

LAPORAN PRAKTIKUM SISTEM OPERASI

MODUL 11: PENJADWALAN PROSES



Disusun oleh:

Nama : Riyan Catur Agusta

NIM : L200210262

Kelas : Praktikum SO E

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

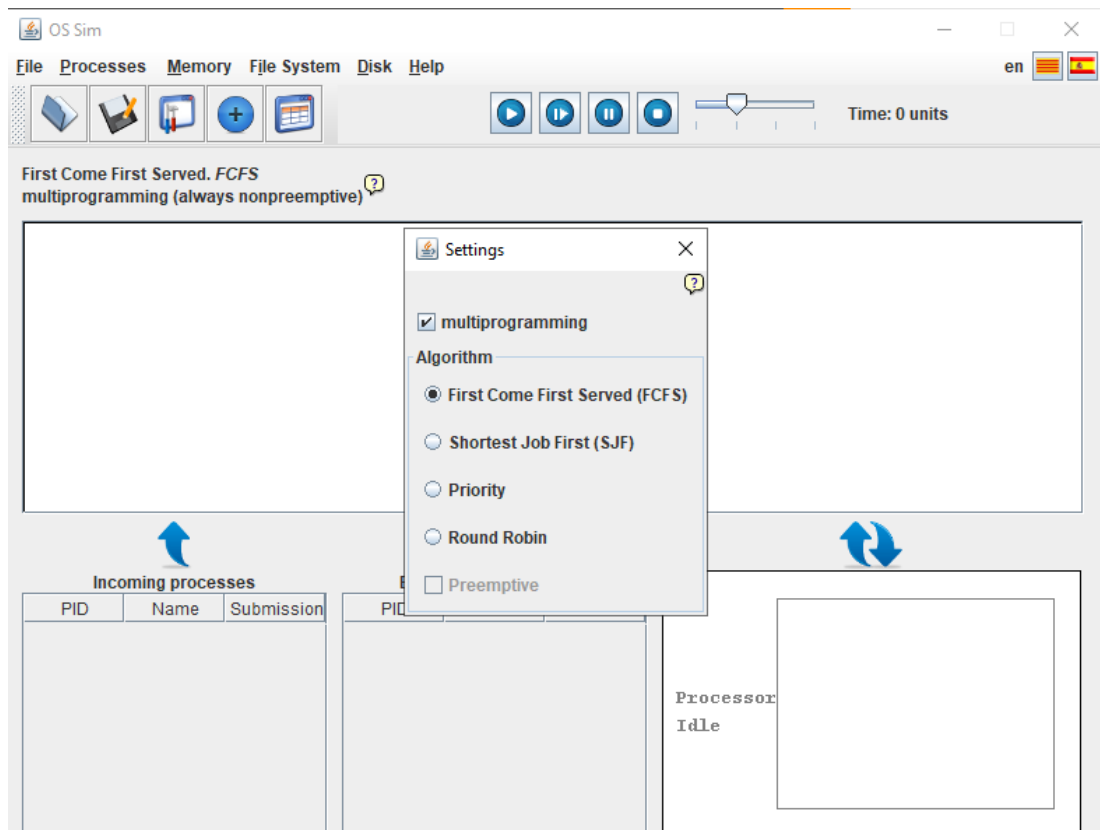
UNIVERSITAS MUHAMMADIYAH SURAKARTA TAHUN 2022/2023

LANGKAH KERJA

Kegiatan 1. Penjadwalan Proses

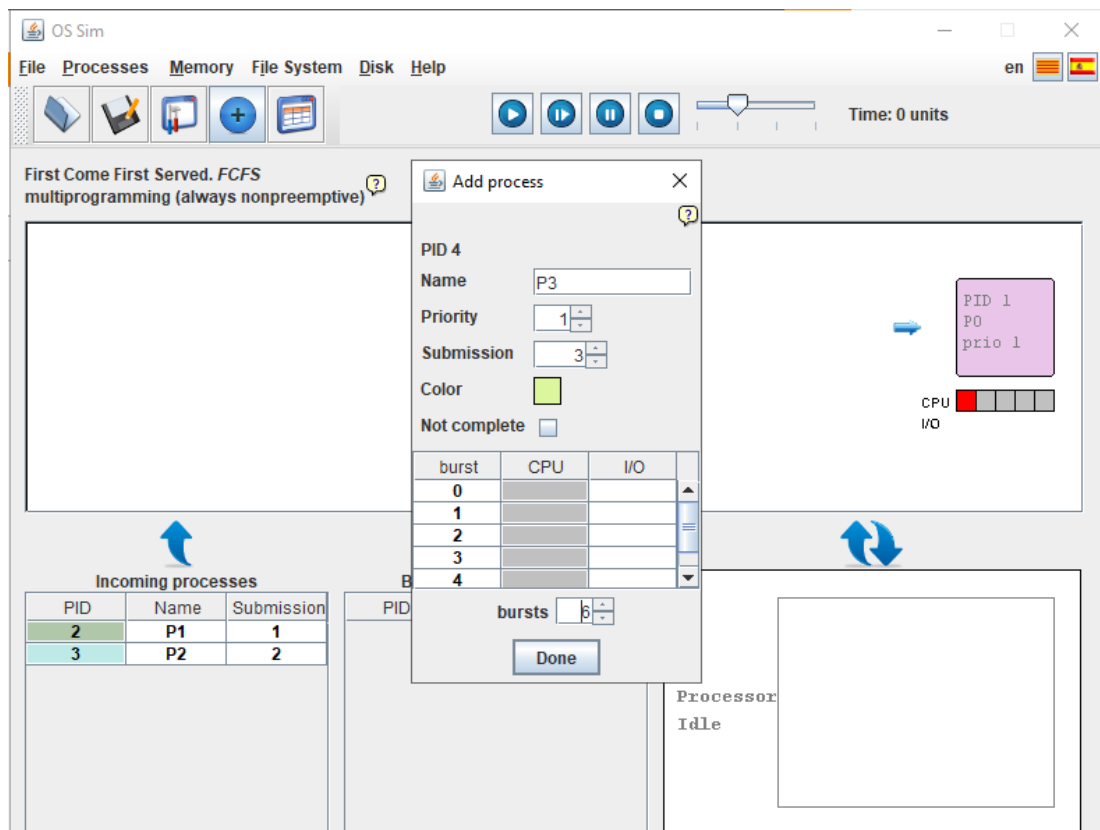
1.1 First-Come, First-Served (FCFS)

- Bukalah program OSSim, selanjutnya pilih menu processes → process scheduling.
- Selanjutnya pilihlah setting dan pilih algoritma **First-Come, First-Served (FCFS)**.



- Lakukan input proses sesuai dengan tabel berikut dengan memulai dengan P0 sebagai input proses yang pertama.

Process	Arrival Time	Burst Time	Service Time
P0	0	5	0
P1	1	3	5
P2	2	8	8
P3	3	6	16



- d) Jika input sudah selesai dilakukan. Pilih tombol start pada bagian atas. Amati dan analisa proses yang terjadi.

OS Sim

File Processes Memory File System Disk Help

Time: 22 units

Process Scheduling Information

Efficiency (%)	1.00
Throughput (processes/time unit)	0.18
Avg. Turnaround Time (time)	11.25
Avg. Waiting Time (time)	5.75
Avg. Response Time (time)	5.75

PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	0	5	1.0	0.0
2	P1	1	1	-	3	4	4	7	0.4285714...	0.0
3	P2	1	2	-	8	6	6	14	0.5714285...	0.0
4	P3	1	3	-	6	13	13	19	0.3157894...	0.0

e) Isilah tabel berikut ini.

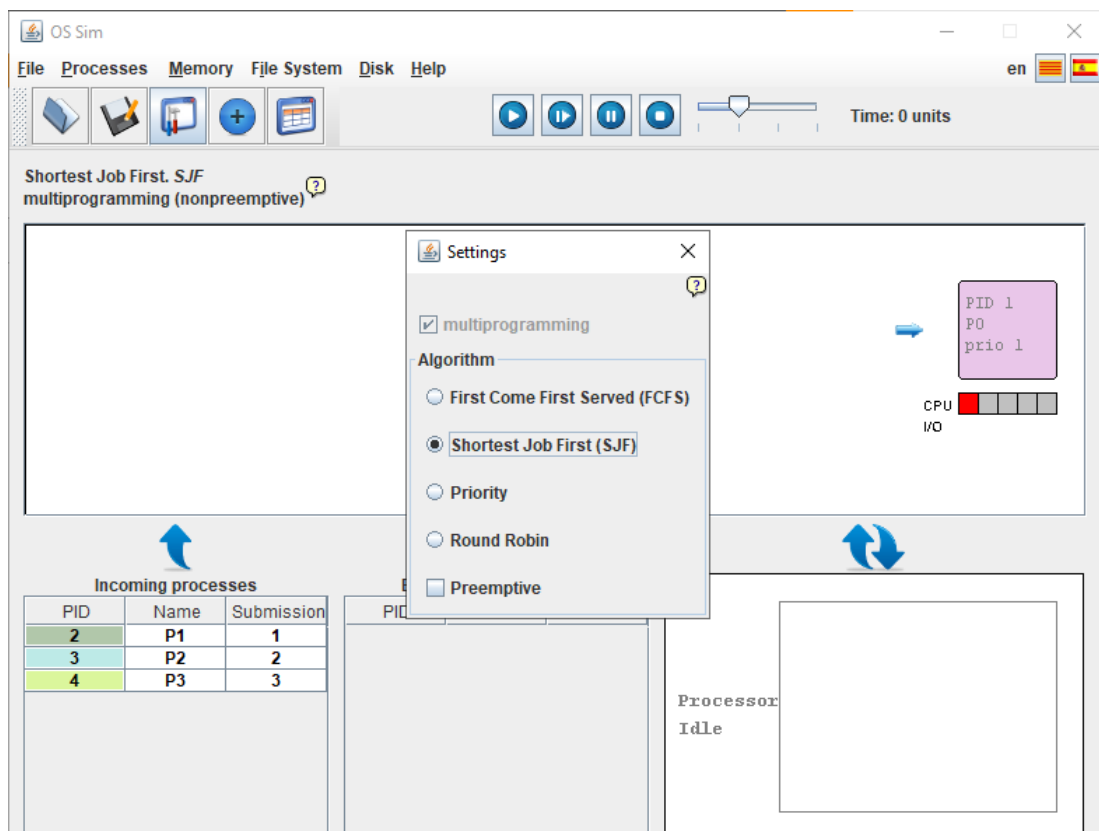
Process	Wait time : Service Time – Arrival Time
P0	0
P1	4
P2	6
P3	13
Av wait time	5.75

Kesimpulan:

FCFS merupakan penjadwalan yang paling sederhana yang digunakan CPU. Dengan menggunakan algoritma ini setiap proses yang berada pada status ready dimasukkan kedalam FIFO queue atau antrian dengan prinsip first in first out, sesuai dengan waktu kedatangannya. Proses yang tiba terlebih dahulu yang akan dieksekusi.

1.2 Shortest Job First (SJF)

- Bukalah program OS Sim, selanjutnya pilih menu processes → process scheduling
- Selanjutnya pilihlah setting dan pilih algoritma Shortest Job First (SJF). Algoritma ini terdiri dari 2 jenis yaitu non-preemptive dan preemptive. Untuk mengaktifkan preemptive dengan mencentang menu tersebut. Sebaliknya jika menonaktifkan maka hanya cukup menghilangkan centangnya saja.



- c) Selanjutnya klik tombol start. Amati dan analisa proses yang terjadi. Lakukan perbandingan dari hasil keduanya.

Non-Preemptive

OS Sim											
File Processes Memory File System Disk Help											
Time: 22 units											
Process Scheduling Information											
Efficiency (%) 1.00											
Throughput (processes/time unit) 0.18											
Avg. Turnaround Time (time) 10.75											
Avg. Waiting Time (time) 5.25											
Avg. Response Time (time) 5.25											
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO	
1	P0	1	0	-	5	0	0	5	1.0	0.0	
2	P1	1	1	-	3	4	4	7	0.4285714...	0.0	
4	P3	1	3	-	6	5	5	11	0.5454545...	0.0	
3	P2	1	2	-	8	12	12	20	0.4	0.0	

Preemptive

OS Sim											
File Processes Memory File System Disk Help											
Time: 22 units											
Process Scheduling Information											
Efficiency (%) 1.00											
Throughput (processes/time unit) 0.18											
Avg. Turnaround Time (time) 10.50											
Avg. Waiting Time (time) 5.00											
Avg. Response Time (time) 4.25											
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO	
2	P1	1	1	-	3	0	0	3	1.0	0.0	
1	P0	1	0	-	5	0	3	8	0.625	0.0	
4	P3	1	3	-	6	5	5	11	0.5454545...	0.0	
3	P2	1	2	-	8	12	12	20	0.4	0.0	

d) Isilah tabel berikut.

Non-Preemptive

Process	Wait time : Service Time – Arrival Time
P0	0
P1	4
P2	5
P3	12
Av wait time	5.25

Preemptive

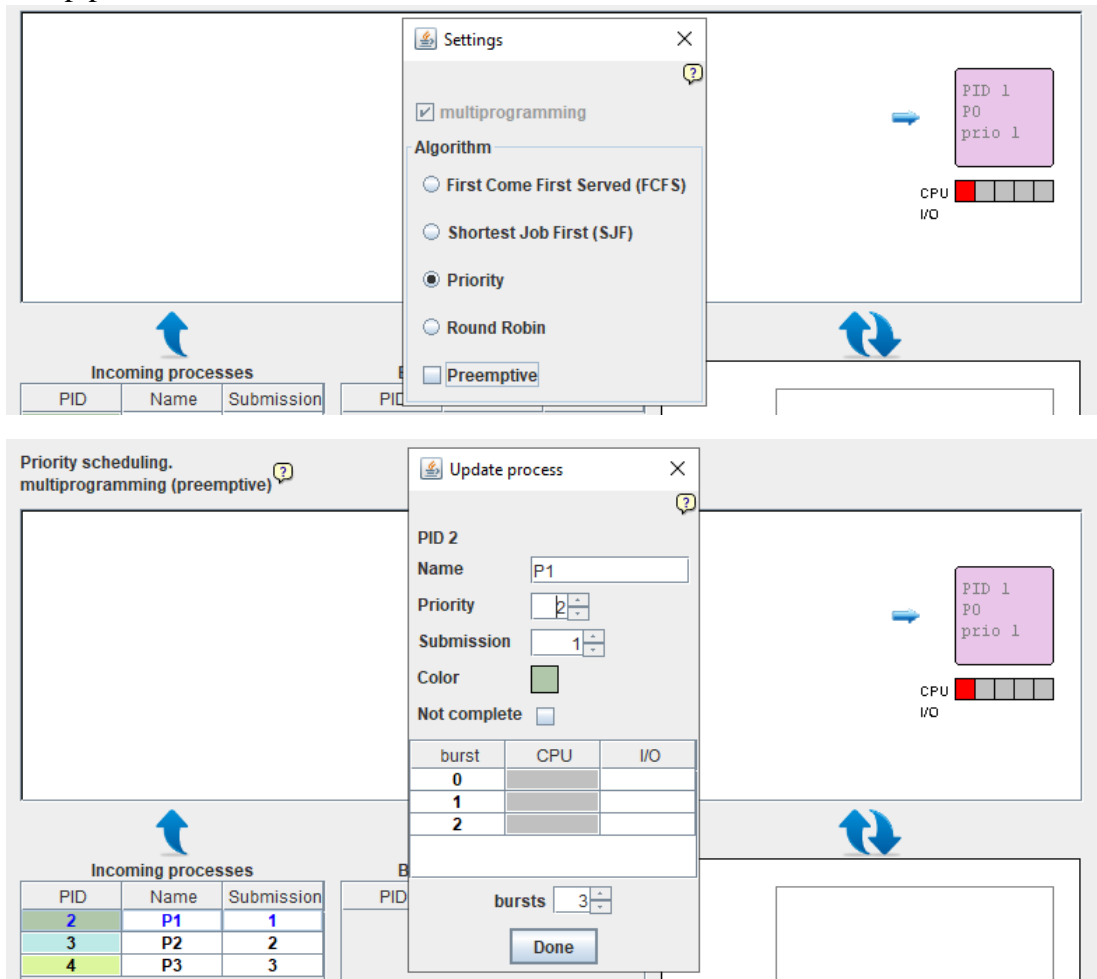
Process	Wait time : Service Time – Arrival Time
P0	0
P1	3
P2	5
P3	12
Av wait time	5.00

Kesimpulan:

Pada algoritma ini setiap proses yang ada di ready queue akan dieksekusi berdasarkan burst time terkecil. Hal ini mengakibatkan waiting time yang pendek untuk setiap proses dan karena hal tersebut maka waiting time rata-ratanya juga menjadi pendek, sehingga dapat dikatakan bahwa algoritma ini adalah algoritma yang optimal.

1.3 Priority

- a) Pilihlah menu setting dan pilih algoritma Priority. Selanjutnya tambahkan priority pada setiap proses.



Settings

- ☒ multiprogramming
- Algorithm
 - ☐ First Come First Served (FCFS)
 - ☐ Shortest Job First (SJF)
 - ☒ Priority
 - ☐ Round Robin
- ☐ Preemptive

Update process

PID 2

Name: P1

Priority: 2

Submission: 1

Color: [Green]

Not complete: ☐

burst	CPU	I/O
0		
1		
2		

bursts: 3

Done

Priority scheduling. multiprogramming (preemptive)

Incoming processes

PID	Name	Submission
2	P1	1
3	P2	2
4	P3	3

Process Scheduling Information

Efficiency (%) 1.00

Throughput (processes/time unit) 0.18

Avg. Turnaround Time (time) 11.50

Avg. Waiting Time (time) 6.00

Avg. Response Time (time) 6.00

PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	0	5	1.0	0.0
4	P3	3	3	-	6	2	2	8	0.75	0.0
2	P1	2	1	-	3	10	10	13	0.2307692...	0.0
3	P2	1	2	-	8	12	12	20	0.4	0.0

- b) Selanjutnya klik tombol start. Lakukan pengamatan dan analisa proses yang terjadi. Lengkapilah tabel berikut!

Process Scheduling Information										
Efficiency (%)					1.00					
Throughput (processes/time unit)					0.18					
Avg. Turnaround Time (time)					11.50					
Avg. Waiting Time (time)					6.00					
Avg. Response Time (time)					6.00					
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
1	P0	1	0	-	5	0	0	5	1.0	0.0
4	P3	3	3	-	6	2	2	8	0.75	0.0
2	P1	2	1	-	3	10	10	13	0.2307692...	0.0
3	P2	1	2	-	8	12	12	20	0.4	0.0

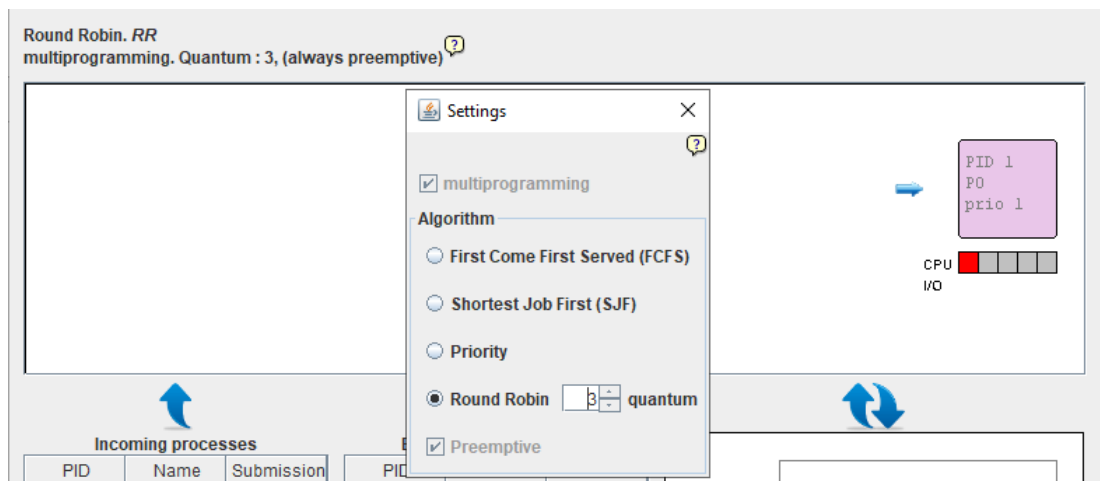
Process	Wait time : Service Time – Arrival Time
P0	0
P1	2
P2	10
P3	12
Av wait time	6.00

Kesimpulan:

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing.

1.4 Round Robin

- Pilihlah menu setting dan pilih algoritma Round Robin. Selanjutnya tambahkan quantum time sebesar 3.



- b) Selanjutnya klik tombol start. Lakukan pengamatan dan analisa proses yang terjadi. Lengkapilah tabel berikut:

Process Scheduling Information										
Efficiency (%)		1.00								
Throughput (processes/time unit)		0.18								
Avg. Turnaround Time (time)		14.00								
Avg. Waiting Time (time)		8.50								
Avg. Response Time (time)		3.00								
PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
2	P1	2	1	-	3	2	2	5	0.6	0.0
1	P0	1	0	-	5	0	9	14	0.3571428...	0.0
4	P3	3	3	-	6	6	11	17	0.3529411...	0.0
3	P2	1	2	-	8	4	12	20	0.4	0.0

Process	Wait time : Service Time – Arrival Time
P0	2
P1	9
P2	11
P3	12
Av wait time	8.50

Kesimpulan:

Algoritma ini menggilir proses yang ada di antrian. Proses akan mendapat jatah sebesar time quantum. Jika time quantum-nya habis atau proses sudah selesai, CPU akan dialokasikan ke proses berikutnya. Tentu proses ini cukup adil karena tak ada proses yang diprioritaskan, semua proses mendapat jatah waktu yang sama dari CPU yaitu $(1/n)$, dan tak akan menunggu lebih lama dari $(n-1)q$ dengan q adalah lama 1 quantum.

Algoritma ini sepenuhnya bergantung besarnya time quantum. Jika terlalu besar, algoritma ini akan sama saja dengan algoritma first come first served. Jika terlalu kecil, akan semakin banyak peralihan proses sehingga banyak waktu terbuang.