PRAKTIKUM SISTEM OPERASI

MODUL 11

Disusun Oleh :

MUHAMMAD MIFTAHUL HUDA

L200210230

E

TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2022/2023

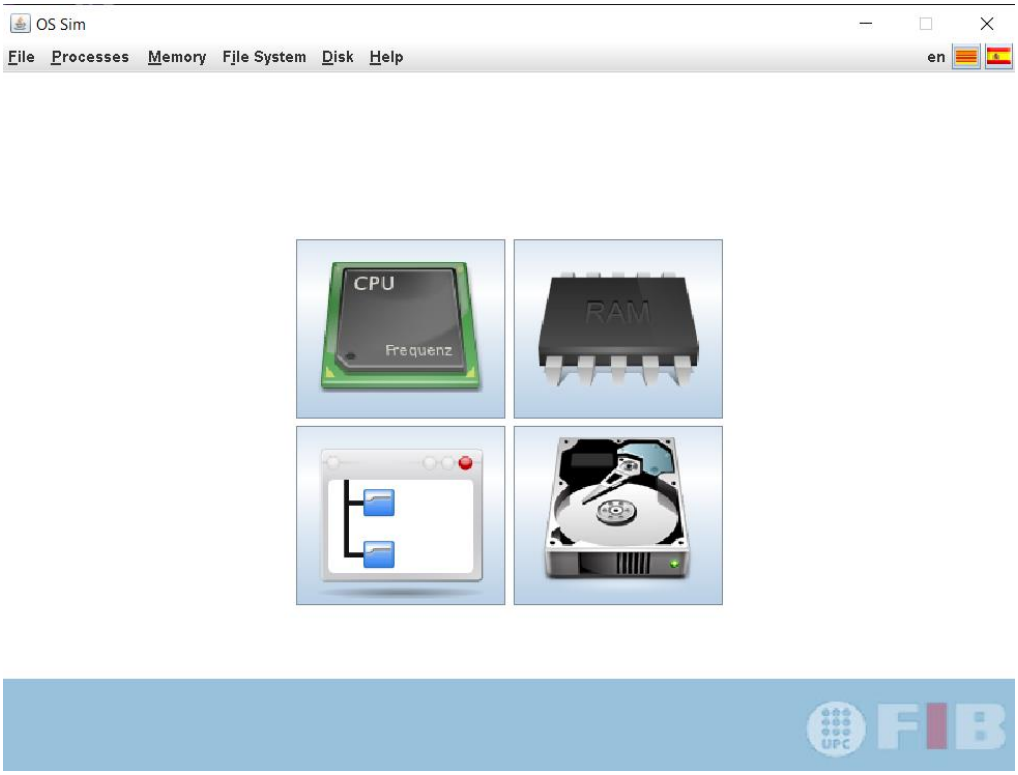
Kegiatan 1

First-Come, First-Served (FCFS)

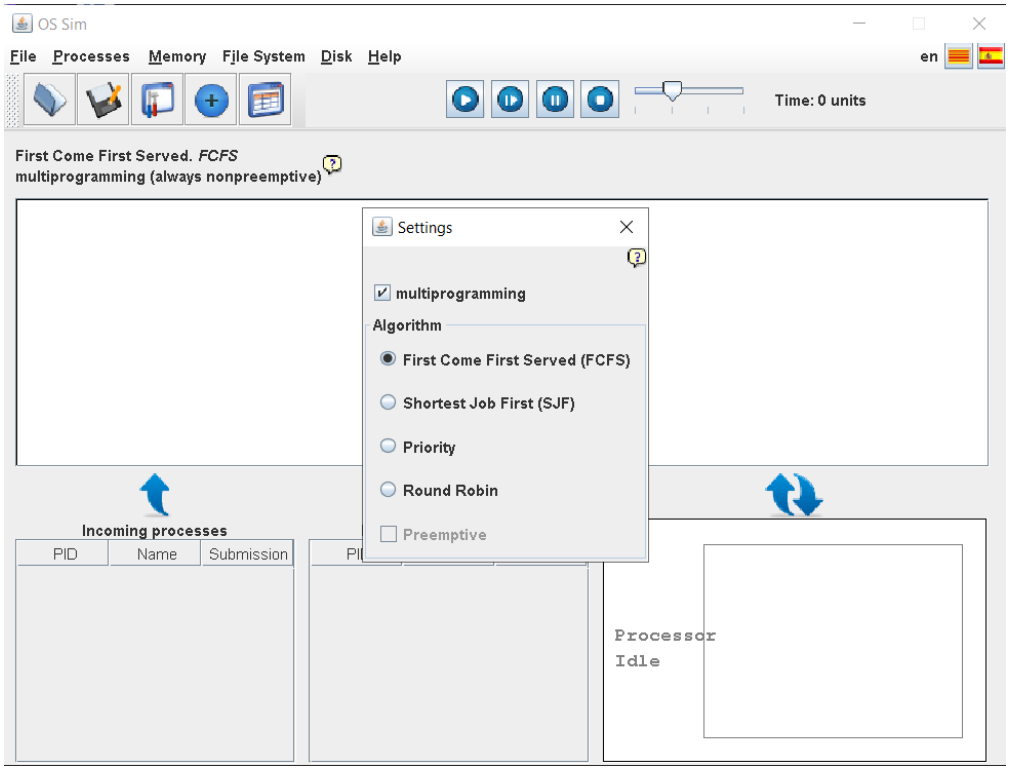
1. Membuka program OSSim, selanjutnya memilih menu processes -> process scheduling
2. Selanjutnya memilih setting dan pilih algoritma First-Come, First-Served (FCFS)
3. Melakukan input proses sesuai dengan tabel berikut dengan memulai dengan P0 sebagai input proses yang pertama.

Proses	Arrival time	Burst time	Service time
P0	0	5	0
P1	1	3	5
P2	2	8	8
P3	3	6	16

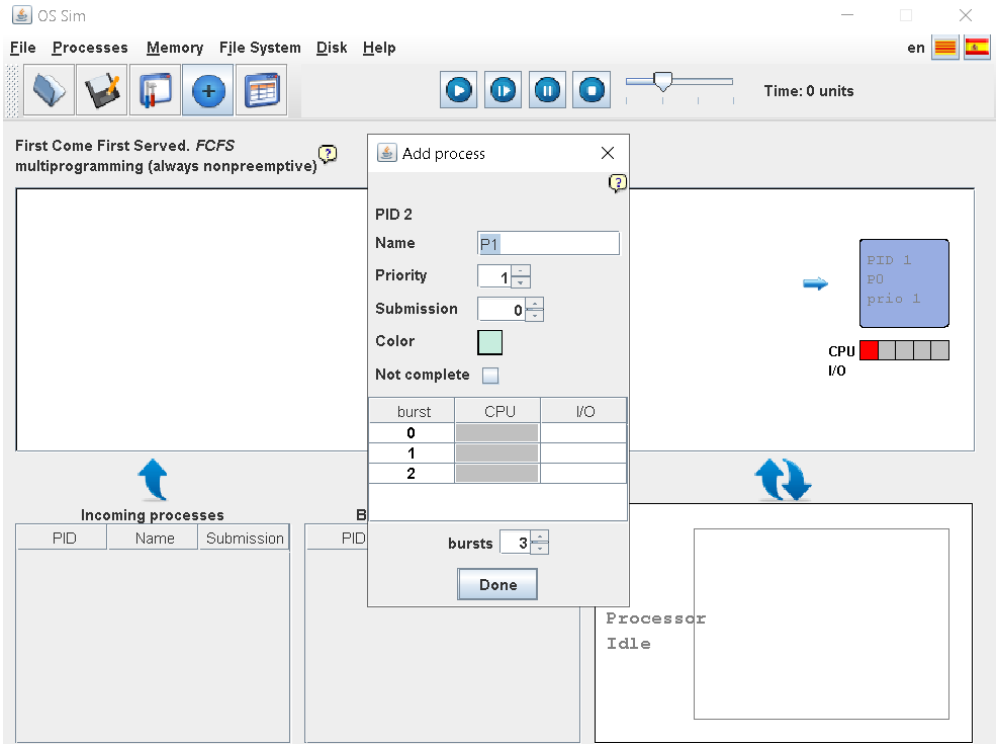
- Membuka program OSSim.jar

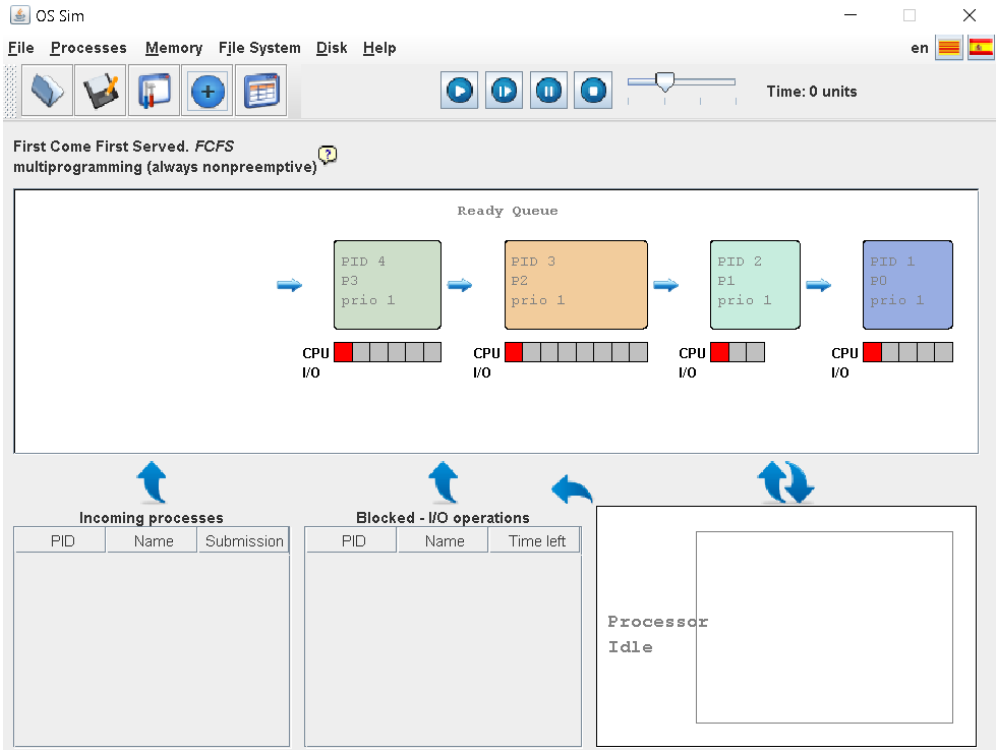


- Memilih algoritma FCFS

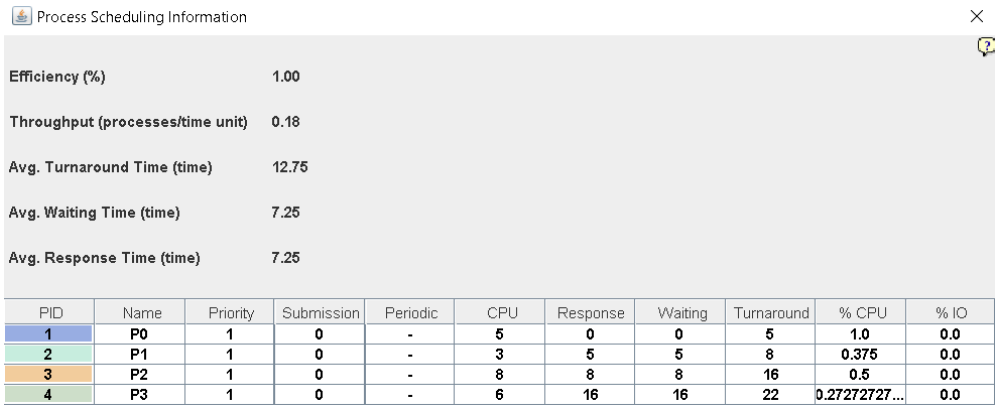


- Proses Penginputan


OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
First Come First Served. FCFS
multiprogramming (always nonpreemptive)
Add process
PID 2
Name P1
Priority 1
Submission 0
Color
Not complete
burst CPU I/O
0
1
2
bursts 3
Done
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Processor
Idle


OS Sim
File Processes Memory File System Disk Help
en
Time: 0 units
First Come First Served. FCFS
multiprogramming (always nonpreemptive)
Ready Queue
PID 4
P3
prio 1
PID 3
P2
prio 1
PID 2
P1
prio 1
PID 1
P0
prio 1
CPU
I/O
Incoming processes
PID Name Submission
Blocked - I/O operations
PID Name Time left
Processor
Idle

- Output



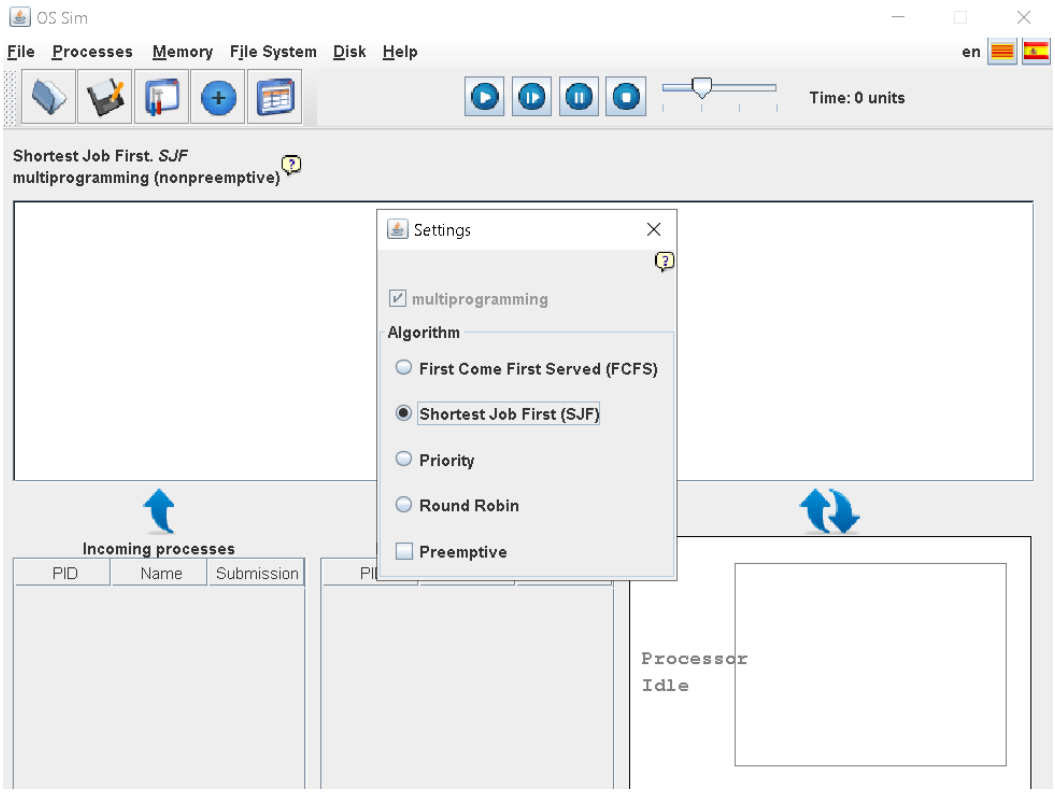
- Tabel :

Process	Wait Time : Service Time-Arrival Time
P0	0
P1	5
P2	8
P3	16
Av Wait Time	7.25

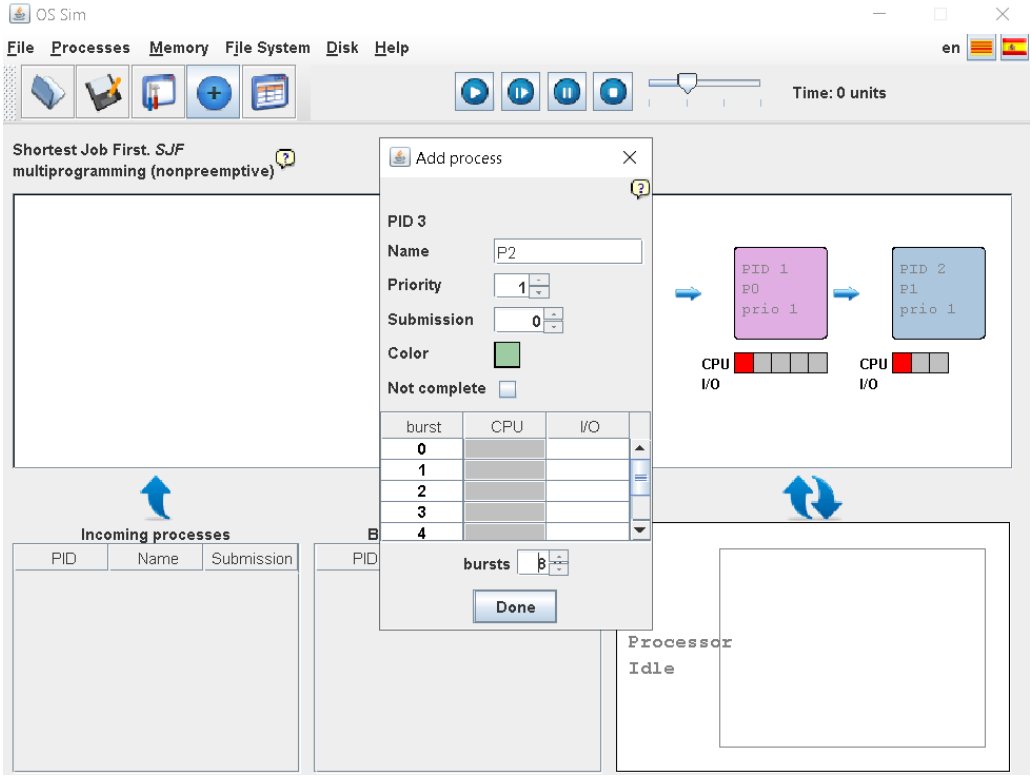
Shortest Job First (SJF)

1. Membuka program OS Sim, selanjutnya memilih menu processes -> process scheduling
2. Selanjutnya memilih setting dan pilih algoritma Shortest Job first (SJF). Algoritma ini terdiri dari 2 jenis yaitu non-preemtive dan preemptive. Untuk mengaktifkan preemptive dengan mencentang menu tersebut. Sebaliknya jika menonaktifkan maka hanya cukup menghilangkan centangnya saja.
3. Selanjutnya klik tombol start. Mengamati dan menganalisa proses yang terjadi dan melakukan perbandingan dari hasil keduanya.

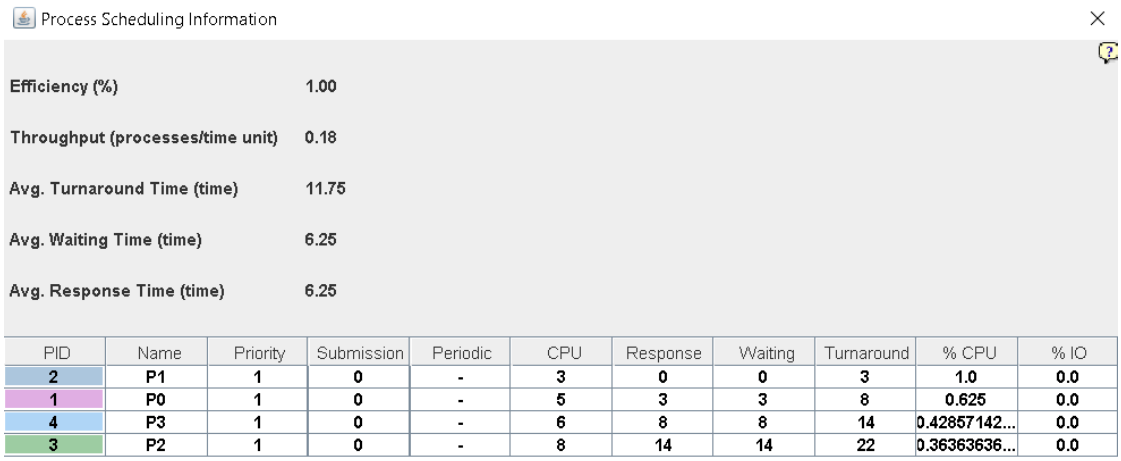
- SJF (non Preemptive)



- Proses Penginputan



- Output

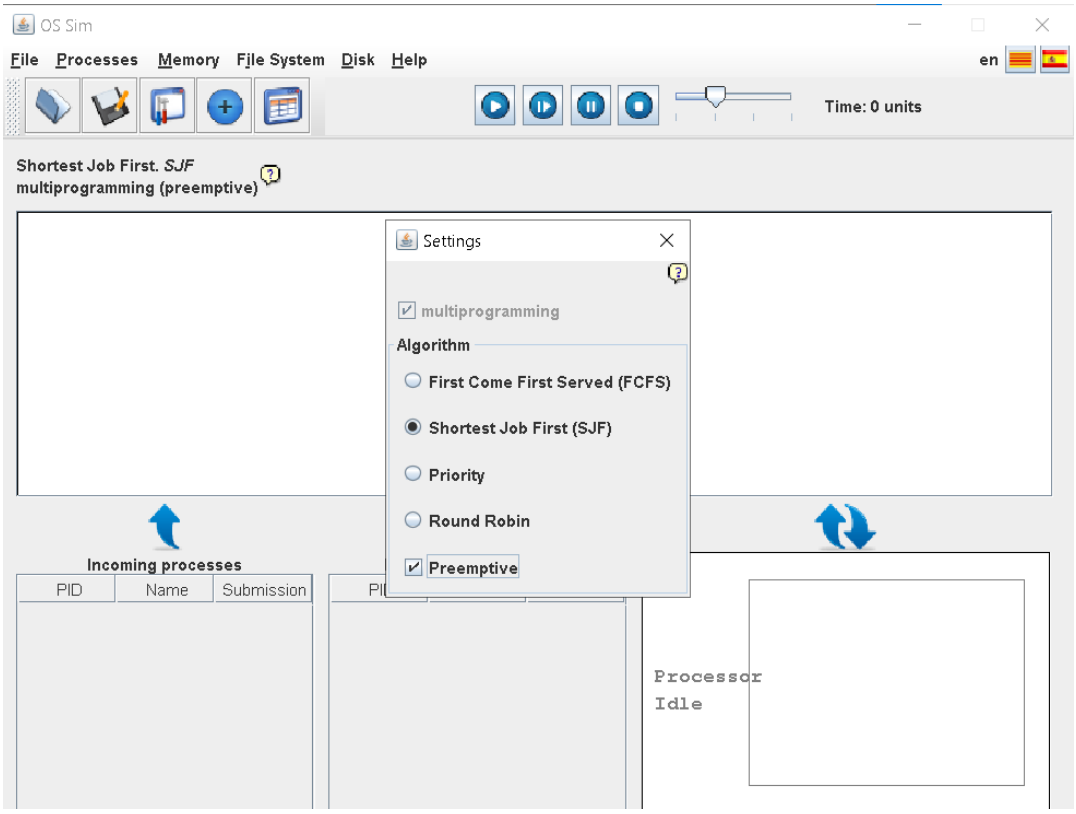


- Tabel (nonpreemtive)

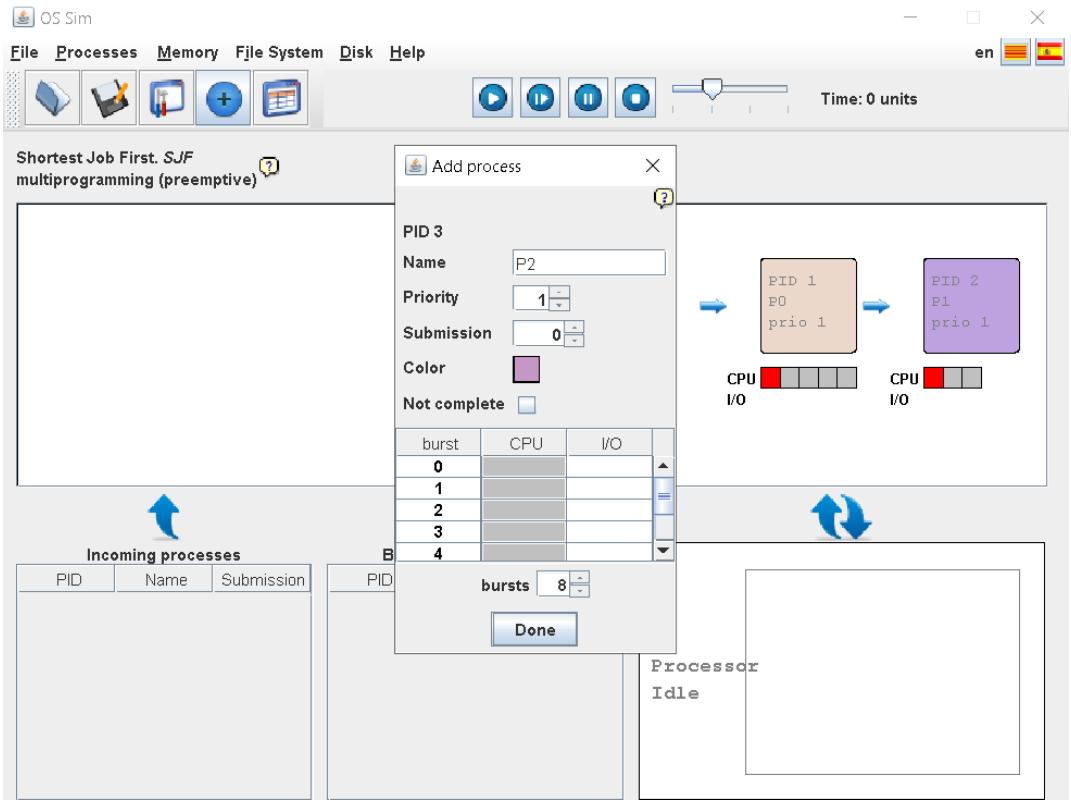
Process	Wait Time : Service Time-Arrival Time
P0	0
P1	3
P2	8
P3	14
Av Wait Time	6.25

Pada algoritma ini, proses dengan burst time terkecil akan dieksekusi terlebih dahulu, sehingga akan dieksekusi berurutan sesuai dengan besar burst time yang dibutuhkan.

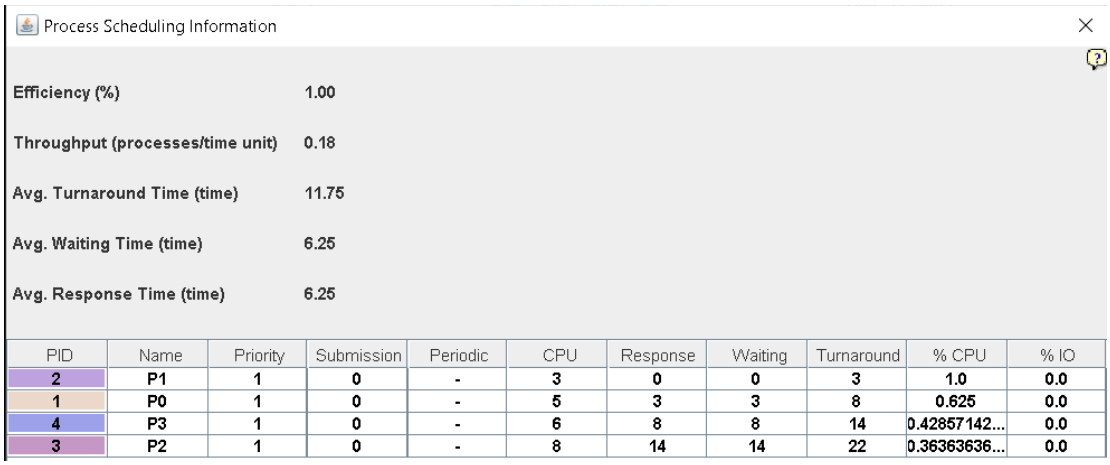
- SJF (Preemtive)



- Proses Penginpputan



- Output



PID	Name	Priority	Submission	Periodic	CPU	Response	Waiting	Turnaround	% CPU	% IO
2	P1	1	0	-	3	0	0	3	1.0	0.0
1	P0	1	0	-	5	3	3	8	0.625	0.0
4	P3	1	0	-	6	8	8	14	0.42857142...	0.0
3	P2	1	0	-	8	14	14	22	0.36363636...	0.0

- Tabel (Preemtive)

Process	Wait Time : Service Time-Arrival Time
P0	0
P1	3
P2	8
P3	14
Av Wait Time	6.25

Pada algoritma ini, proses yang memiliki burst time paling sedikit akan dieksekusi terlebih dahulu, dan berlaku pada proses yang pertama kali datang (preemtive) sehingga ratarata waktu tunggu akan kecil.

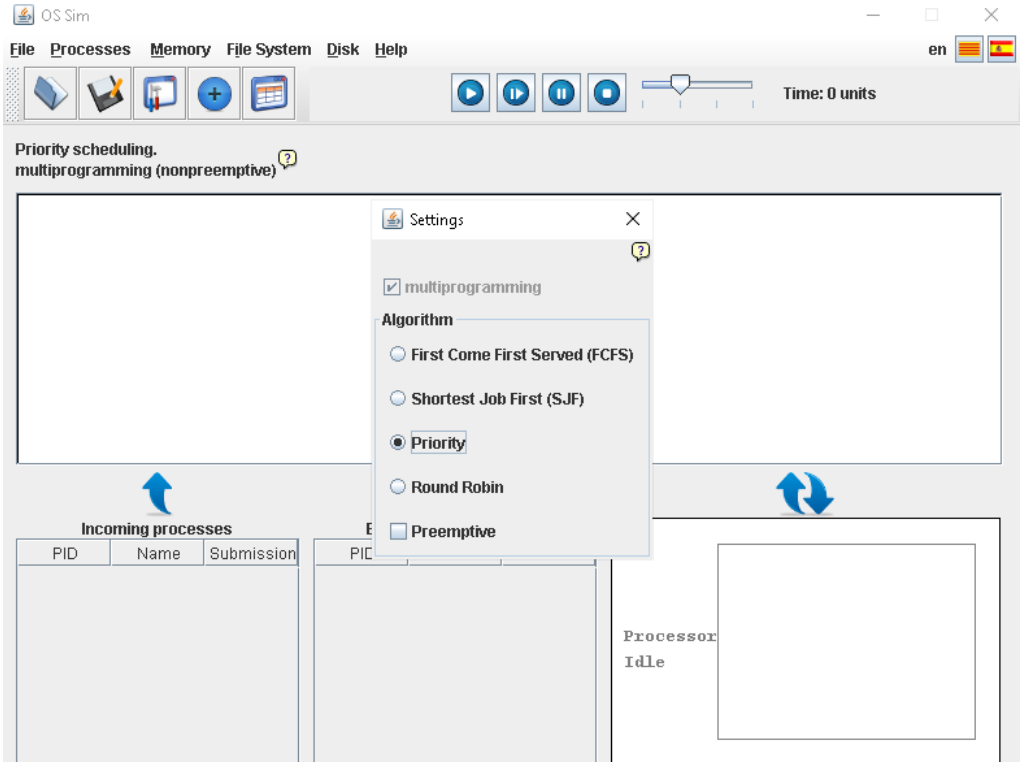
Priority

1. Memilih menu setting dan pilih algoritma Priority. Selanjutnya menambahkan priority pada setiap proses.

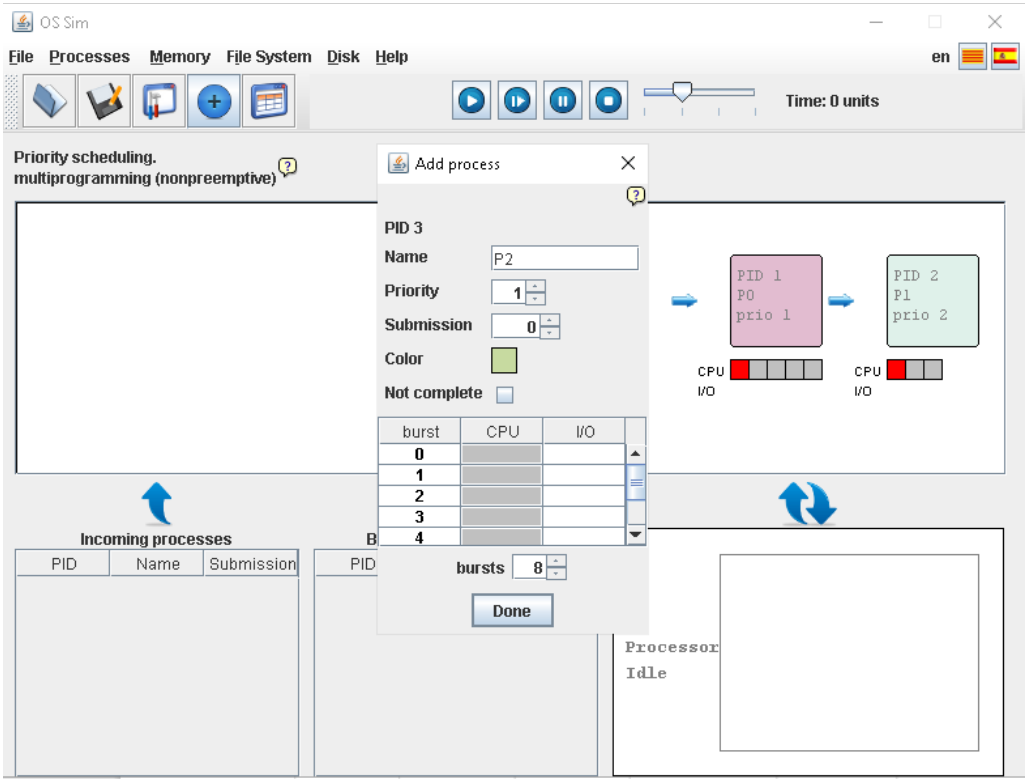
Proses	Arrival Time	Burst time	Priority	Service Time
P0	0	5	1	0
P1	1	3	2	11
P2	2	8	1	14
P3	3	6	3	5

2. Selanjutnya melakukan pengamatan dan Analisa proses yang terjadi.

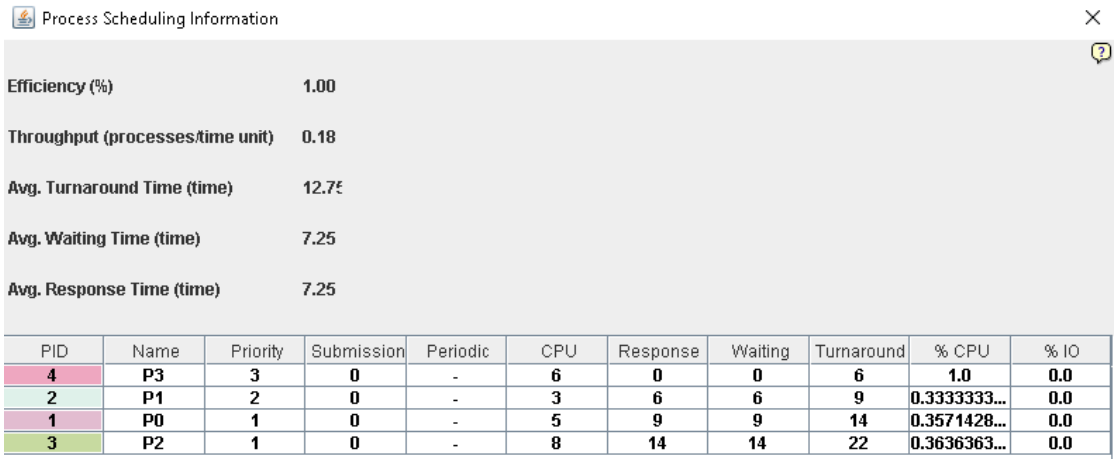
    - Memilih algoritma Priority



- Penginputan Proses



- Output



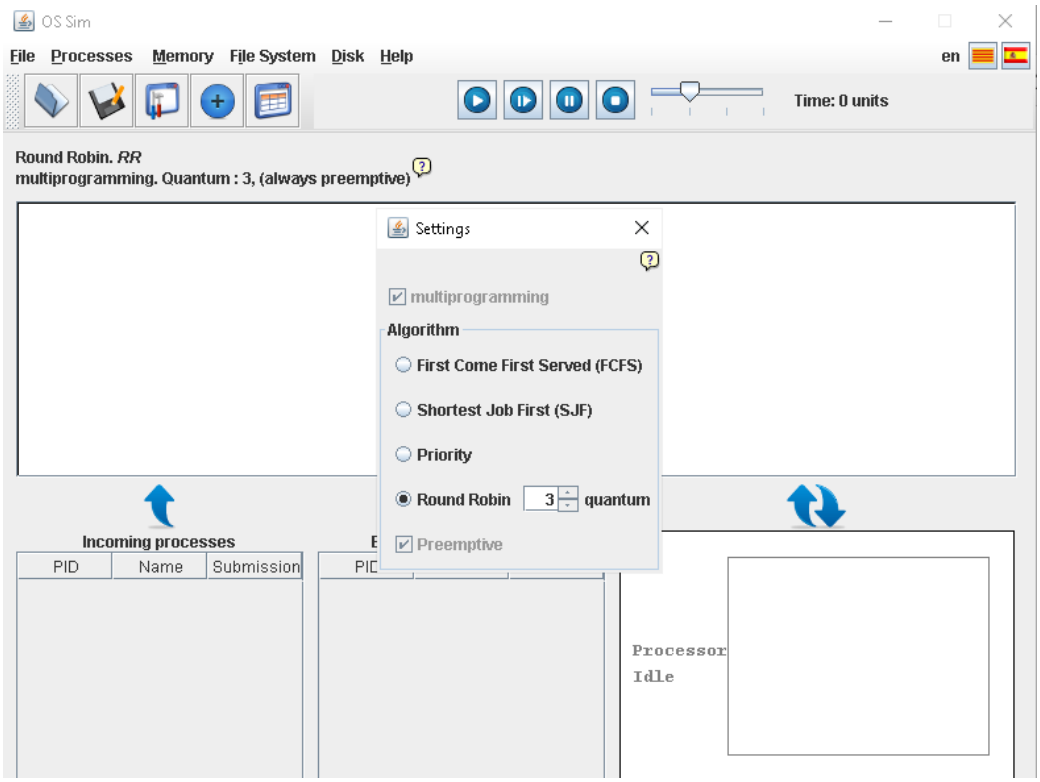
- Tabel

Process	Wait Time : Service Time-Arrival Time
P0	0
P1	6
P2	9
P3	14
Av Wait Time	7.25

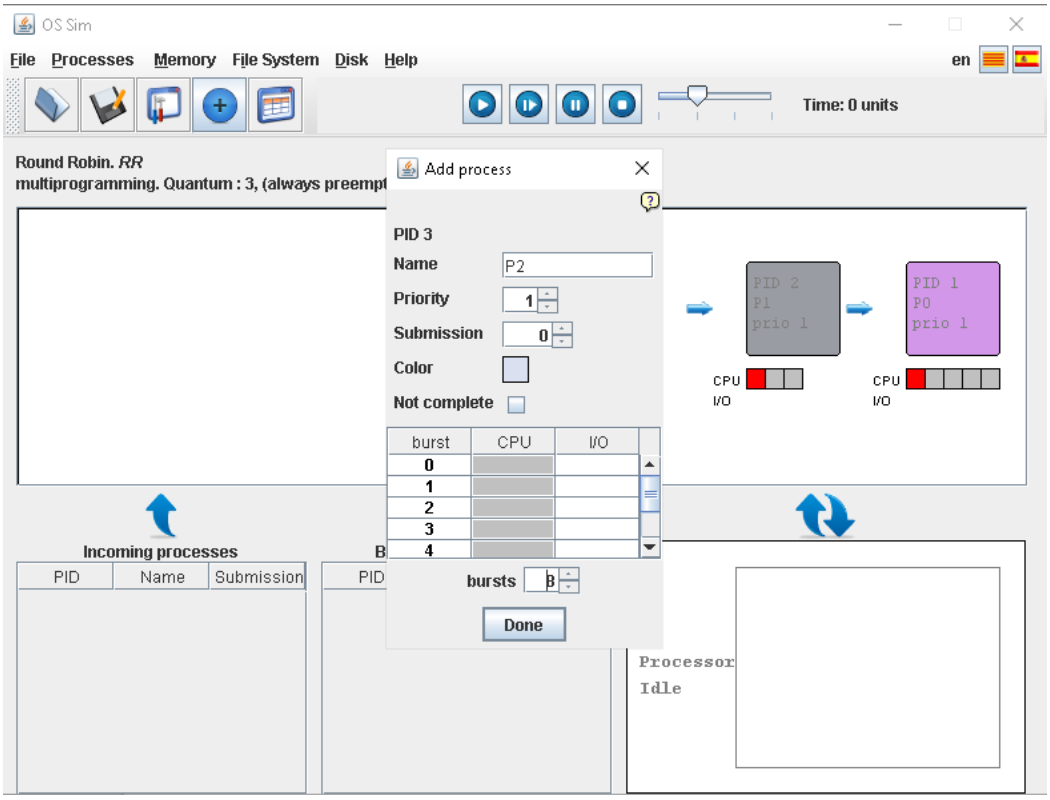
Round Robin

1. Memilih menu setting dan pilih algoritma Round Robin. Selanjutnya menambahkan quantum time sebesar 3.
2. Selanjutnya melakukan pengamatan dan menganalisa proses yang terjadi.

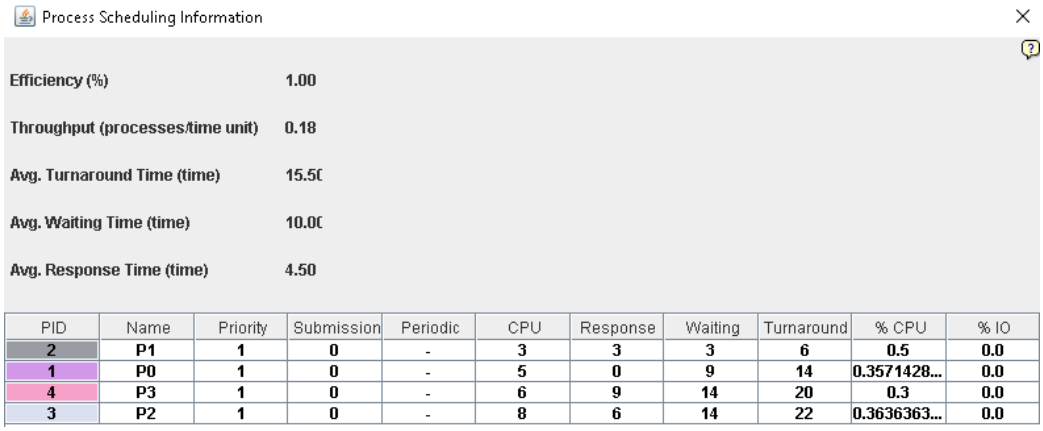
- Memilih algoritma Round Robin dengan quantum 3



- Penginputan Proses



- Output



- Tabel

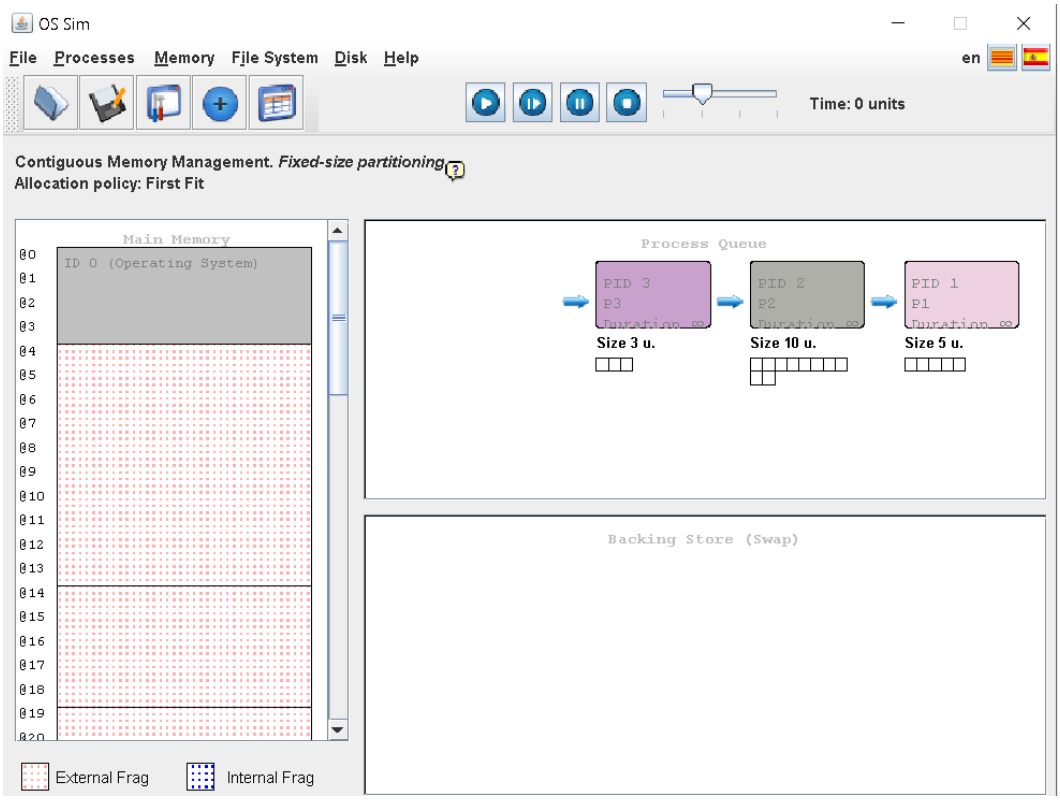
Process	Wait Time : Service Time-Arrival Time
P0	3
P1	9
P2	14
P3	14
Av Wait Time	10.00

Kegiatan 2. Manajemen Memori

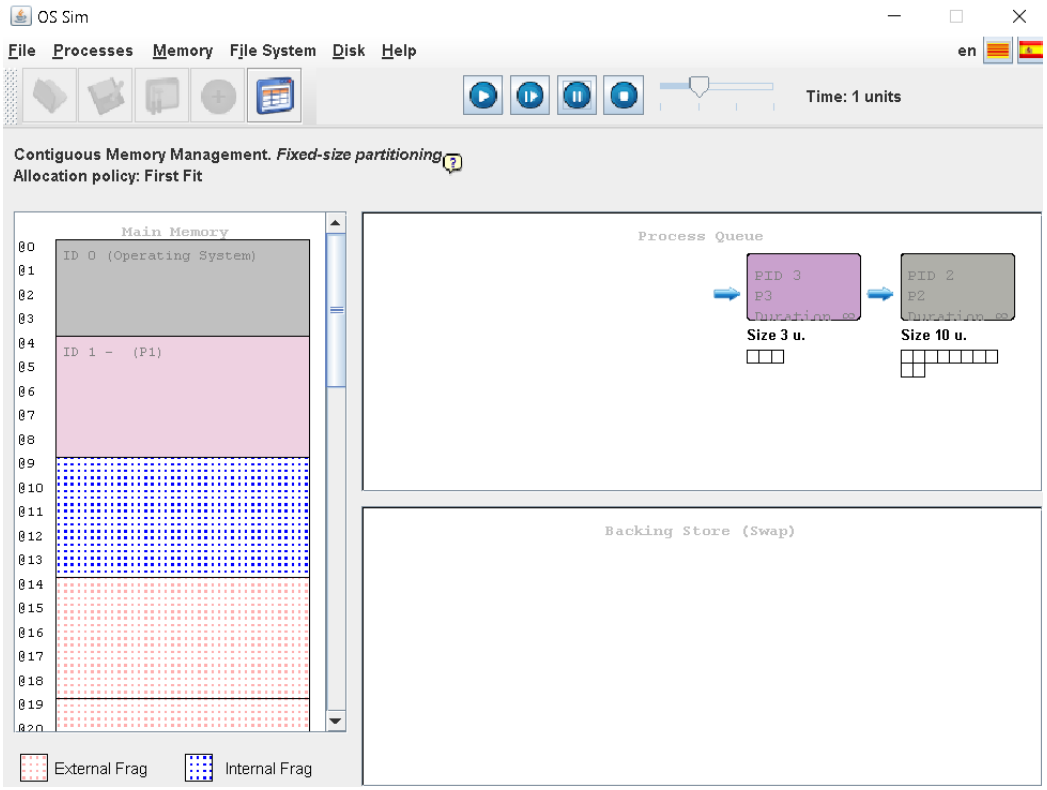
2.1 Contiguous memory management dengan menggunakan partisi berukuran tetap (ficed-size partition) dan aturan first fit.

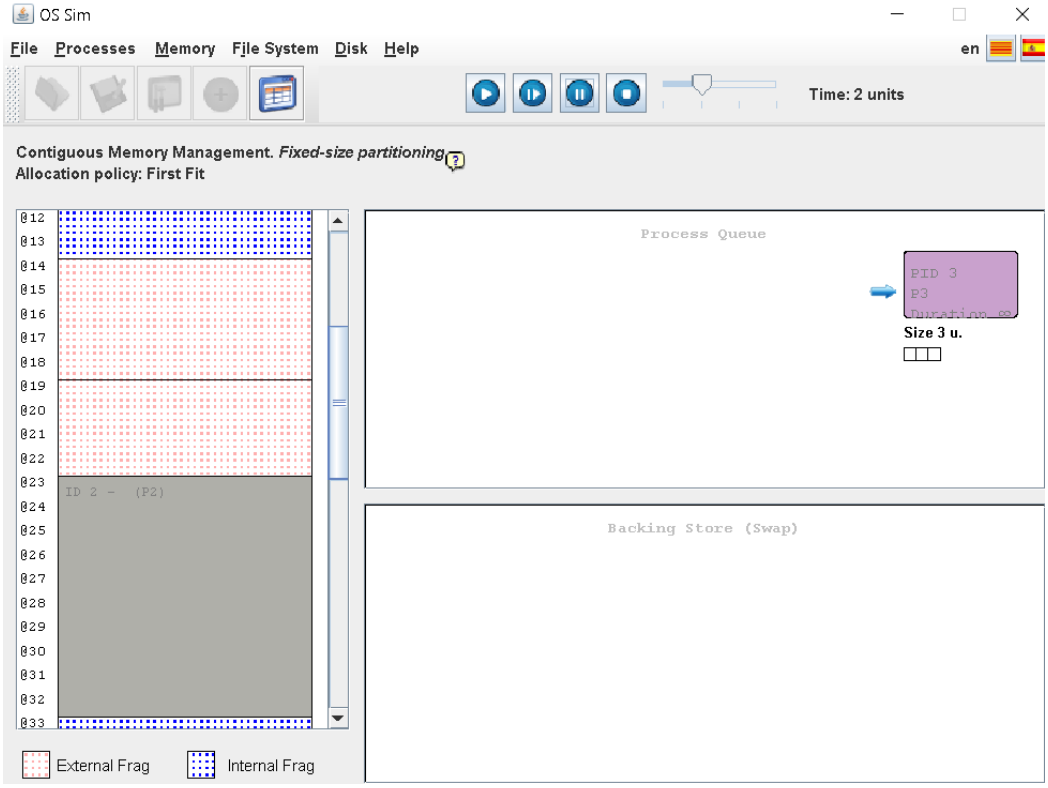
Memori dibagi menjadi partisi-partisi berukuran tetap, lalu seluruh proses akan dialokasikan pada satu dari partisi yang tersedia. Pada aturan first fit, partisi pertama yang dapat memuat proses akan dipilih.

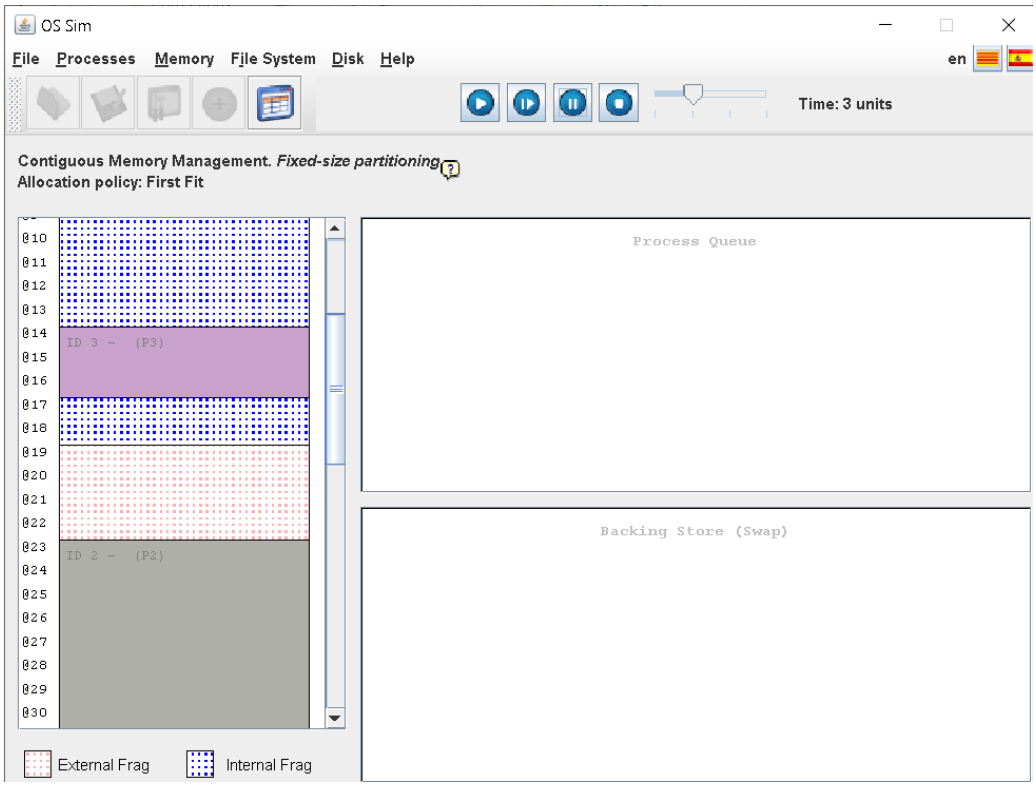
1. Pilih menu help >> examples...>> memory management >> Contiguous memory management with fixed-size partitions (first adjustment). Pilihlah folder hingga menampilkan sebagai berikut.



2. Selanjutnya mengeklik tombol play untuk melihat proses yang terjadi dan mengeklik tombol stop untuk berhenti.
3. Melakukan Analisa proses dengan melakukan klik pada tombol next.


OS Sim
File Processes Memory File System Disk Help
en
Time: 1 units
Contiguous Memory Management. Fixed-size partitioning
Allocation policy: First Fit
Main Memory
ID 0 (Operating System)
ID 1 - (P1)
Process Queue
PID 3
P3
Size 3 u.
PID 2
P2
Size 10 u.
Backing Store (Swap)
External Frag
Internal Frag


OS Sim
File Processes Memory File System Disk Help
en
Time: 2 units
Contiguous Memory Management. Fixed-size partitioning
Allocation policy: First Fit
ID 2 - (P2)
Process Queue
PID 3
P3
Size 3 u.
Backing Store (Swap)
External Frag
Internal Frag



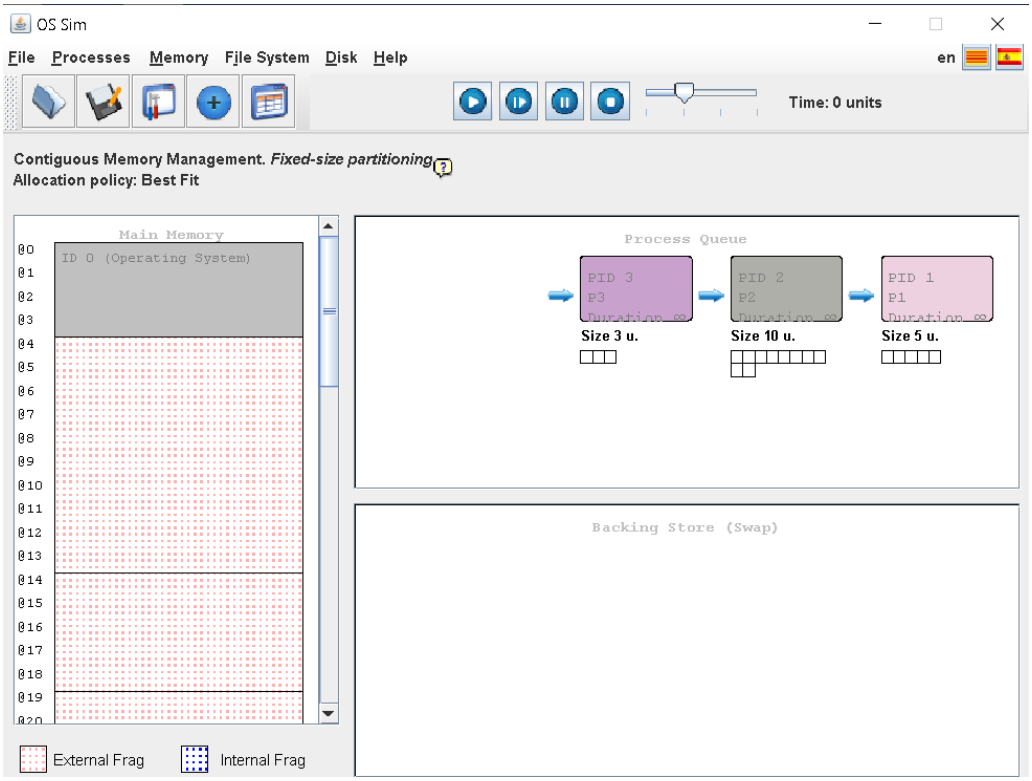
Dengan aturan first fit, banyak terjadi internal fragmentasi. Ini terjadi karena ketika suatu proses menemukan partisi yang cukup untuk dimuat, makai a akan dimuat di partisi tersebut tanpa memandang apakah sisa partisi banyak atau tidak.

2.2 Contiguous memory management dengan menggunakan partisi berukuran tetap (fixed-size partition) dan aturan best fit

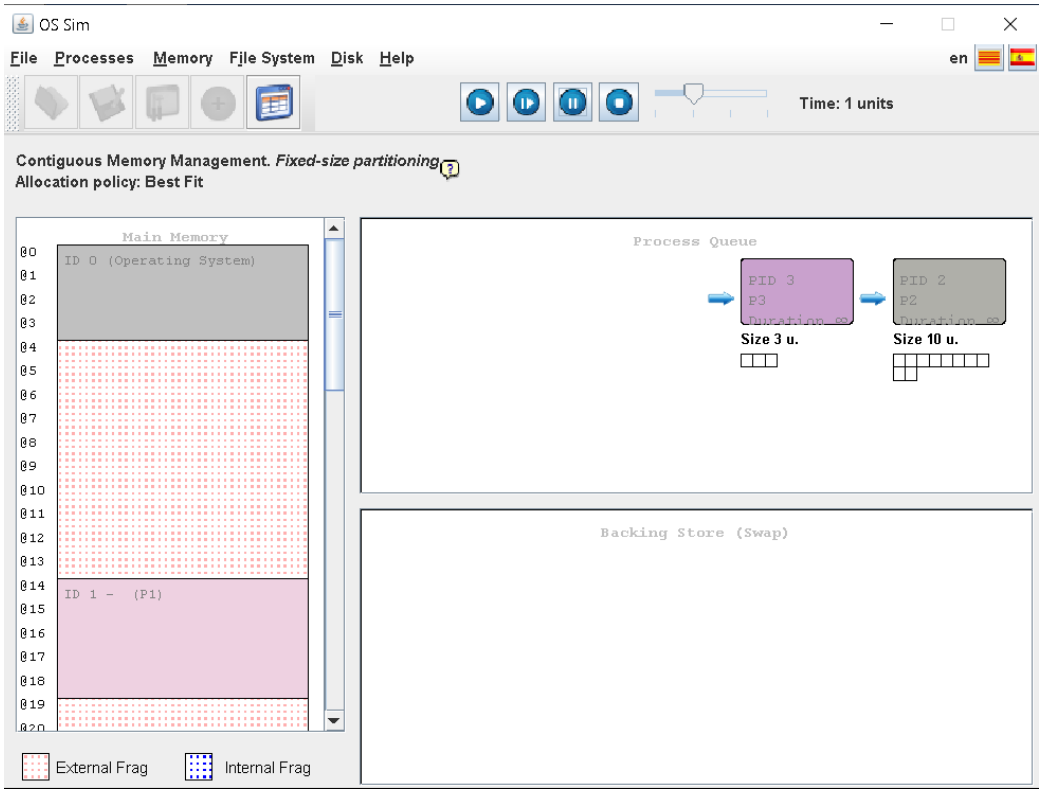
Memori dibagi menjadi partisi-partisi berukuran tetap, lalu seluruh proses akan dialokasikan pada satu dari partisi yang tersedia. Pada aturan best fit, partisi terbaik (yang ukurannya sama atau hampir sama) yang dapat memuat proses akan dipilih

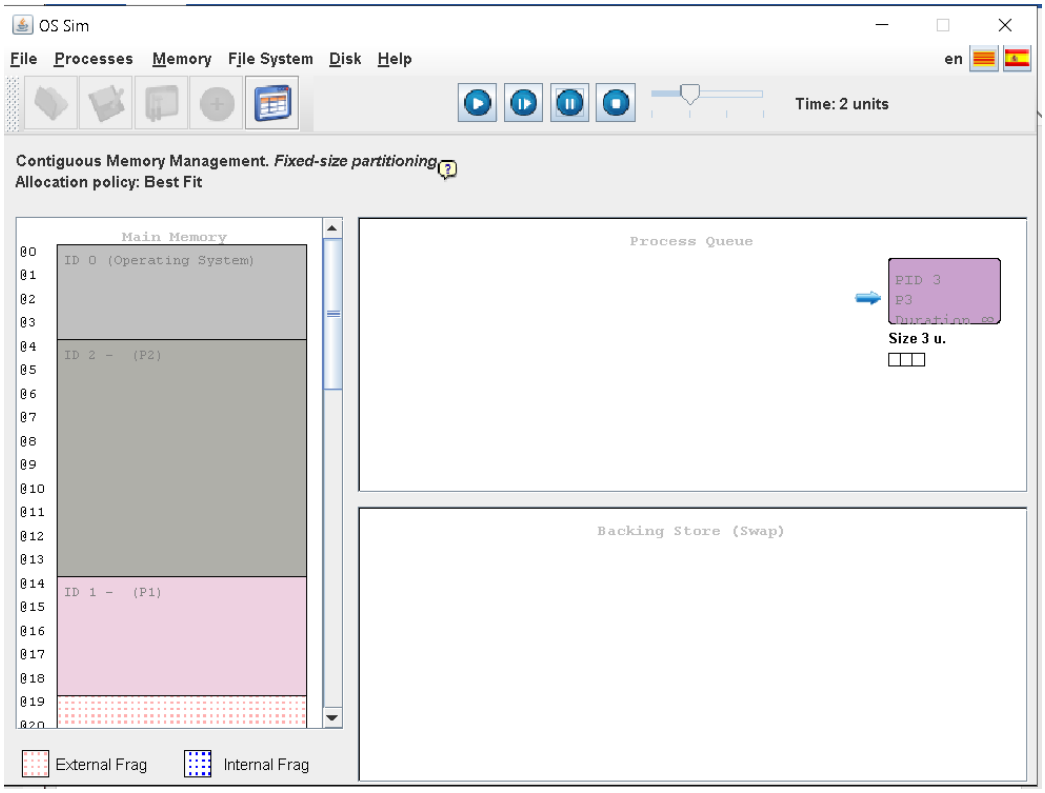
Langkah:

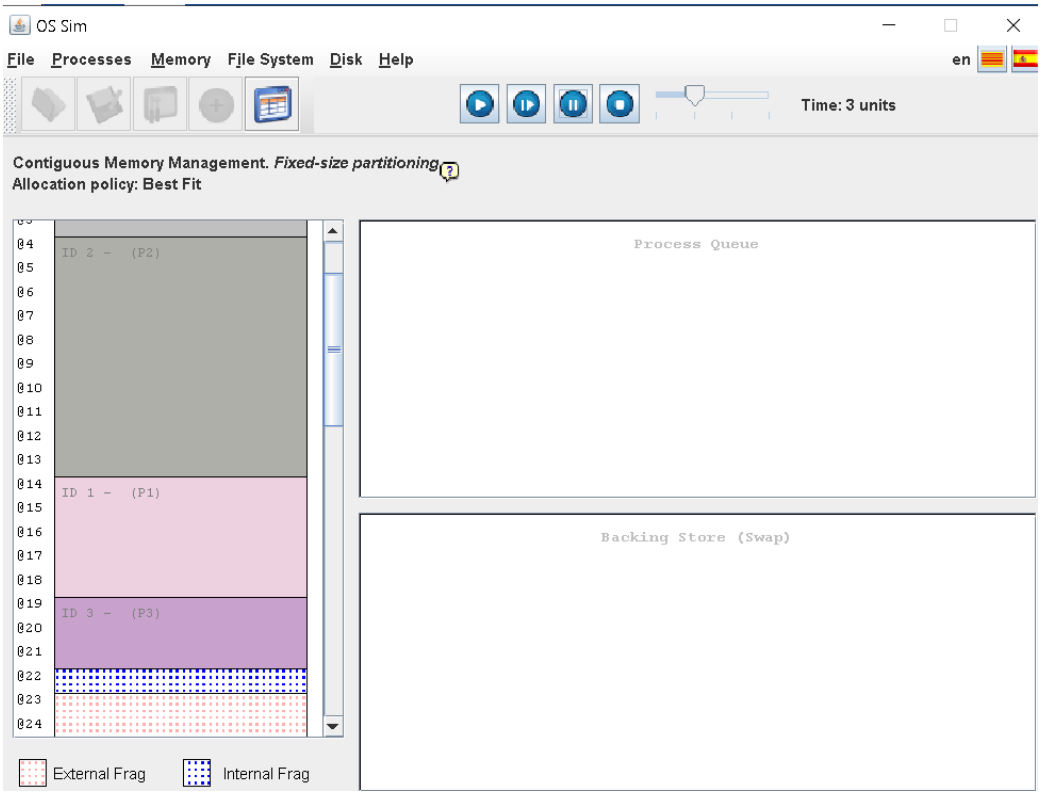
1. Memilih menu help >> examples..>> memory management >> Contiguous memory management with fixed-size partitions (best fit)>> pilih folder



2. Selanjutnya mengeklik tombol play untuk melihat proses yang terjadi dan mengeklik tombol stop untuk berhenti.
3. Melakukan Analisa proses dengan mengeklik pada tombol next.


OS Sim
File Processes Memory File System Disk Help
en
Time: 1 units
Contiguous Memory Management. Fixed-size partitioning
Allocation policy: Best Fit
Main Memory
ID 0 (Operating System)
ID 1 - (P1)
Process Queue
PID 3
P3
Size 3 u.
PID 2
P2
Size 10 u.
Backing Store (Swap)
External Frag
Internal Frag


OS Sim
File Processes Memory File System Disk Help
en
Time: 2 units
Contiguous Memory Management. Fixed-size partitioning
Allocation policy: Best Fit
Main Memory
ID 0 (Operating System)
ID 2 - (P2)
ID 1 - (P1)
Process Queue
PID 3
P3
Size 3 u.
Backing Store (Swap)
External Frag
Internal Frag



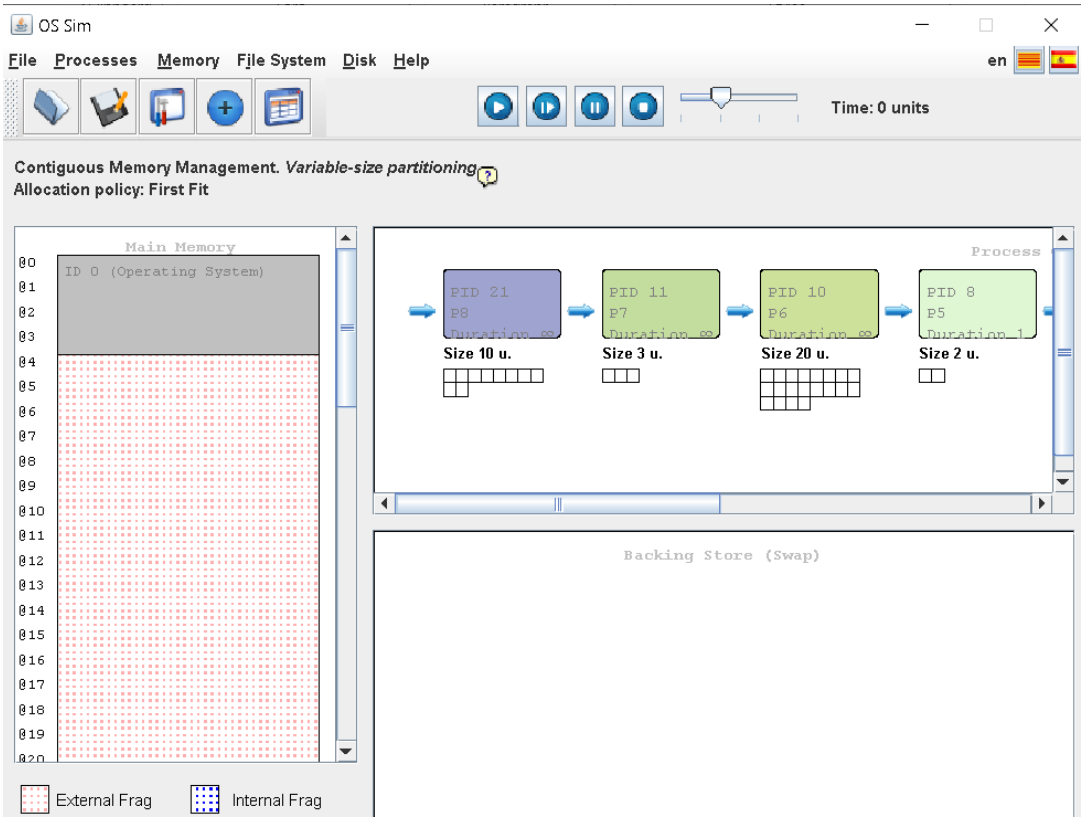
Dengan menggunakan aturan best fit, internal fragmentasi berkurang. Ini terjadi karena proses dimuat pada ukuran partisi terkecil akan tetapi masih dapat memuat proses tersebut sehingga internal fragmentasi dapat diminimalisasi.

2.3 Contiguous memory management dengan menggunakan partisi berukuran tidak tetap (variable-size partition) >> defragmentasi

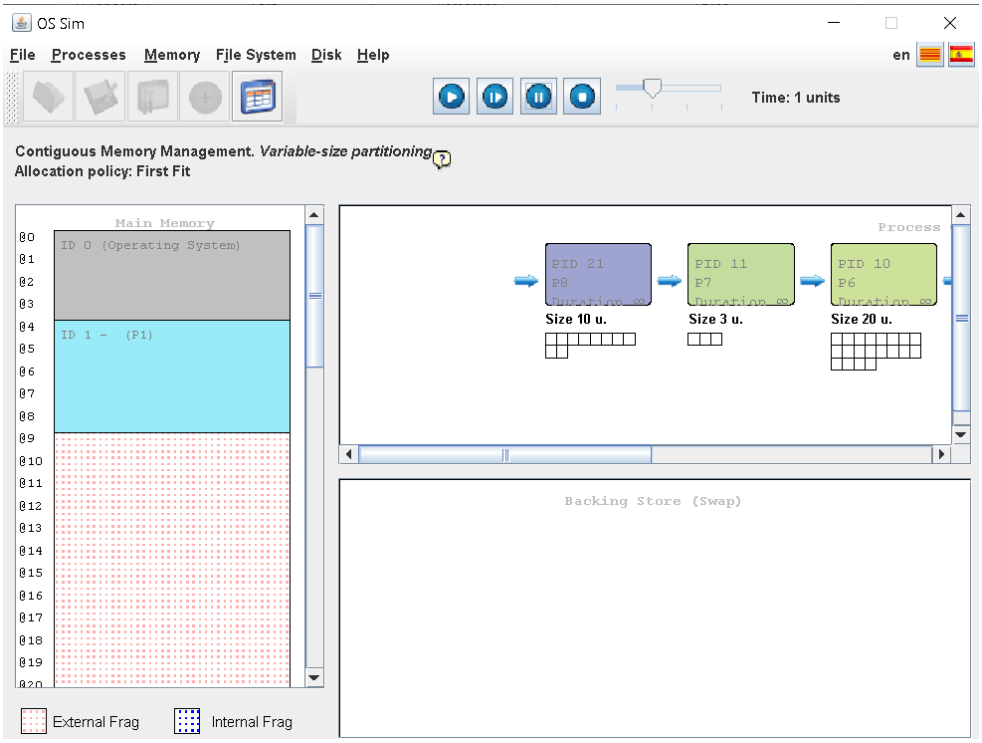
Memori pertama tidak dipartisi, lalu partisi akan dibuat saat proses dialokasikan dan mempunyai ukuran yang sama dengan ukuran proses. Selanjutnya partisi akan dibebaskan (saat proses sudah selesai menggunakan memori) dan digunakan oleh proses lain. Jika ukuran proses lebih kecil dari ukuran partisi, maka partisi akan dibagi menjadi dua bagian, bagian pertama sama besarnya dengan proses dan bagian kedua

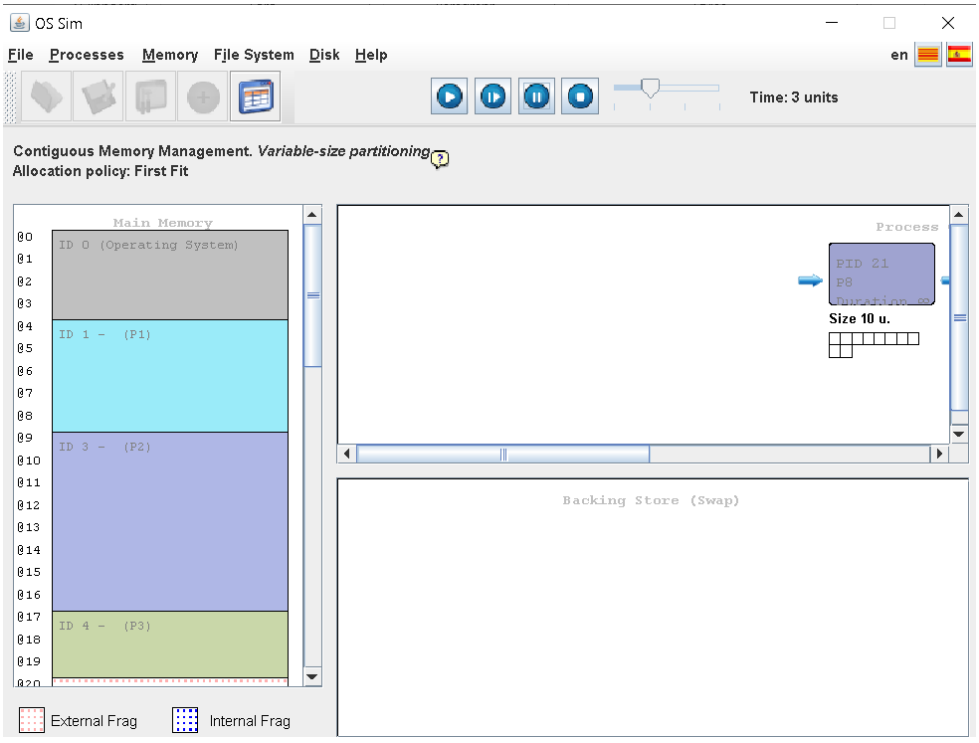
Langkah:

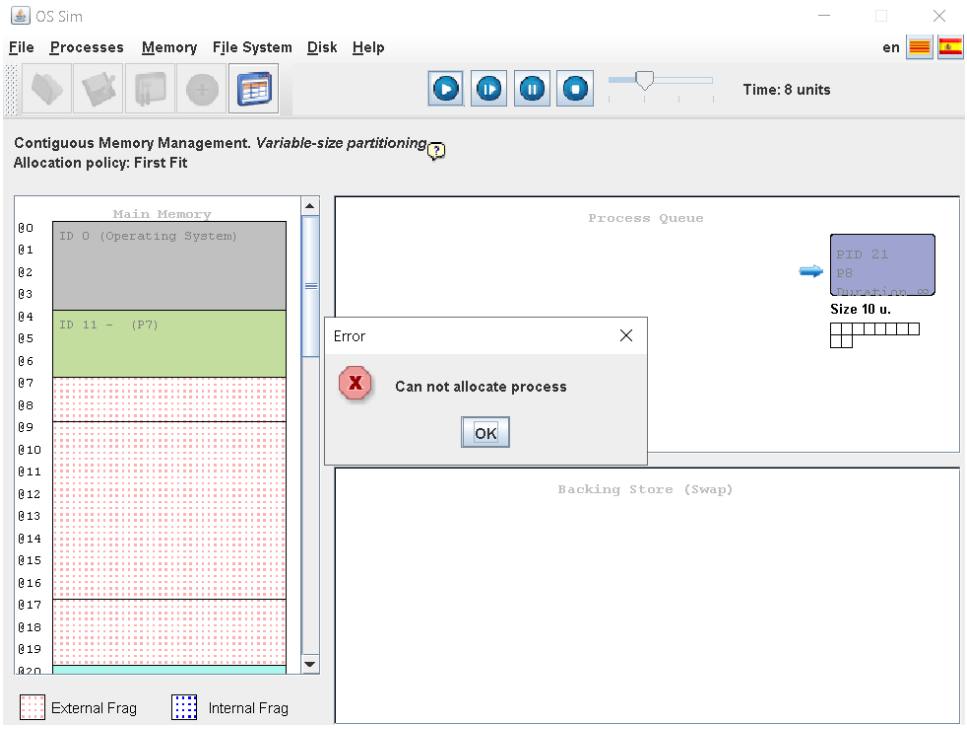
1. Memilih menu help >> examples...>> memory management >>Contiguous memory management with variable-sized partitions (Defragmentation) >> pilih folder



2. Selanjutnya mengeklik tombol play untuk melihat proses yang terjadi dan mengeklik tombol stop untuk berhenti .
3. Melakukan analisa proses dengan melakukan klik pada tombol next.


OS Sim
File Processes Memory File System Disk Help
en
Time: 1 units
Contiguous Memory Management. Variable-size partitioning
Allocation policy: First Fit
Main Memory
ID 0 (Operating System)
ID 1 - (P1)
Process
PID 21
P8
Size 10 u.
PID 11
P7
Size 3 u.
PID 10
P6
Size 20 u.
Backing Store (Swap)
External Frag
Internal Frag


OS Sim
File Processes Memory File System Disk Help
en
Time: 3 units
Contiguous Memory Management. Variable-size partitioning
Allocation policy: First Fit
Main Memory
ID 0 (Operating System)
ID 1 - (P1)
ID 3 - (P2)
ID 4 - (P3)
Process
PID 21
P8
Size 10 u.
Backing Store (Swap)
External Frag
Internal Frag



Pada langkah terakhir terdapat sebuah proses yang tidak dapat dimuat. Ini terjadi karena pembuatan partisi berdasarkan proses yang dimuat pertama kali tetapi tidak ada partisi yang dapat memuat proses terakhir tersebut

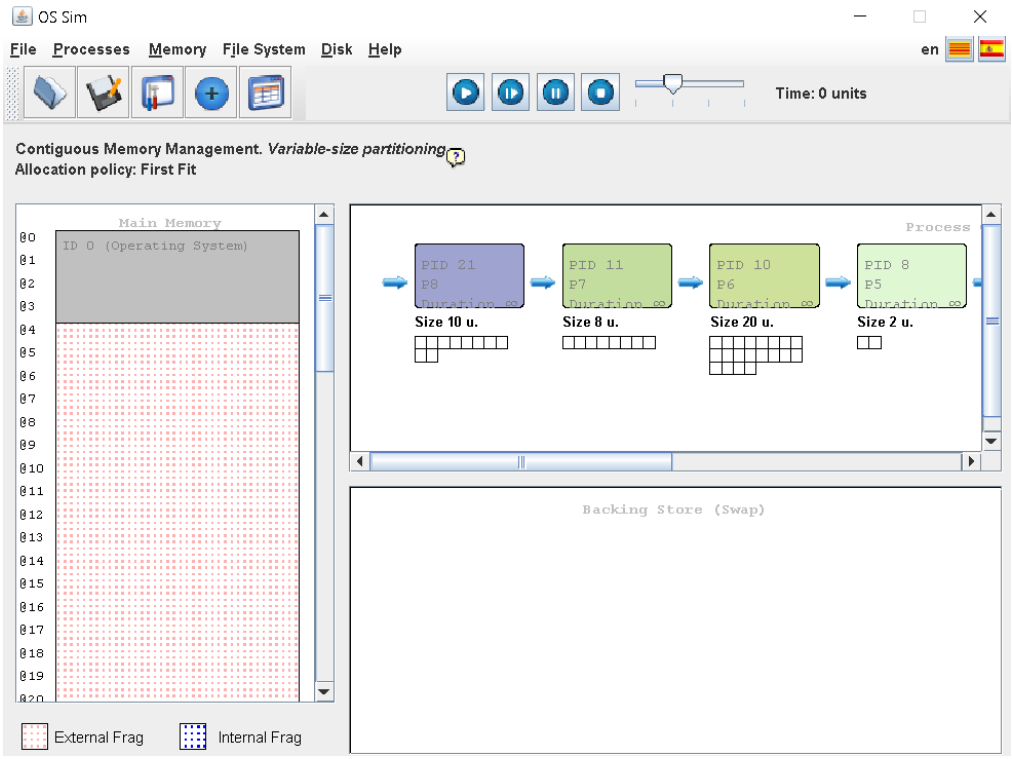
2.4 Contiguous memory management dengan menggunakan partisi berukuran tidak tetap (variable-size partition) >> swap

Memori pertama tidak dipartisi, lalu partisi akan dibuat saat proses dialokasikan dan mempunyai ukuran yang sama dengan ukuran proses. Selanjutnya partisi akan dibebaskan (saat proses sudah selesai menggunakan memori) dan digunakan oleh proses lain. Jika ukuran proses lebih kecil dari ukuran partisi, maka partisi akan dibagi menjadi dua bagian, bagian pertama sama besarnya dengan proses dan bagian kedua adalah sisanya.

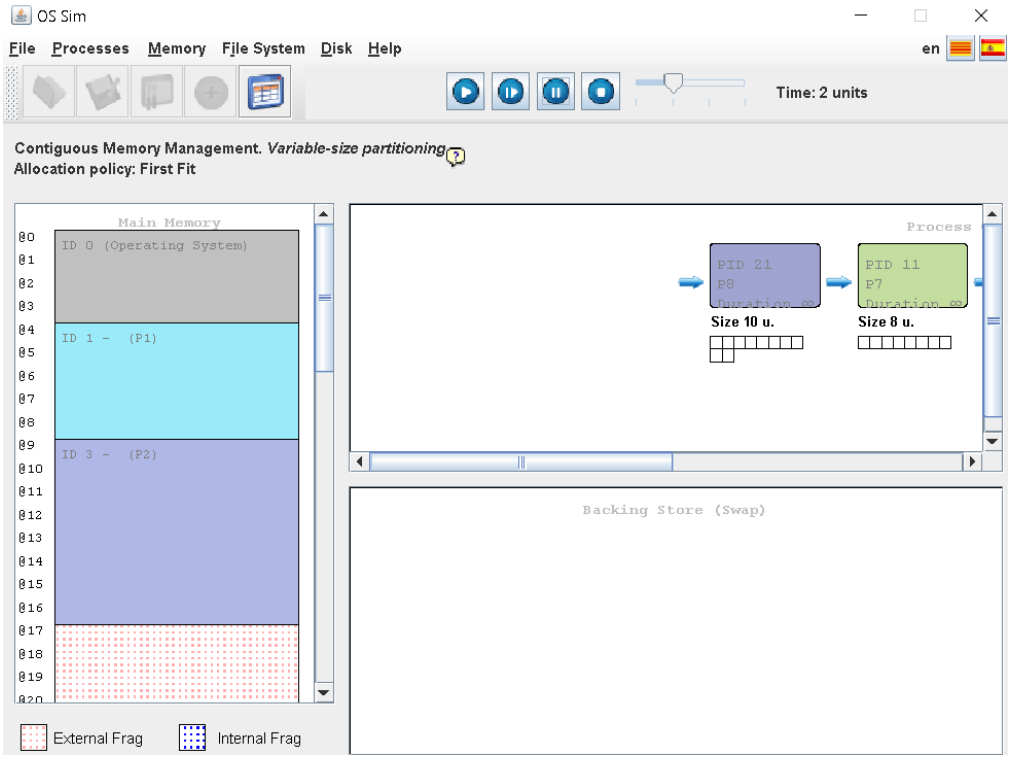
Memori mempunyai ukuran yang berhingga, hanya beberapa proses saja yang bisa dimuat di memori pada saat yang sama, tetapi space swap memungkinkan kita untuk menyimpan beberapa proses yang sedang kurang aktif, sehingga space pada memori dapat dialokasikan untuk proses yang lain yang lebih aktif. Proses yang telah di swap out akan dialokasikan kembali ke memori saat dibutuhkan.

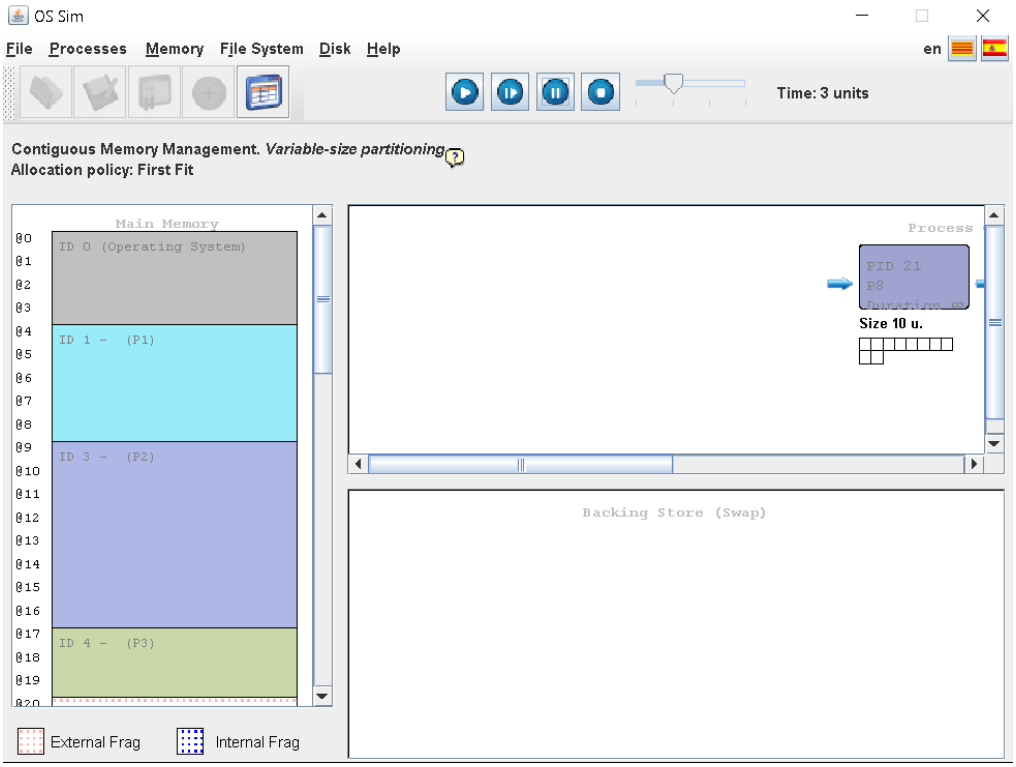
Langkah:

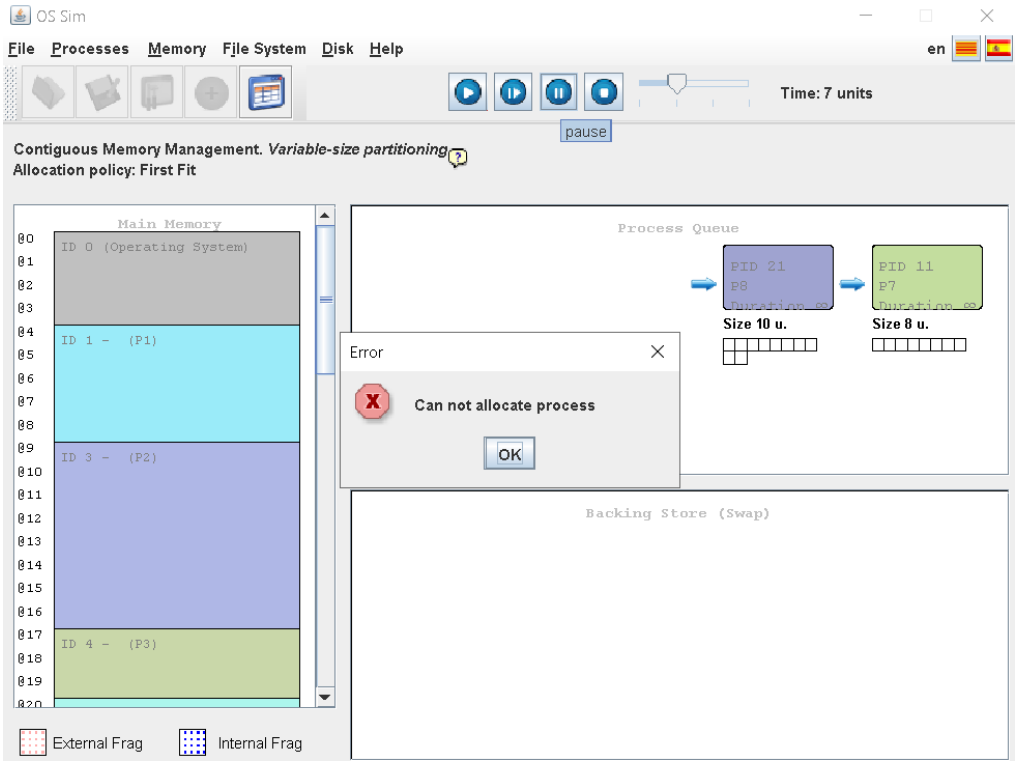
1. Memilih menu help >> examples...>> memory management >>Contiguous memory management with variable-sized partitions (Swap) >> pilih folder



2. Selanjutnya mengeklik tombol play untuk melihat proses yang terjadi dan mengeklik tombol stop untuk berhenti.
3. Melakukan Analisa proses dengan melakukan klik pada tombol next.







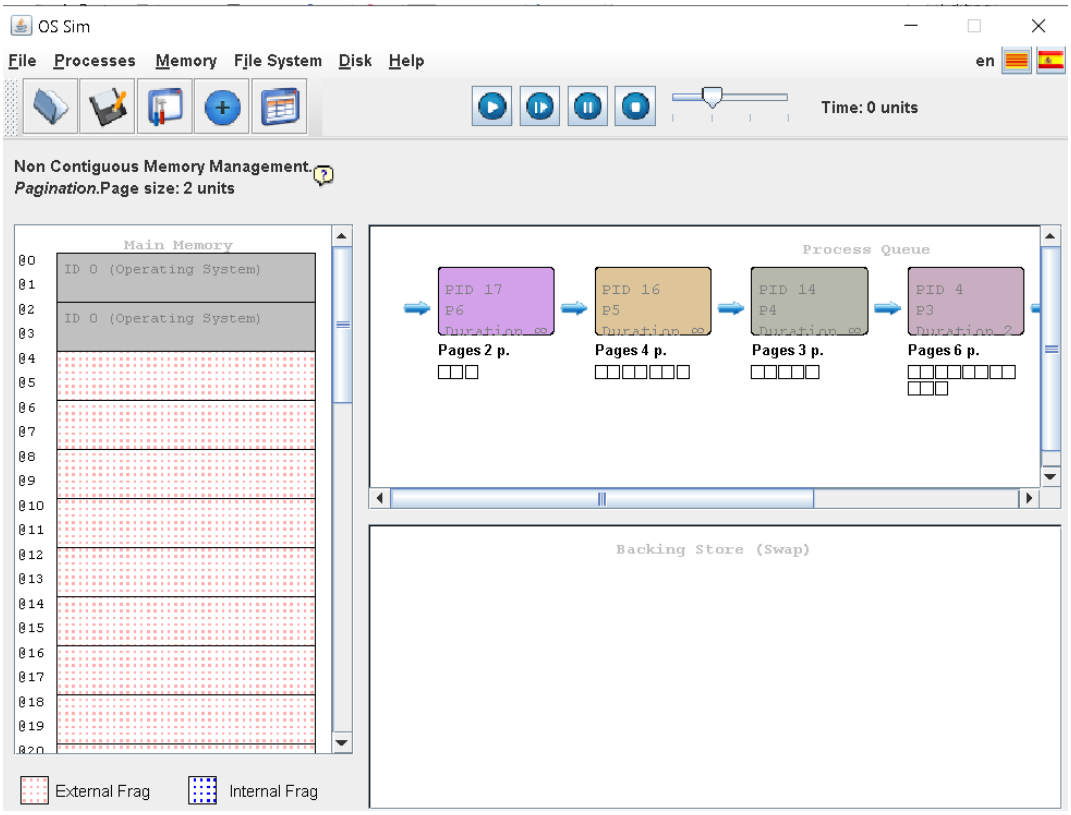
Sampai langkah ke 7 akan ada notifikasi Can not allocate process. Hal ini mungkin terjadi karena tidak adanya proses yang tidak aktif sehingg tidak terjadi swap dan partisi tidakada yang mencukupi untuk memuat proses yang tersisa.

2.5 Pagination (ukuran page 2 unit)

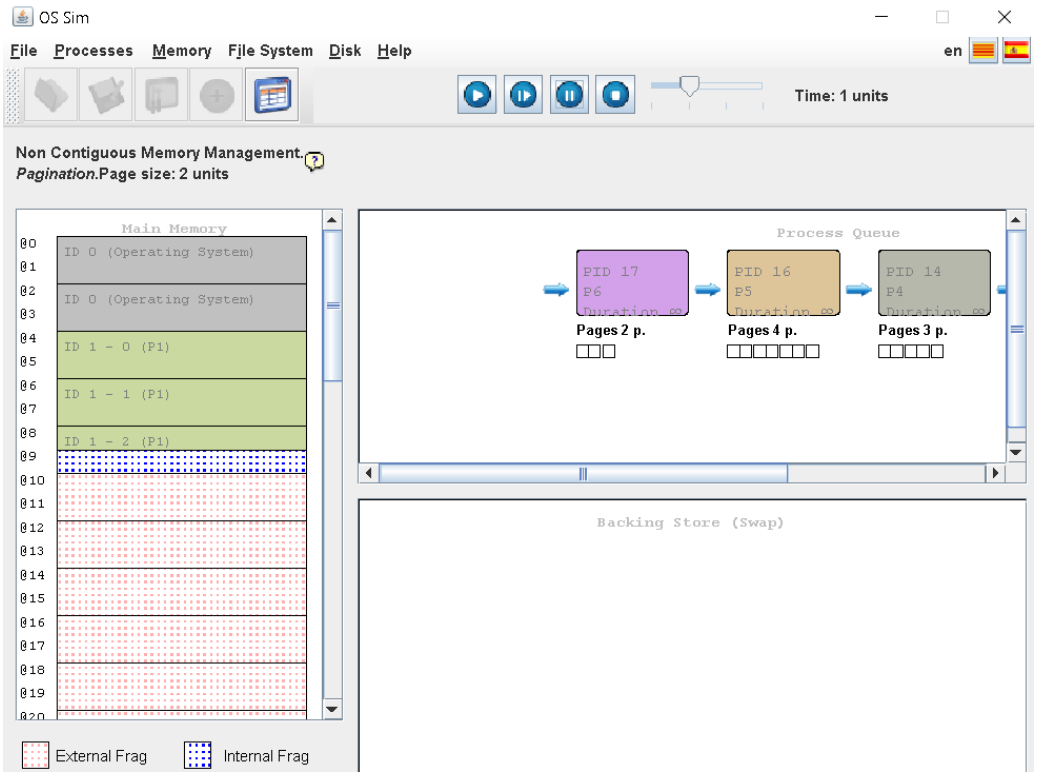
Proses akan dibagi menjadi beberapa page dengan ukuran tetap (contohnya adalah 2 unit), memori lalu dibagi menjadi beberapa frame berukuran sama. Page dari proses lalu dialokasikan pada frame memori yang kosong.

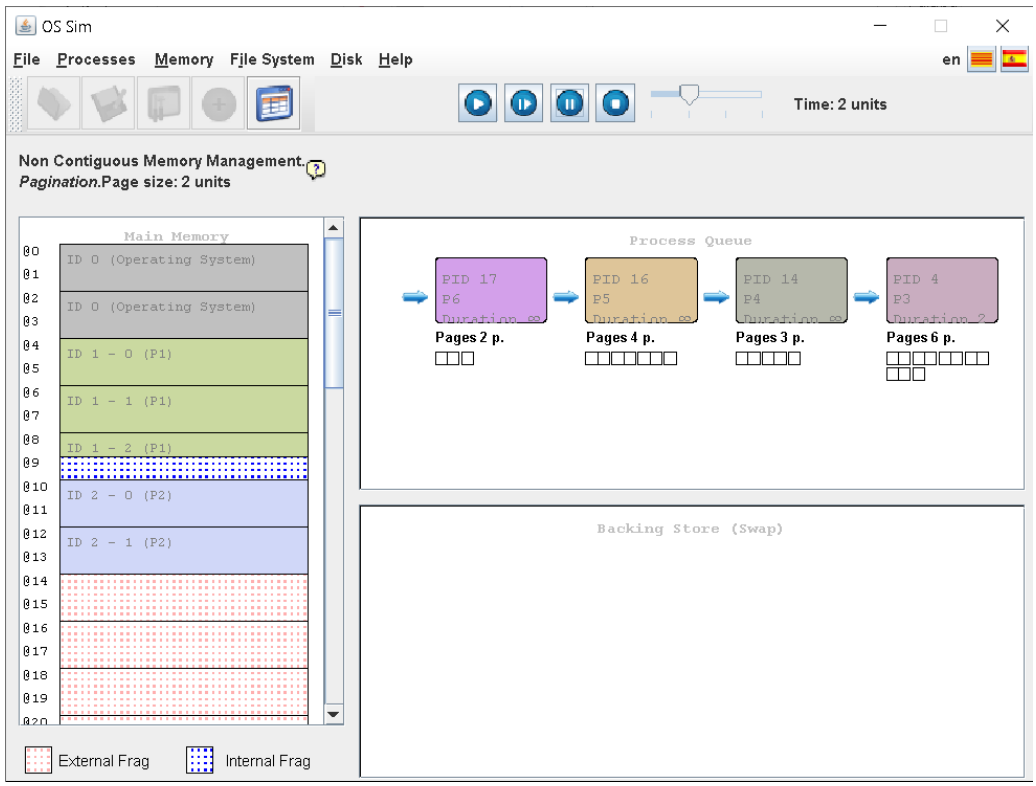
Langkah:

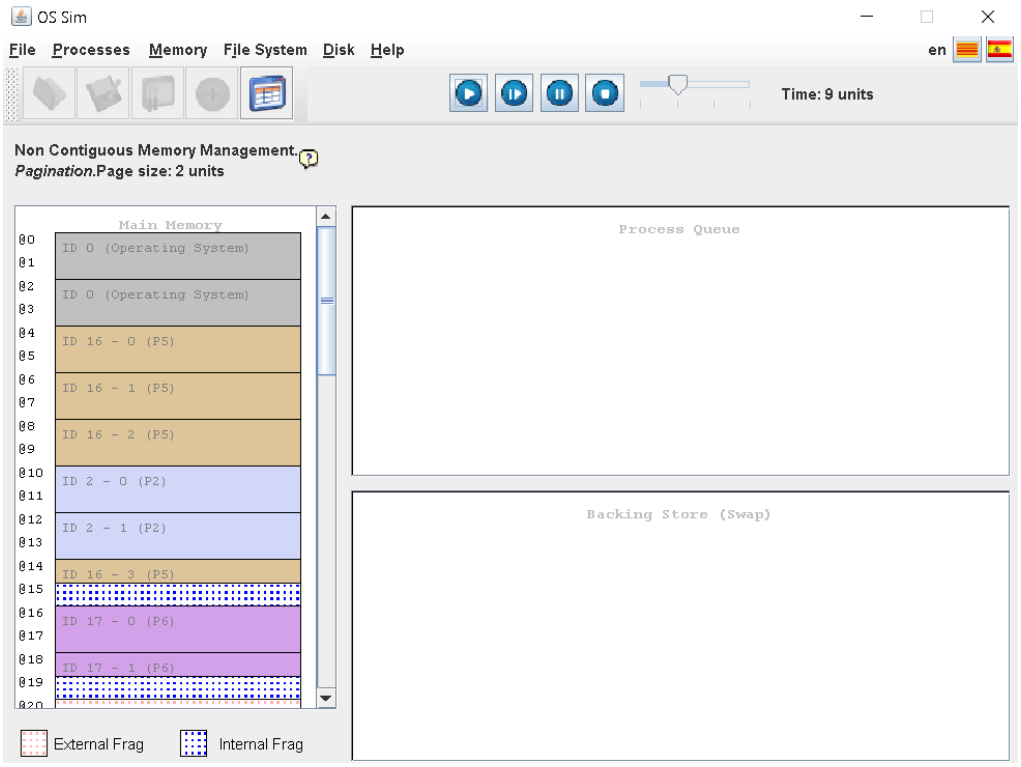
1. Memilih menu help >> examples...>> memory management >>Pagination) >> pilih folder



2. Selanjutnya mengeklik tombol play untuk melihat proses yang terjadi dan mengeklik tombol stop untuk berhenti .
3. Melakukan analisa proses dengan melakukan klik pada tombol next.


OS Sim
File Processes Memory File System Disk Help
en
Time: 1 units
Non Contiguous Memory Management.
Pagination.Page size: 2 units
Main Memory
@0 ID 0 (Operating System)
@2 ID 0 (Operating System)
@4 ID 1 - 0 (P1)
@6 ID 1 - 1 (P1)
@8 ID 1 - 2 (P1)
@9
@10
@11
@12
@13
@14
@15
@16
@17
@18
@19
@20
External Frag
Internal Frag
Process Queue
PID 17
P6
Pages 2 p.
PID 16
P5
Pages 4 p.
PID 14
P4
Pages 3 p.
Backing Store (Swap)


OS Sim
File Processes Memory File System Disk Help
en
Time: 2 units
Non Contiguous Memory Management.
Pagination.Page size: 2 units
Main Memory
@0 ID 0 (Operating System)
@2 ID 0 (Operating System)
@4 ID 1 - 0 (P1)
@6 ID 1 - 1 (P1)
@8 ID 1 - 2 (P1)
@9
@10 ID 2 - 0 (P2)
@11
@12 ID 2 - 1 (P2)
@13
@14
@15
@16
@17
@18
@19
@20
External Frag
Internal Frag
Process Queue
PID 17
P6
Pages 2 p.
PID 16
P5
Pages 4 p.
PID 14
P4
Pages 3 p.
PID 4
P3
Pages 6 p.
Backing Store (Swap)



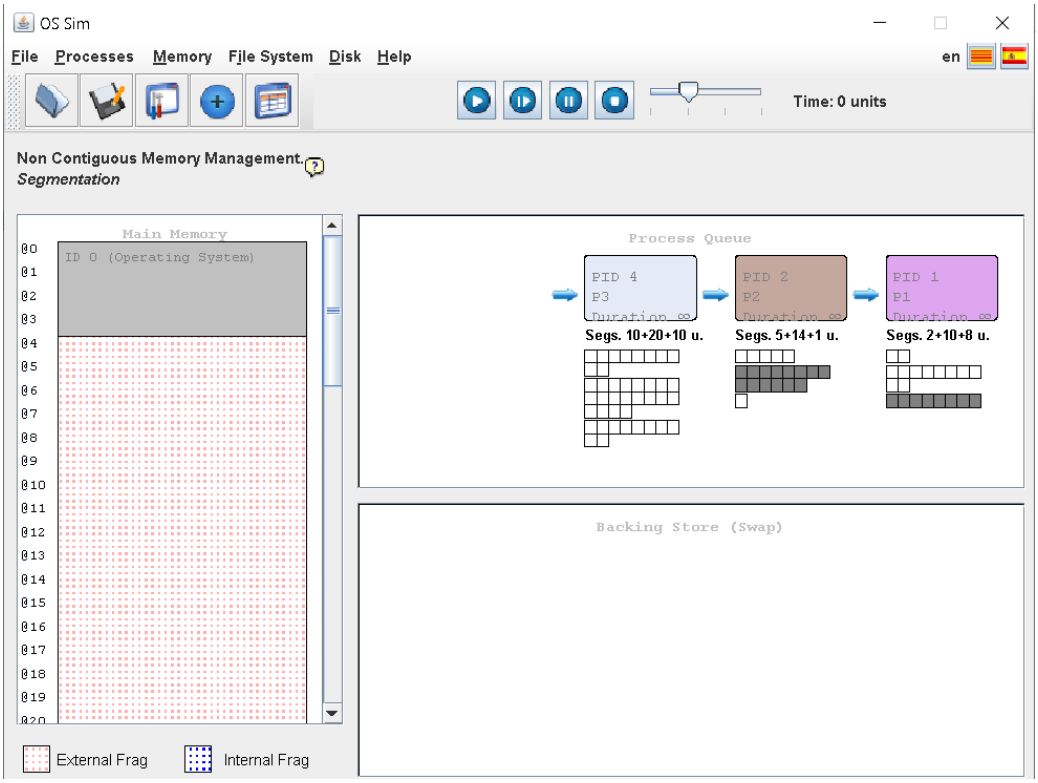
Terlihat pada percobaan ini bahwa setiap proses dibagi menjadi beberapa page denganukuran tetap setiap pagenya, yaitu 2 unit.

2.6 Segmentation (alokasi parsial)

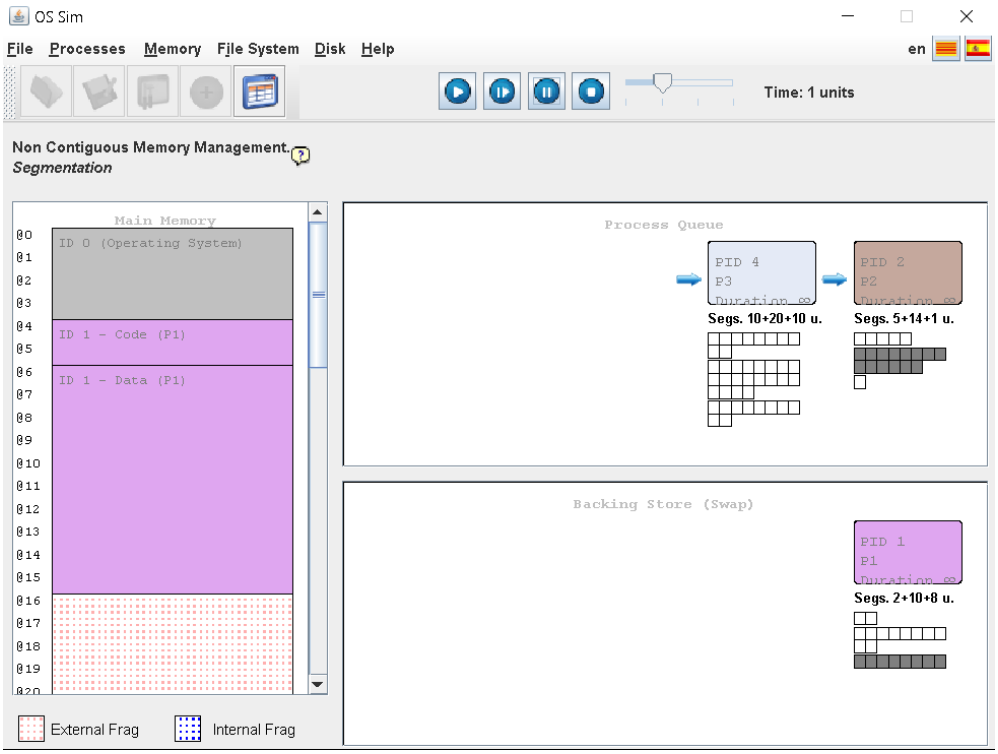
Proses dibagi menjadi beberapa segmen, tiap segmen bisa mempunyai ukuran yang berbeda. Segmen ini akan dialokasikan ke memori secara independen, lalu partisi akan dibuat untuk menampung segmen tersebut dengan ukuran yang sama. Pembagian ini adalah secara fungsional dan berdasarkan pada logical structure dari proses tersebut (data, stack, code, dll). Tidak semua segmen harus dimuat pada memori agar proses dapat berjalan, selama sebuah segmen tidak dipakai maka dapat di-swap out

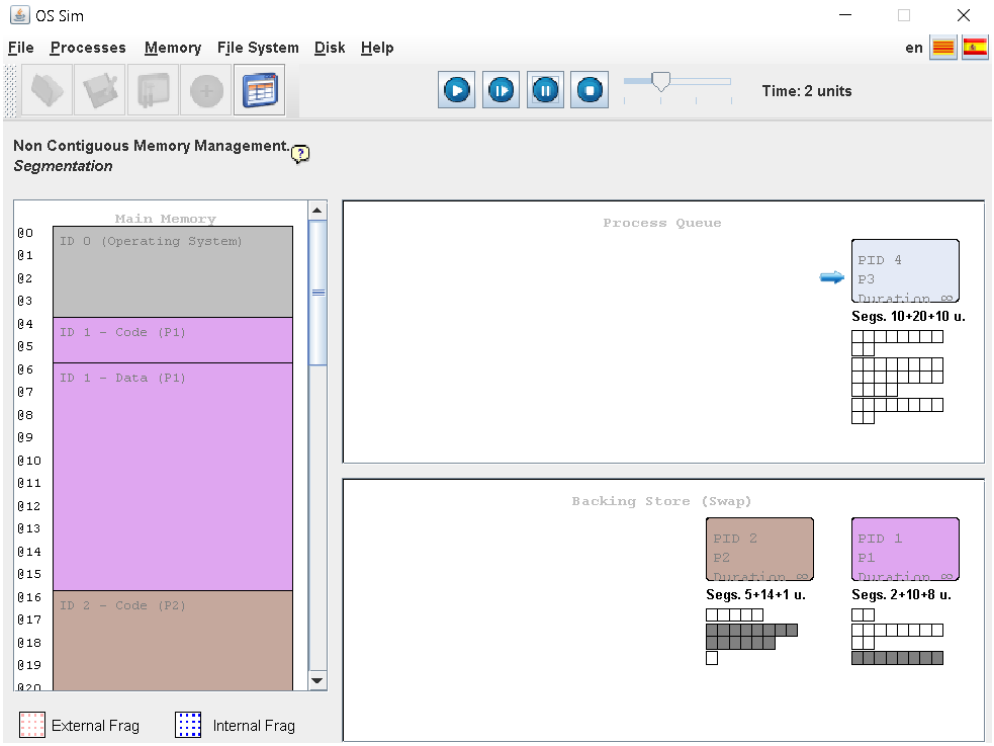
Langkah:

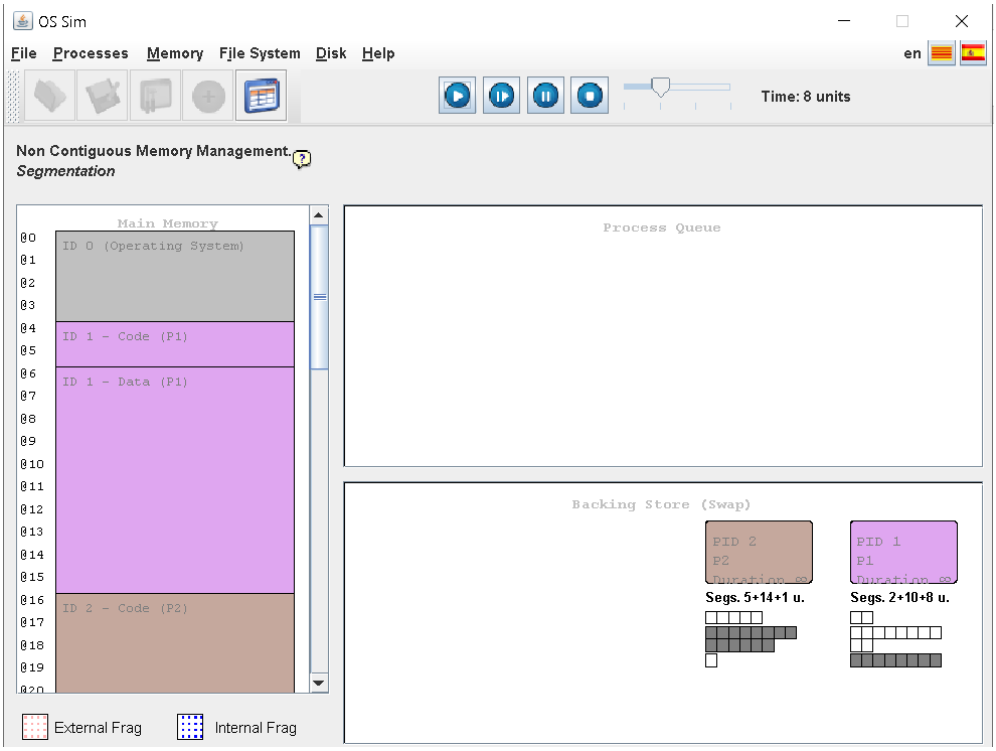
1. Memilih menu help >> examples...>> memory management >>Segmentasi >> pilih folder



2. Selanjutnya mengeklik tombol play untuk melihat proses yang terjadi dan mengeklik tombol stop untuk berhenti.
3. Lakukan analisa proses dengan melakukan klik pada tombol next.







Semua proses dapat dimuat, karena setiap proses dibagi menjadi beberapa segmen berdasarkan logical structure-nya. Dan karena tidak semua segmen proses tersebut dimuat di memori, semua proses dapat dimuat. Dan yang segmen yang tidak dimuat tadi dapat di swap-out