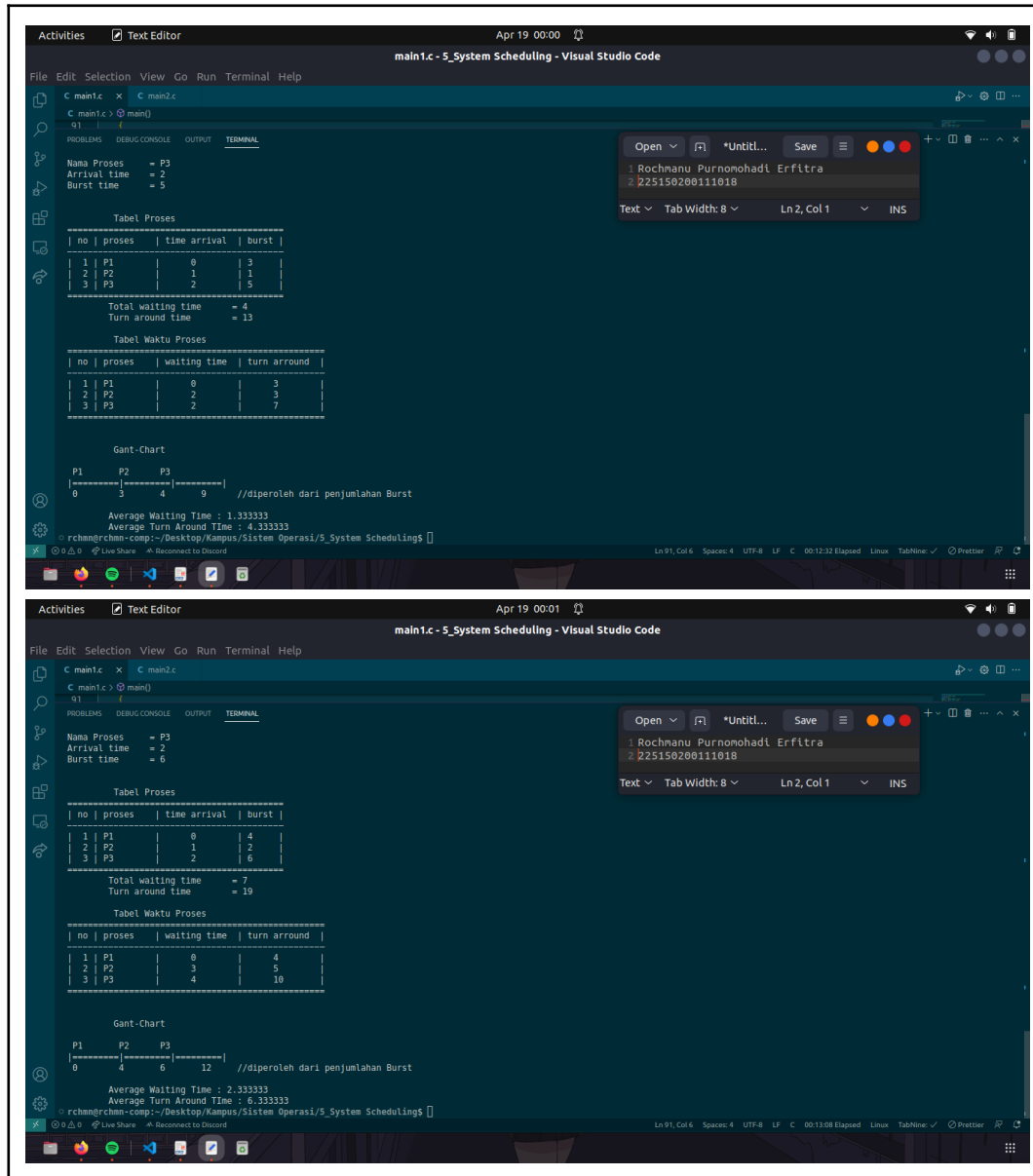
Jawab:



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER

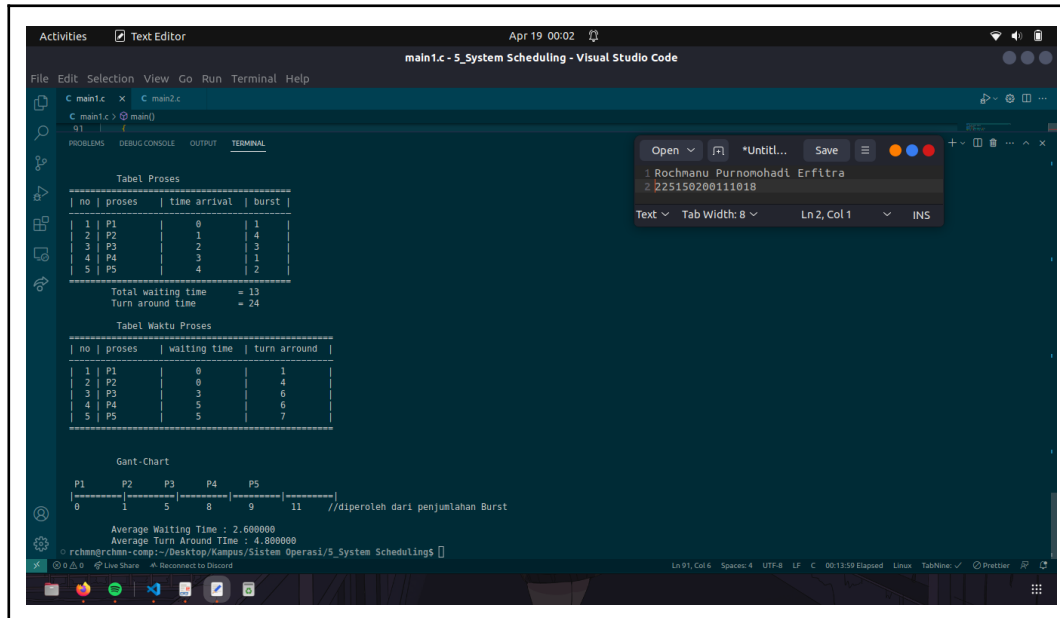
FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA





LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA



3. Dari hasil percobaan tersebut, simpulkanlah jenis algoritma penjadwalan apa yang digunakan dalam program tersebut?

Jawab:

Algoritma yang digunakan pada program diatas adalah FCFS (First Come First Serve) dimana process yang memiliki arrival time terkecil yang artinya datang terlebih dahulu akan menjadi process pertama juga yang akan diselesaikan.

5.5. Pembahasan

1. Masukkan variabel berikut dengan: jumlah proses sebanyak 3, time quantum 2. Amati hasil outputnya!

Nama Proses	Arrival Time	Burst Time
P1	0	3
P2	1	2



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

P3	2	1
----	---	---

Jawab:

```
main2.c - 5_System Scheduling - Visual Studio Code
C main2.c x
C main2.c > main()
57
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
| 3 | P3 | | 3 | 0 |
| 4 | P4 | | 5 | 6 |
| 5 | P5 | | 5 | 7 |
-----
Gantt-Chart
P1 P2 P3 P4 P5
|-----|-----|-----|-----|-----|
0 1 5 8 9 11 //diperoleh dari penjumlahan Burst
Average Waiting Time : 2.600000
Average Turn Around Time : 4.800000
rchen@rchen-comp:~/Desktop/Kampus/Sistem Operasi/5_System Scheduling$ cd "/home/rchen/Desktop/Kampus/Sistem Operasi/5_System Scheduling/" && gcc main2.c -o main2 && "/home/rchen/Desktop/Kampus/Sistem Operasi/5_System Scheduling/main2"
Masukkan jumlah proses : 3
Masukkan arrival time untuk Proses P1 : 0
Masukkan burst time untuk Proses P1 : 3
Masukkan arrival time untuk Proses P2 : 1
Masukkan burst time untuk Proses P2 : 2
Masukkan arrival time untuk Proses P3 : 2
Masukkan burst time untuk Proses P3 : 1
Masukkan time quantum 2
Process |Turnaround time|waiting time
P[2] | 3 | 1
P[3] | 3 | 2
P[1] | 6 | 3
Average Waiting Time = 2.000000
Average Turnaround Time = 4.000000
rchen@rchen-comp:~/Desktop/Kampus/Sistem Operasi/5_System Scheduling$
```

Program berjalan dengan normal, dan mengeluarkan output yang benar sesuai kaidah algoritma round robin.

2. Jika variabel pada pada nomor 4 di atas diubah menjadi: jumlah proses sebanyak 3, time quantum 2. Amati hasil outputnya!

Nama Proses	Arrival Time	Burst Time
P1	0	5
P2	20	4
P3	25	5

Jawab:



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

```
main2.c - 5_System Scheduling - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C main1.c x C main2.c x
C main2.c > @ main()
57
PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
Average Waiting Time = 2.000000
Average Turnaround Time = 4.000000
r/chmn/rchmn-comp/-/Desktop/Kampus/Sistem Operasi/5_System Scheduling$ cd "/home/rchmn/Desktop/Kampus/Sistem
o/Kampus/Sistem Operasi/5_System Scheduling/"main2
Masukkan jumlah proses : 3
Masukkan arrival time untuk Proses P1 : 0
Masukkan burst time untuk Proses P1 : 5
Masukkan arrival time untuk Proses P2 : 20
Masukkan burst time untuk Proses P2 : 4
Masukkan arrival time untuk Proses P3 : 25
Masukkan burst time untuk Proses P3 : 15
Masukkan time quantum 2

Process |Turnaround time|waiting time
P[1] | 5 | 0
^C
r/chmn/rchmn-comp/-/Desktop/Kampus/Sistem Operasi/5_System Scheduling$ cd "/home/rchmn/Desktop/Kampus/Sistem
o/Kampus/Sistem Operasi/5_System Scheduling/" && gcc main2.c -o main2 && "/home/rchmn/Deskto
p/Kampus/Sistem Operasi/5_System Scheduling/"main2
Masukkan jumlah proses : 3
Masukkan arrival time untuk Proses P1 : 0
Masukkan burst time untuk Proses P1 : 5
Masukkan arrival time untuk Proses P2 : 20
Masukkan burst time untuk Proses P2 : 4
Masukkan arrival time untuk Proses P3 : 25
Masukkan burst time untuk Proses P3 : 15
Masukkan time quantum 2

Process |Turnaround time|waiting time
P[1] | 5 | 0
```

Sebenarnya program ini menggunakan algoritma round robin dimana ia mengandalkan quantum time, namun khusus pada test case ini mungkin terjadi kesalahan sehingga menyebabkan program terus menerus dieksekusi karena mungkin program ini tidak mendeteksi adanya proses selanjutnya karena mungkin waiting time yang terlalu terlampau jauh yaitu 0 ke 20.

3. Berdasarkan hasil pengamatan anda, maka algoritma apa yang digunakan dalam program tersebut?

Jawab:

Algoritma yang digunakan adalah algoritma round robin dimana algoritma round robin menggunakan sistem time sharing dengan static quantum time untuk setiap proses yang akan dieksekusi CPU. Algoritma ini tergantung pada ukuran quantum time yang diberikan.

5.6. Kesimpulan

Kita mempelajari dasar dari cpo scheduling yang nantinya konsep ini akan lebih diterapkan dalam deadlock dan cara menghindarinya pada materi selanjutnya