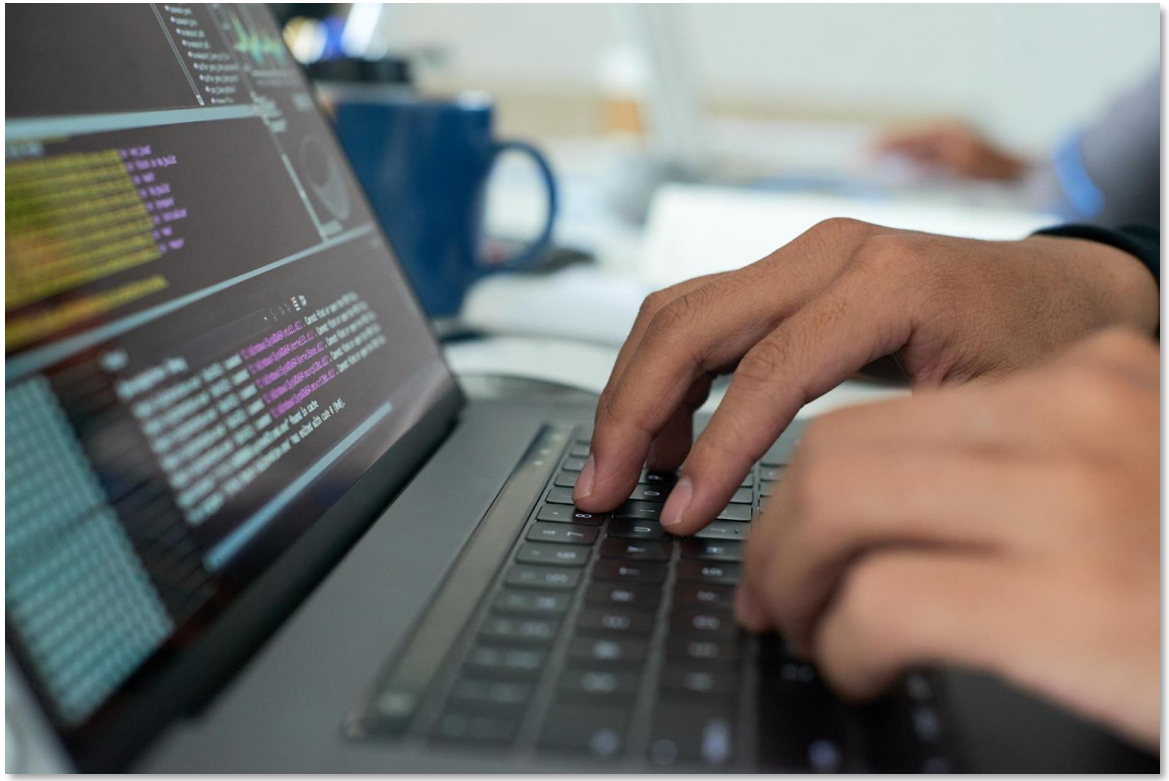
U.T.2_2. CPU Scheduling



Sistemas Microinformáticos y Redes

Sistemas Operativos Monopuesto Bilingüe

Scheduling Processor Algorithms

- The states in which a process can be are:
  - Ready: The process is ready to be executed, awaiting for free CPU resources.
  - Blocked: The process is waiting for a resource that other process is using.
  - Executing: The process is using the CPU.

Scheduling Processor Algorithms

- We will know next information for the studied algorithms:
  - Input Time (T_I) (llegada al sistema)
  - Execution Time (T_x): (Total time that the algorithm needs)
- With this data, we can calculate:
  - Response Time (T_R): Time while the process reaches the system, until it finishes.
  - Awaiting Time (T_E): Time that the process is waiting for a resource.

$$T_E = T_R - T_X$$

Algorithm
FIFO (First In, First Out)

- The first process which reaches the system, is the first one in finishing it tasks.

Algorithm
FIFO (First In, First Out)

FIFO example

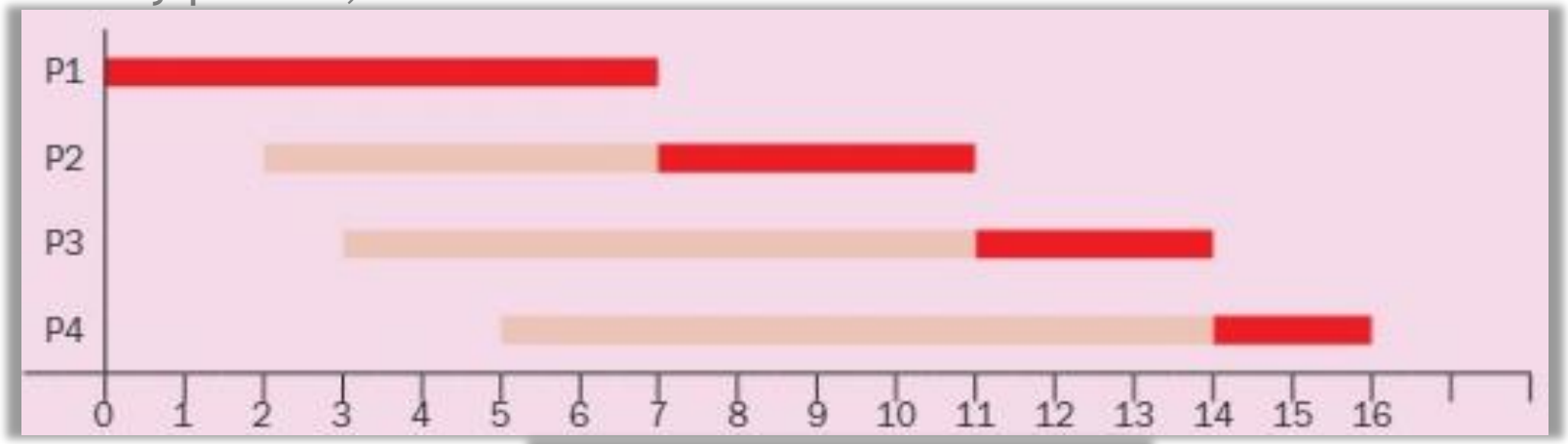
Having the next processes, with the arriving and execution time indicated, calculate the awaiting time and average time, using FIFO Algorithm of CPU Scheduling.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algorithm
FIFO (First In, First Out)

FIFO example

We have to draw the time-chart, to see in pink color the awaiting time of every process, and in red color the execution time of the CPU.



PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	5	9
P3	8	11
P4	9	11
TIEMPOS MEDIOS	5,5	9,5

Algorithm
SJF (Short Job First)

- This algorithm chooses the shortest process among all which are waiting to use the CPU. In case of tie/draw, FIFO algorithm is applied.
- It's better for the processes which take less time to be executed..

Algorithm
SJF (Short Job First)

- ***SJF Example***

  Having the next processes, when the arriving and execution time indicated, calculate the awaiting time and average time, using SJF Algorithm of CPU Scheduling.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algorithm
SJF (Short Job First)

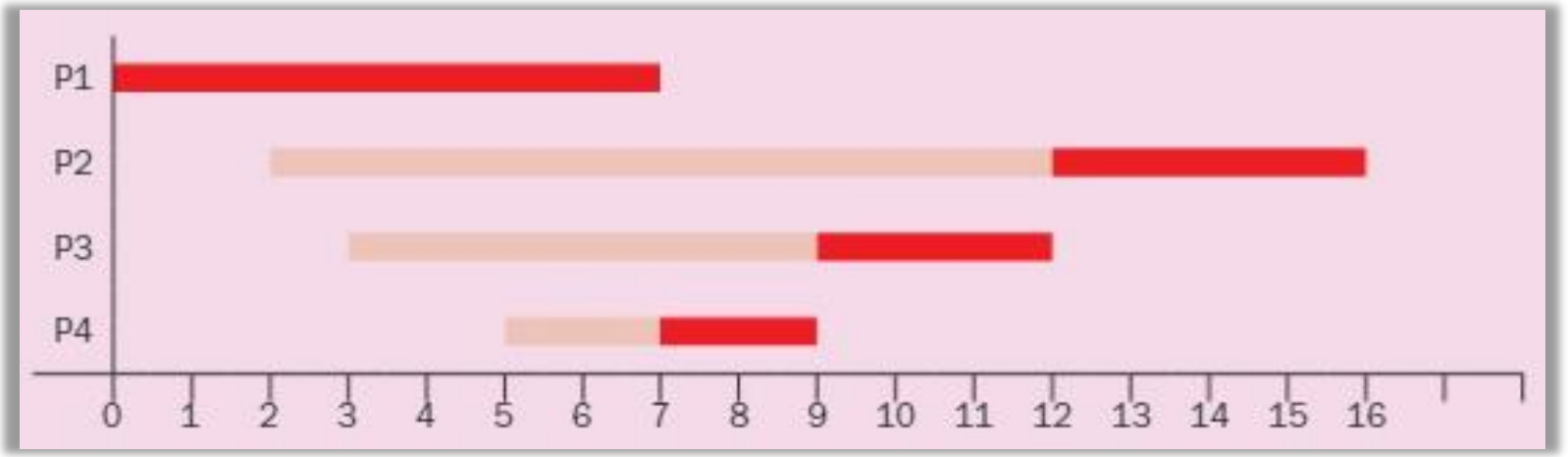
- ***SJF Example***

We have to draw the time-chart, to see in pink color the awaiting time of every process, and in red color the execution time of the CPU.

The response time will be the time between the moment when process arrived and the moment when the process finished.

Algorithm
SJF (Short Job First)

- ***SJF Example***



PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	10	14
P3	6	9
P4	2	4
TIEMPOS MEDIOS	4,5	8,5

Algorithm
SRTF (Short Remaining Time First)

- This algotirithm chooses among the awaiting processes, the one which takes less time to finish. In case of tie/draw, FIFO algorithm applies.
- This is a preemptive algorithm, so it can produce a change of context, every time that a new process arrives.
- It's better the short processes.

Algorithm
SRTF (Short Remaining Time First)

- **SRTF Example**

  Having the next processes, with the arriving and execution time indicated, calculate the awaiting time and average time, using SRTF Algorithm of CPU Scheduling.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

Algorithm
SRTF (Short Remaining Time First)

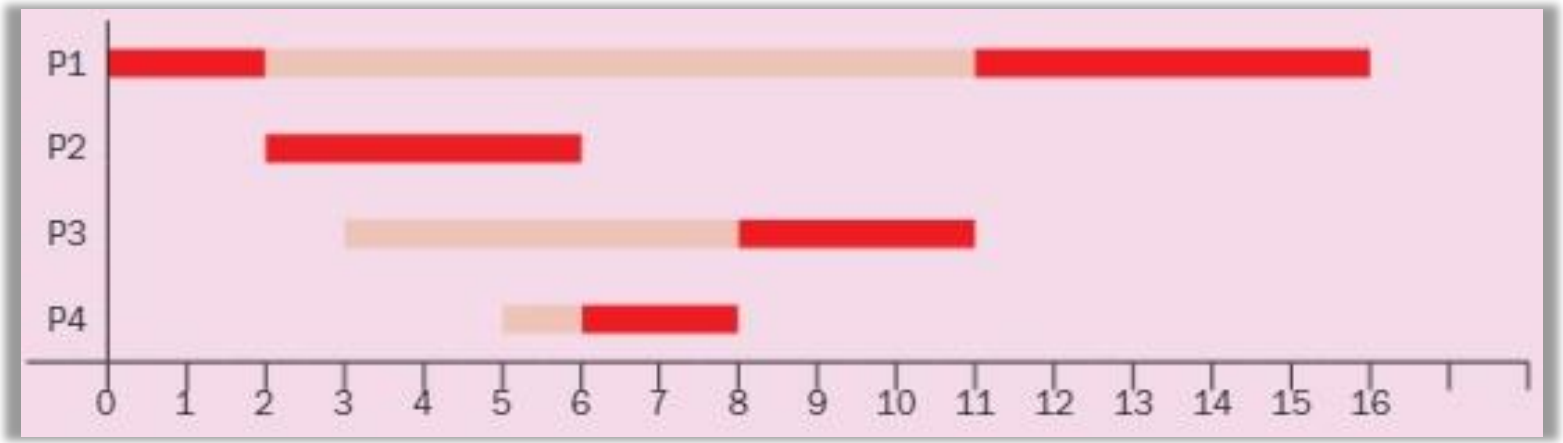
- **Ejemplo SRTF**

We have to draw the time-chart, to see in pink color the awaiting time of every process, and in red color the execution time of the CPU.

The response time will be the time between the moment when process arrived and the moment when the process finished.

Algorithm
SRTF (Short Remaining Time First)

- **Ejemplo SRTF**



PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	0	4
P3	5	8
P4	1	3
TIEMPOS MEDIOS	3,75	7,75

Algorithm based on priorities

- This algorithm consists in associate a priority to each process.

  In case of tie/draw of priority, algorithm FIFO applies.

- There are two different versions of this algorith, preemptive and non-preemptive version.

Algorithm based on priorities

- Priority of each process is indicated in an integer number. The lowest the number is, the higest is the priority.
- There can be several processess with the same priority.
- This algorithm is better for longer processes with high priority.

Algorithm based on priorities

- **Non-Preemptive Priority Based Algorithm Example:**

  Having the next processes, with the arriving and execution time indicated, calculate the awaiting time and average time, using priority-based Algorithm of CPU Scheduling in its **non-preemptive version**.

PROCESO	TIEMPO DE LLEGADA	PRIORIDAD	TIEMPO DE EJECUCIÓN
P1	0	4	7
P2	2	2	4
P3	3	1	3
P4	5	3	2

Algorithm based on priorities

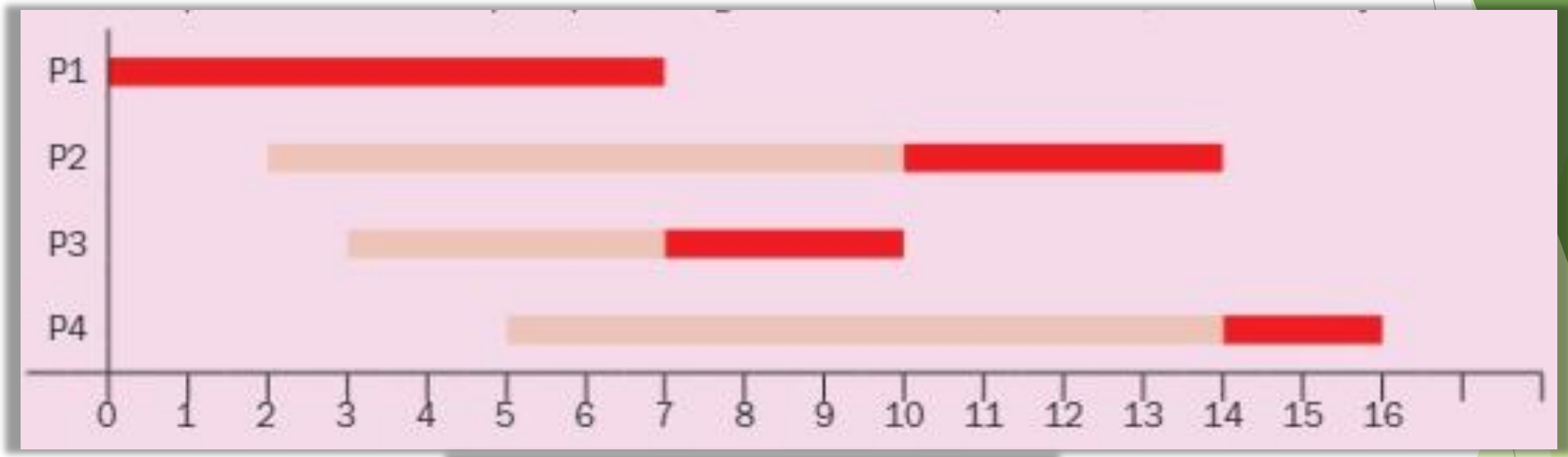
- **Non-Preemptive Priority Based Algorithm Example:**

We have to draw the time-chart, to see in pink color the awaiting time of every process, and in red color the execution time of the CPU.

The response time will be the time between the moment when process arrived and the moment when the process finished.

Algorithm based on priorities

- Non-Preemptive Priority Based Algorithm Example:



PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	0	7
P2	8	12
P3	4	7
P4	9	11
TIEMPOS MEDIOS	5,25	9,25

Algorithm based on priorities

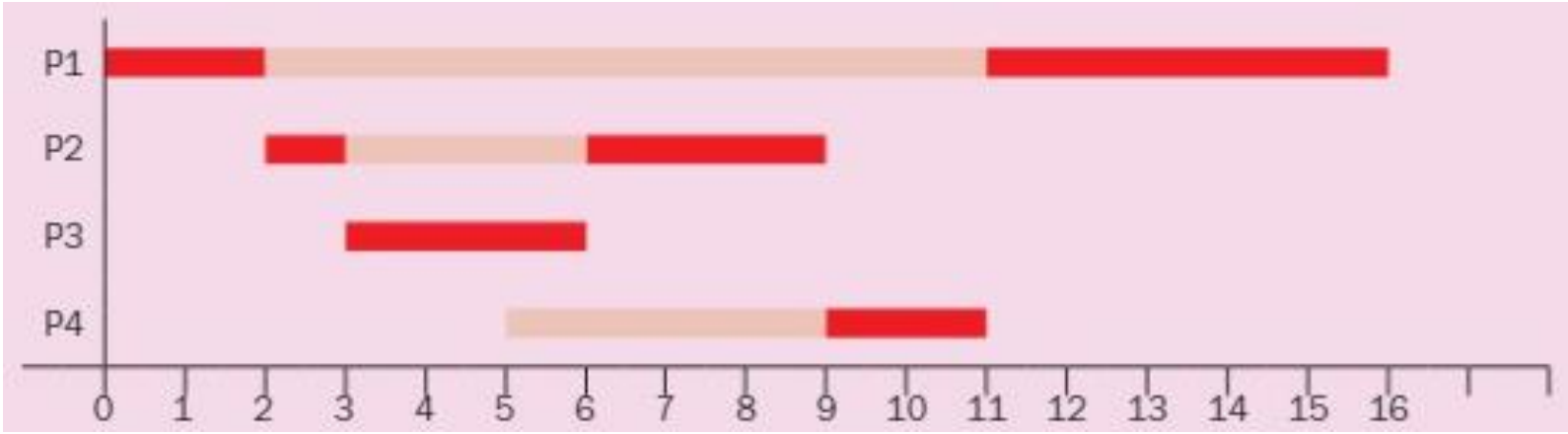
- **Preemptive Priority Based Algorithm Example:**

Having the next processes, with the arriving and execution time indicated, calculate the awaiting time and average time, using priority-based Algorithm of CPU Scheduling in its ***preemptive version***.

PROCESO	TIEMPO DE LLEGADA	PRIORIDAD	TIEMPO DE EJECUCIÓN
P1	0	4	7
P2	2	2	4
P3	3	1	3
P4	5	3	2

Algoritmo por Prioridades

- **Preemptive Priority Based Algorithm Example:**



PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	3	7
P3	0	3
P4	4	6
TIEMPOS MEDIOS	4	8

RR (Round Robin CPU Algorithm)

- This algorithm gives the same execution time to all the processes (Quantum). In case of tie/draw, FIFO applies.
- It needs that the change of context is very fast, because it happens several times.
- This is a preeptive algorithm.

RR (Round Robin CPU Algorithm)

- **Algoritmo Round Robin**

Having the next processes, with the arriving and execution time indicated, calculate the awaiting time and average time, using Round Robin Algorithm with quantum=2.

PROCESO	TIEMPO DE LLEGADA	TIEMPO DE EJECUCIÓN
P1	0	7
P2	2	4
P3	3	3
P4	5	2

RR (Round Robin CPU Algorithm)

- **Round Robin Algorithm Example.**

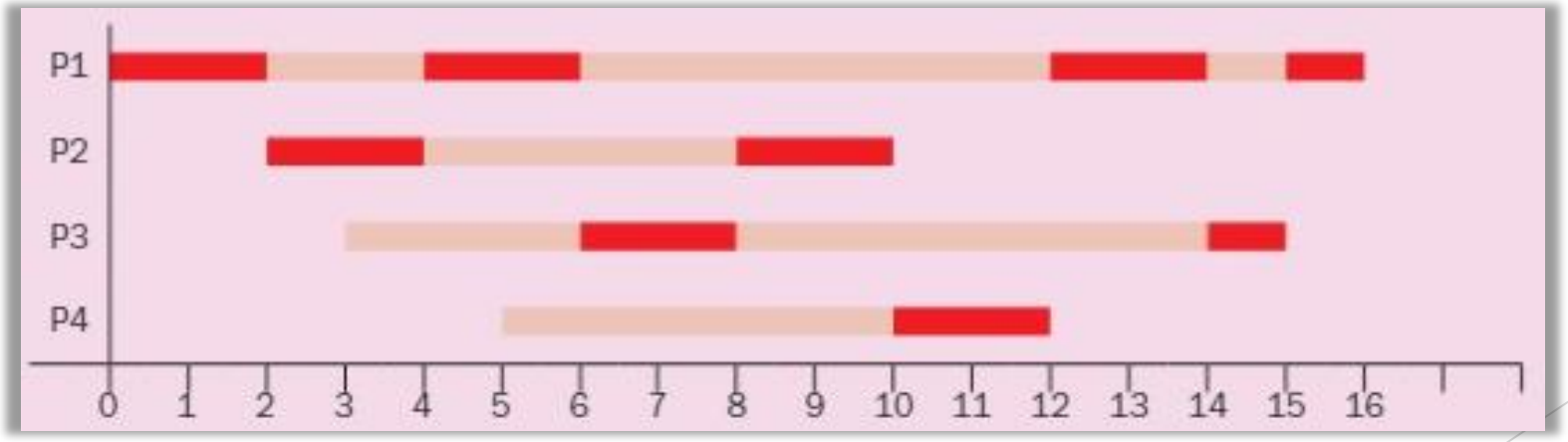
We have to draw the time-chart, to see in pink color the awaiting time of every process, and in red color the execution time of the CPU.

The response time will be the time between the moment when process arrived and the moment when the process finished.

RR (Round Robin CPU Algorithm)

- **Round Robin Algorithm Example.**

This is the sequence of context´s changes during the execution of the processes P1, P2, P3 and P4.



RR (Round Robin CPU Algorithm)

- Round Robin Algorithm Example Final Result.

PROCESO	TIEMPO DE ESPERA	TIEMPO DE RESPUESTA
P1	9	16
P2	4	8
P3	9	12
P4	5	7
TIEMPOS MEDIOS	6,75	10,75