**[Problemas de Sistemas Operativos: de la base al diseño](http://arcos.inf.uc3m.es/%7essoo-va/" \t "_blank" \o "http://arcos.inf.uc3m.es/%7essoo-va/)  
Jesús Carretero, Félix García y Fernando Pérez. Amazon. 2015.**

# Exercise 1.

Given the next set of processes:

|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | **Arrival** | **CPU Time** | **Priority** |
| **P1** | 0 | 10 | 3 |
| **P2** | 1 | 1 | 1 |
| **P3** | 3 | 4 | 3 |
| **P4** | 4 | 2 | 4 |
| **P5** | 5 | 5 | 2 |

a) Write a diagram that illustrates the execution of these processes using:

1. FIFO.

2. Scheduling with preemptive (or expulsive) priorities

3. Scheduling with preemptive priorities and with Round Robin (q = 2) for the processes of the same priority. (If the execution slice of a process ends at the same instant that a new process arrives on the system, then the new process is placed in the ready-to-run queue before the process that expires the slice.)

b) Calculate the waiting time for each scheduling process and algorithm.

***SOLUTION***

1. FIFO.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
| P3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| P4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| P5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

1. Scheduling with preemptive priorities.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| P2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| P4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

* En el instante t=10, se pone a ejecutar P3 porque estaba a la cola de listos antes que P1.

1. Scheduling with preemptive priorities and Round Robin (q=2) for equal priority processes.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| P2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| P4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

* In time t=10, P1 is run as it was in the ready queue before P3.

1. **.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** | **P5** |
| **Fifo** | 0 | 9 | 8 | 11 | 12 |
| **Priorities** | 10 | 0 | 7 | 16 | 0 |
| **Priorities and Round Robin** | 10 | 0 | 10 | 16 | 0 |

# Exercise 2

In a given operating system, process scheduling is carried out according to priority. The system has three priority levels (1, 2 and 3), with 1 being the highest priority level and 3 the least priority. Within each priority level, round-robin scheduling is performed with a 200 ms time slice.

Furthermore, an aging algorithm is used to avoid starvation of the less priority processes. If a process remains 600 ms. in a queue of a certain priority level without executing, it is passed to the next higher queue. When that process is kicked out of the CPU, it returns to its original priority queue. If applied to a process that is in the highest priority queue, the aging algorithm is admitted that its priority can be even higher (negative number of priority is allowed).

The following table specifies a set of processes with their priority, their arrival time and the total time required for their execution.

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Priority | Arrival T. | Execution T. |
| P1 | 3 | 100 | 600 |
| P2 | 2 | 200 | 500 |
| P3 | 1 | 300 | 200 |
| P4 | 1 | 400 | 300 |
| P5 | 2 | 500 | 700 |

Calculate:

a) The time that each process remains on hold from its arrival in the system until it ends.

b) The return time of each process (time elapsed since the process arrives until its execution ends).

c) Average waiting time and average return time.

Repeat the calculations in case the policy used in each priority queue is FIFO.

Repeat the calculations in case the policy used in each priority queue is FIFO.

***SOLUTION***

The following table expresses the evolution of the processes in the CPU and in the queues. Each process is expressed by 4 values ​​(pe, pr, tr, te), where pe represents the effective priority, pr the actual priority, tr the remaining time and te the aging time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | **CPU** | **Q1** | **Q2** | **Q3** |
| 100 | P1(3,3,600,0) |  |  |  |
| 200 | P2(2,2,500,0) |  |  | P1(3,3,500,0) |
| 300 | P3(1,1,200,0) |  | P2(2,2,400,0) | P1(3,3,500,100) |
| 400 | P3(1,1,100,0) | P4(1,1,300,0) | P2(2,2,400,100) | P1(3,3,500,200) |
| 500 | P3->FIN  P4(1,1,300,0) |  | P2(2,2,400,200)  P5(2,2,700,0) | P1(3,3,500,300) |
| 800 | P4->FIN  P2(2,2,400,0) |  | P5(2,2,700,300)  P1(2,3,500,0) |  |
| 1000 | P5(2,2,700,0) |  | P1(2,3,500,200)  P2(2,2,200,0) |  |
| 1200 | P1(2,3,500,0) |  | P2(2,2,200,200)  P5(2,2,500,0) |  |
| 1400 | P2(2,2,200,0) |  | P5(2,2,500,0) | P1(3,3,300,0) |
| 1600 | P2->FIN  P5(2,2,500,0) |  |  | P1(3,3,300,200) |
| 1800 | P5(2,2,300,0) |  |  | P1(3,3,300,400) |
| 2000 | P5(2,2,100,0) |  | P1(3,3,300,0) |  |
| 2100 | P5-> FIN  P1(3,3,300,0) |  |  |  |
| 2400 | P1-> FIN |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Proces** | **Tfin** | **Tret** | **Tesp** |
| **P1** | 2400 | 2400-100=2300 | 2300-600=1700 |
| **P2** | 1600 | 1600-200=1400 | 1400-500=900 |
| **P3** | 500 | 500-300=200 | 200-200=0 |
| **P4** | 800 | 800-400=400 | 400-300=100 |
| **P5** | 2100 | 2100-500=1600 | 1600-700=900 |
| **Promedio** |  | **1180** | **720** |

En el caso de FIFO, se tiene:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **T** | **CPU** | **Q1** | **Q2** | **Q3** |
| 100 | P1(3,3,600,0) |  |  |  |
| 200 | P2(2,2,500,0) |  |  | P1(3,3,500,0) |
| 300 | P3(1,1,200,0) |  | P2(2,2,400,0) | P1(3,3,500,100) |
| 400 | P3(1,1,100,0) | P4(1,1,300,0) | P2(2,2,400,100) | P1(3,3,500,200) |
| 500 | P3->FIN  P4(1,1,300,0) |  | P2(2,2,400,200)  P5(2,2,700,0) | P1(3,3,500,300) |
| 800 | P4->FIN  P2(2,2,400,0) |  | P5(2,2,700,300)  P1(2,3,500,0) |  |
| 1100 | P5(1,2,700,0) |  | P1(2,3,500,300)  P2(2,2,100,0) |  |
| 1400 | P5(1,2,400,0) | P1(1,3,500,0) | P2(2,2,100,300) |  |
| 1700 | P5(1,2,100,0) | P1(1,3,500,300)  P2(1,2,100,0) |  |  |
| 1800 | P5->FIN  P1(1,3,500,0) | P2(1,2,100,100) |  |  |
| 2300 | P1->FIN  P2(1,2,100,0) |  |  |  |
| 2400 | P2->FIN |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Proces** | **Tfin** | **Tret** | **Tesp** |
| **P1** | 2300 | 2300-100=2200 | 2200-600=1600 |
| **P2** | 2400 | 2400-200=2200 | 2200-500=1700 |
| **P3** | 500 | 500-300=200 | 200-200=0 |
| **P4** | 800 | 800-400=400 | 400-300=100 |
| **P5** | 1800 | 1800-500=1300 | 1300-700=600 |
| **Promedio** |  | **1260** | **800** |

# Exercise 3.

Consider a multithreaded operating system in which 4 processes are running concurrently, whose separate theoretical execution traces are as follows.

Processes A and B are low priority, while processes C and D are high priority.

**Proces A**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CPU** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| **E/S** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Tiempo | 0 | | 25 | | 50 | | 75 | | 100 | | 125 | | 150 | | 175 | | 200 | | 225 | | 250 | | 300 | | 350 | |

**Proces B**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CPU** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| **E/S** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Tiempo | 0 | | 25 | | 50 | | 75 | | 100 | | 125 | | 150 | | 175 | | 200 | | 225 | | 250 | | 300 | | 350 | |

**Proces C**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CPU** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| **E/S** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Tiempo | 0 | | 25 | | 50 | | 75 | | 100 | | 125 | | 150 | | 175 | | 200 | | 225 | | 250 | | 300 | | 350 | |

**Proces D**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CPU** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| **E/S** | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |
| Tiempo | 0 | | 25 | | 50 | | 75 | | 100 | | 125 | | 150 | | 175 | | 200 | | 225 | | 250 | | 300 | | 350 | |

a) Represent the execution trace of the CPU (indicating which process is in the CPU at each moment), when a priority scheduling algorithm is used. Within each priority level, a Round Robin algorithm is used with a time slice of 40 units. Indicate the status of the scheduling queues at all times.

b) Determine the mean waiting time and the mean normalized return time.

***SOLUTION:***

| Tiempo | En ejecución | Cola AP | Cola BP | Bloqueados |
| --- | --- | --- | --- | --- |
| 0 | A (0 | 50) |  | B (0 | 100) |  |
| 40 | B(0 | 100) |  | A (40 | 10) |  |
| 50 | C (0 | 50) | D (0 | 150) | A (40 | 10), B (10 | 90) |  |
| 90 | D (0 | 150) | C (40 | 10) | A (40 | 10), B (10 | 90) |  |
| 130 | C (40 | 10) | D (40 | 110) | A (40 | 10), B (10 | 90) |  |
| 140 | D (40 | 110) |  | A (40 | 10), B (10 | 90) | C (190) |
| 180 | D (80 | 70) |  | A (40 | 10), B (10 | 90) | C (190) |
| 190 | D (90 | 60) | C (50 | 50) | A (40 | 10), B (10 | 90) |  |
| 220 | C (50 | 50) | D (120 | 30) | A (40 | 10), B (10 | 90) |  |
| 260 | D (120 | 30) | C (90 | 10) | A (40 | 10), B (10 | 90) |  |
| 290 (Fin D) | C (90 | 10) |  | A (40 | 10), B (10 | 90) |  |
| 300 (Fin C) | A (40 | 10) |  | B (10 | 90) |  |
| 310 | B (10 | 90) |  |  | A (360) |
| 350 | B (50 | 50) |  |  | A (360) |
| 360 | B (60 | 40) |  | A (50 | 50) |  |
| 390 | A (50 | 50) |  | B (90 | 10) |  |
| 430 | B (90 | 10) |  | A (90 | 10) |  |
| 440 | A (90 | 10) |  |  | B (490) |
| 450 |  |  |  | B (490), A (500) |
| 490 | B (100 | 100) |  |  | A (500) |
| 500 | B (110 | 90) |  | A (100 | 50) |  |
| 530 | A (100 | 50) |  | B (140 | 60) |  |
| 570 | B (140 | 60) |  | A (140 | 10) |  |
| 610 | A (140 | 10) |  | B (180 | 20) |  |
| 620 Fin A | B (180 | 20) |  |  |  |
| 640 Fin B |  |  |  |  |

| Proces | Llegada | Servicio | Inicio | Fin | Retorno | Espera | R Norm |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | 0 | 250 | 0 | 620 | 620 | 620 – 250 = 370 | 620/250 = 2.48 |
| B | 0 | 250 | 40 | 640 | 600 | 600 – 250 = 350 | 600 / 250 = 2,4 |
| C | 50 | 150 | 50 | 300 | 300 – 50 = 250 | 250 – 150 = 100 | 250 / 150 = 1.667 |
| D | 50 | 150 | 90 | 290 | 290 -90 = 200 | 200 – 150 = 50 | 200 / 150 = 1,333 |

Normalized Turnaround time = 1,97

# Exercise 4.

An operating system uses a cyclic (round-robin) scheduler. At any given moment there are no jobs running and you want to run four jobs whose arrival times to the system are as follows:

|  |  |
| --- | --- |
| Process | Arrival Time |
| A | 0 |
| B | 500 |
| C | 200 |
| D | 400 |

The execution traces of the processes indicate the following alternative sequences of execution in CPU and time required for I / O operations.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Proces A | | Proces B | | Proces C | | Proces D | |
| CPU | E/S | CPU | E/S | CPU | E/S | CPU | E/S |
| 200 |  | 20 |  | 500 |  | 1500 |  |
|  | 500 |  | 100 |  | 100 |  |  |
| 100 |  | 40 |  | 1000 |  |  |  |
|  |  |  | 100 |  |  |  |  |
|  |  | 30 |  |  |  |  |  |
|  |  |  | 100 |  |  |  |  |
|  |  | 50 |  |  |  |  |  |

When a context switch occurs, the operating system requires 10 TU to perform a full context switch. If the same process that was already running is selected, the context switch only requires 5 TUs.

On the other hand, when an I / O operation ends, a service treatment routine must be executed by the operating system for 5UT, after which the execution of the current process continues until its time slice is completely consumed. For simplicity, it is considered that each process performs input / output on a different device and there are no interactions between them. In the event of an I / O interrupt and a clock interrupt at the same time, the I / O interrupts are considered to have higher priority.

The cases with a time slice of 200 TU and 500 TU will be considered.

a) Service time (time spent on productive tasks: CPU / ES).

b) Waiting time (time that each process spends in waiting queues).

c) Average waiting time.

d) Return time (difference between arrival time and end time).

e) Normalized return time (ratio between return time and service time).

In addition, the student is asked to make a critical comment on the results.

***SOLUTION:***

**200 UT slice:**

| Tiempo | En ejecución | Cola | E/S | Rodaja |
| --- | --- | --- | --- | --- |
| 0 | A(200) |  |  | 0 |
| 200 | CC | C(500) | A(500) | 200 |
| 210 | C(500) |  | A(490) | 0 |
| 400 | C(310) | D(1500) | A(300) | 190 |
| 410 | CC | D(1500), C(300) | A(290) | 200 |
| 420 | D(1500) | C(300) | A(280) | 0 |
| 500 | D(1420) | C(300), B(20) | A(200) | 80 |
| 620 | CC | C(300), B(20), D(1300) | A(80) | 200 |
| 630 | C(300) | B(20), D(1300) | A(70) | 0 |
| 700 | E/S – A | C(230), B(20), D(1300) |  | 70 |
| 705 | C(230) | B(20), D(1300), A(100) |  | 70 |
| 835 | CC | B(20), D(1300), A(100), C(100) |  | 200 |
| 845 | B(20) | D(1300), A(100), C(100) |  | 0 |
| 865 | CC | D(1300), A(100), C(100) | B(100) | 20 |
| 875 | D(1300) | A(100), C(100) | B(90) | 0 |
| 965 | E/S – B | D(1210), A(100), C(100) |  | 90 |
| 970 | D(1210) | A(100), C(100), B(40) |  | 90 |
| 1080 | CC | A(100), C(100), B(40), D(1100) |  | 200 |
| 1090 | A(100) | C(100), B(40), D(1100) |  | 0 |
| 1190 | CC – Fin A | C(100), B(40), D(1100) |  | 100 |
| 1200 | C(100) | B(40), D(1100) |  | 0 |
| 1300 | CC | B(40), D(1100) | C(100) | 100 |
| 1310 | B(40) | D(1100) | C(90) | 0 |
| 1350 | CC | D(1100) | C(50) | 40 |
| 1360 | D(1100) |  | C(40), B(100) | 0 |
| 1400 | E/S – C | D(1060) | B(60) | 40 |
| 1405 | D(1060) | C(1000) | B(55) | 40 |
| 1460 | E/S – B | D(1005), C(1000) |  | 95 |
| 1465 | D(1005) | C(1000), B(30) |  | 95 |
| 1570 | CC | C(1000), B(30), D(900) |  | 200 |
| 1580 | C(1000) | B(30), D(900) |  | 0 |
| 1780 | CC | B(30), D(900), C(800) |  | 200 |
| 1790 | B(30) | D(900), C(800) |  | 0 |
| 1820 | CC – Fin B | D(900), C(800) |  | 30 |
| 1830 | D(900) | C(800) |  | 0 |
| 2030 | CC | C(800), D(700) |  | 200 |
| 2040 | C(800) | D(700) |  | 0 |
| 2240 | CC | D(700), C(600) |  | 200 |
| 2250 | D(700) | C(600) |  | 0 |
| 2450 | CC | C(600), D(500) |  | 200 |
| 2460 | C(600) | D(500) |  | 0 |
| 2660 | CC | D(500), C(400) |  | 200 |
| 2670 | D(500) | C(400) |  | 0 |
| 2870 | CC | C(400), D(300) |  | 200 |
| 2880 | C(400) | D(300) |  | 0 |
| 3080 | CC | D(300), C(200) |  | 200 |
| 3090 | D(300) | C(200) |  | 0 |
| 3290 | CC | C(200), D(100) |  | 200 |
| 3300 | C(200) | D(100) |  | 0 |
| 3500 | CC – Fin C | D(100) |  | 200 |
| 3510 | D(100) |  |  | 0 |
| 3610 | Fin D |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Proces | Llegada | Servicio | Fin | Retorno | Espera | Ret. N. |
| A | 0 | 800 | 1190 | 1190 | 390 | 1.488 |
| B | 500 | 290 | 1820 | 1320 | 1030 | 4.551 |
| C | 200 | 1600 | 3500 | 3300 | 1700 | 2.063 |
| D | 400 | 1500 | 3610 | 3210 | 1710 | 2.14 |

Average waiting time 1207.05.

Average turnaround 2.561.

**500 UT slice:**

| Tiempo | En ejecución | Cola | E/S | Rodaja |
| --- | --- | --- | --- | --- |
| 0 | A(200) |  |  | 0 |
| 200 | CC | C(500) | A(500) | 200 |
| 210 | C(500) |  | A(490) | 0 |
| 400 | C(310) | D(1500) | A(300) | 190 |
| 500 | C(210) | D(1500), B(20) | A(200) | 290 |
| 700 | E/S – A | C(10), D(1500), B(20) |  | 490 |
| 705 | C(10) | D(1500), B(20), A(100) |  | 490 |
| 715 | CC | D(1500), B(20), A(100) | C(100) | 500 |
| 725 | D(1500) | B(20), A(100) | C(90) | 0 |
| 815 | E/S – C | D(1410), B(20), A(100) |  | 90 |
| 820 | D(1410) | B(20), A(100), C(1000) |  | 90 |
| 1230 | CC | B(20), A(100), C(1000), D(1000) |  | 500 |
| 1240 | B(20) | A(100), C(1000), D(1000) |  | 0 |
| 1260 | CC | A(100), C(1000), D(1000) | B(100) | 20 |
| 1270 | A(100) | C(1000), D(1000) | B(90) | 0 |
| 1360 | E/S – B | A(10), C(1000), D(1000) |  | 90 |
| 1365 | A(10) | C(1000), D(1000), B(40) |  | 90 |
| 1375 | CC – Fin A | C(1000), D(1000), B(40) |  | 100 |
| 1385 | C(1000) | D(1000), B(40) |  | 0 |
| 1885 | CC | D(1000), B(40), C(500) |  | 500 |
| 1895 | D(1000) | B(40), C(500) |  | 0 |
| 2395 | CC | B(40), C(500), D(500) |  | 500 |
| 2405 | B(40) | C(500), D(500) |  | 0 |
| 2445 | CC | C(500), D(500) | B(100) | 40 |
| 2455 | C(500) | D(500) | B(90) | 0 |
| 2545 | E/S – B | C(410), D(500) |  | 90 |
| 2550 | C(410) | D(500), B(30) |  | 90 |
| 2960 | CC – Fin C | D(500), B(30) |  | 500 |
| 2970 | D(500) | B(30) |  | 0 |
| 3470 | CC – Fin D | B(30) |  | 500 |
| 3480 | B(30) |  |  | 0 |
| 3510 | Fin B |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Proces | Llegada | Servicio | Fin | Retorno | Espera | Ret. N. |
| A | 0 | 800 | 1375 | 1375 | 575 | 1.719 |
| B | 500 | 290 | 3510 | 3010 | 2720 | 10.379 |
| C | 200 | 1600 | 2960 | 2760 | 1160 | 1.725 |
| D | 400 | 1500 | 3470 | 2460 | 960 | 1.64 |

Average waiting time1353.75.

Average turnaround 3.866.

Critical comment: As the time slice increases, tasks with low interactivity and long execution periods (C and D) are favored. In addition, the total execution time of the job set is reduced.

# Exercise 5

On a given operating system that schedules using priority queues, processes run based on their priority (1 being the highest). It is assumed that the processes enter the system in its priority queue and do not move from it.

When several processes have the same priority, a round robin scheduling policy is used, with a 100 ms slice. When the slice of a process is finished, it is put at the end of its priority queue.

The following table specifies for each process, its priority, its arrival time and the time it takes to execute, distinguishing CPU and blocking I / O.

Note: With blocking input / output the process frees the CPU and it is ready for the next process. When the I / O data is ready, the process moves to the ready-to-run queue, but at the end of its priority queue.

|  |  |  |  |
| --- | --- | --- | --- |
| PROCESSES | PRIORITY | ARRIVAL | EXECUTION TIME |
| P1 | 3 | 0 | 250 CPU + 100 E/S + 200 CPU |
| P2 | 2 | 200 | 300 CPU |
| P3 | 1 | 400 | 100 CPU + 250 E/S + 50 CPU |
| P4 | 1 | 500 | 400 CPU |
| P5 | 2 | 400 | 100 CPU + 100 E/S + 100 CPU |

For the following two situations:

a) Scheduling without expulsion. The preparation with the highest priority is chosen but the one who is there is not expelled.

b) Scheduling with expulsion. The one who is executing is expelled if one of higher priority arrives or his time slice expires.

It is requested

1. Make a cronogram of the execution of the processes.

2. Calculate the time that each process is kept on hold from its arrival in the system until it ends.

3. Calculate the return time of each process (time elapsed since the process arrives until the end of its execution).

4. Average waiting time and average return time.

***SOLUCIÓN Exercise 3: (Octubre. Examen Parcial 1. Leganés. Curso 2010-2011)***

1. **Non-preemptive scheduling:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | CPU | CPU | CPU | CPU | CPU | E/S | E/S |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU | CPU | CPU |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 |  |  |  |  |  | CPU | CPU | CPU | CPU | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| P3 |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU | E/S | E/S | E/S | E/S | E/S |  |  |  | CPU |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |
| P4 |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU | CPU | CPU | CPU | CPU | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |
| P5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU | E/S | E/S |  |  | CPU | CPU |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **Avg** |
| **Wait time (ms)** | 850 | 50 | 300 | 150 | 800 | 430 |
| **Turnaround time (ms)** | 1400 | 350 | 700 | 550 | 1100 | 820 |

1. **Preemptive scheduling:**

* Entra primero en la cola de listos el Proces nuevo que entra, no el expulsado 🡪 *implica que a los 950 ms el que ejecuta es P5, antes que P2*.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | CPU | CPU | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | E/S | E/S | CPU | CPU | CPU | CPU |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 |  |  |  |  | CPU | CPU | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| P3 |  |  |  |  |  |  |  |  | CPU | CPU | E/S | E/S | E/S | E/S | E/S |  | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| P4 |  |  |  |  |  |  |  |  |  |  | CPU | CPU | CPU | CPU | CPU | CPU |  | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| P5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU | E/S | E/S | CPU | CPU |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **Avg** |
| **Wait time (ms)** | 1050 | 650 | 50 | 50 | 550 | 470 |
| **Turnaround time(ms)** | 1600 | 950 | 450 | 450 | 850 | 860 |

In addition to this solution, there is another possibility that has also been considered valid:

POSSIBILITY B: (Scheduling with expulsion)

• The process expulsed enters the ready queue first, not the new one that enters ◊ implies that at 950 ms the one that executes is P2, before P5.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P1 | CPU | CPU | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | E/S | E/S |  | CPU | CPU | CPU | CPU |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| P2 |  |  |  |  | CPU | CPU | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| P3 |  |  |  |  |  |  |  |  | CPU | CPU | E/S | E/S | E/S | E/S | E/S |  | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| P4 |  |  |  |  |  |  |  |  |  |  | CPU | CPU | CPU | CPU | CPU | CPU |  | CPU | CPU |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |
| P5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | CPU | CPU | E/S | E/S | CPU | CPU |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500 1600

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **Avg** |
| **Wait time (ms)** | 1000 | 550 | 50 | 50 | 650 | 460 |
| **Turnaround time (ms)** | 1550 | 850 | 450 | 450 | 950 | 850 |

# Exercise 6.

In a given operating system, processes are executed based on multilevel queues with the following characteristics:

- The system has 3 queues:

o The first one follows a Round-Robin scheduling algorithm with a quantum of 2ms.

o The second follows a Round-Robin algorithm with a quantum of 4 ms.

o The third follows a FIFO scheduling.

- The processes enter the system through the first queue.

- Processes are downgraded if the system expulses them from the processor due to quantum expiration.

- Scheduling between queues is based on priorities, with the first being the highest priority, then the second and then the third.

Answer:

1. Draw a cronogram for the following set of processes:

|  |  |  |
| --- | --- | --- |
| PROCESSES | ARRIVAL | EXECUTION |
| P1 | 0 | 1ms CPU + 6ms E/S + 1ms CPU |
| P2 | 1 | 3 ms CPU |
| P3 | 3 | 5ms CPU + 3ms E/S + 1ms CPU |
| P4 | 3 | 3 ms CPU |

Draw the solution in the following table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Proces** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **P1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **P2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **P3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **P4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

b) Indicate for each process their time of stay in the system and the penalty time suffered by each one of them.

c) What is the worst treated process?

***SOLUCIÓN Exercise 3: (Octubre. Examen Parcial 2. Leganés. Curso 2010-2011)***

a)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Proces** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **P1** | CPU | E/S | E/S | E/S | E/S | E/S | E/S | CPU |  |  |  |  |  |  |  |  |
| **P2** |  | CPU | CPU |  |  |  |  |  | CPU |  |  |  |  |  |  |  |
| **P3** |  |  |  | CPU | CPU |  |  |  |  | CPU | CPU | CPU | E/S | E/S | E/S | CPU |
| **P4** |  |  |  |  |  | CPU | CPU |  |  |  |  |  | CPU |  |  |  |

b)

|  |  |  |
| --- | --- | --- |
| **Process** | **Stay** | **Penalty** |
| **P1** | **8** | **0** |
| **P2** | **8** | **5** |
| **P3** | **13** | **4** |
| **P4** | **10** | **7** |

c) The worst treated process is process 4 since it takes 10 periods to finish its execution, when it only has 3 execution periods, it suffers 7 penalty periods.

# Exercise 7.

An operating system uses a cyclic (round-robin) scheduler. At a given moment there are no jobs running and you want to run jobs whose arrival times to the system are as follows:

Priorities are inverse of their value. Thus, a process with priority 1 is prioritized over another with priority 2 or 3.

You are asked to fill in the following tables in the following cases:

a) Round-robin scheduling policy with slice of 1

b) Round-robin scheduling policy with slice of 4

c) SJF (Shortest Job First) scheduling Policy (Non-Expulsive)

NOTE: If the execution slice of a process ends at the same instant that a new process arrives on the system, then the new process is placed in the ready-to-run queue before the process that expires the slice.

***SOLUCIÓN Exercise 2: (Octubre. Examen Parcial 1. Colmenarejo. Curso 2010-2011)***

a) *round-robin*, slice 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | 5 |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  | 3 |  | 2 |  | 1 |  |  |  |
| **B** |  | 7 |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  | 5 |  | 4 |  | 3 | 2 | 1 |
| **C** |  |  | 5 | 4 |  | 3 |  |  |  |  |  | 2 |  | 1 |  |  |  |  |  |  |  |  |  |  |
| **D** |  |  |  |  | 3 |  |  |  |  |  | 2 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |
| **E** |  |  |  |  |  |  | 4 | 3 | 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **End time** | **Turnaround** | **Execution time** | **Wait time** | **Turnaround normalized** |
| A | 21 | 21-0=21 | 5 | 21-5=16 | 21/5=4,2 |
| B | 23 | 24-1=23 | 7 | 23-7=16 | 23/7=3,3 |
| C | 14 | 14-2=12 | 5 | 12-5=7 | 12/5=2,4 |
| D | 13 | 13-4=9 | 3 | 9-3=6 | 9/3=3 |
| E | 10 | 10-6=4 | 4 | 4-4=0 | 6/6=1 |
| **Average** | | 13,8 | 4,8 | 9 | 2,78 |

b) *round-robin* slice = 4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | 5 | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 3 | 2 | 1 |  |  |  |  |
| **B** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 | 6 | 5 |  |  |  | 4 | 3 | 2 | 1 |
| **C** |  |  | 5 | 4 | 3 |  |  |  |  |  | 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| **D** |  |  |  |  |  | 3 |  |  |  |  |  |  | 2 | 1 |  |  |  |  |  |  |  |  |  |  |
| **E** |  |  |  |  |  |  | 4 | 3 | 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **End time** | **Turnaround** | **Execution time** | **Wait time** | **Turnaround normalizado** |
| A | 20 | 20-0=20 | 5 | 20-5=15 | 20/5=4 |
| B | 24 | 24-1=23 | 7 | 23-7=16 | 23/7=3,3 |
| C | 12 | 12-2=10 | 5 | 10-5=5 | 10/5=2 |
| D | 14 | 14-4=10 | 3 | 10-3=7 | 10/3=3,3 |
| E | 10 | 10-6=4 | 4 | 4-4=0 | 4/4=1 |
| **Valores medios** | | 13,4 | 4,8 | 8,6 | 2,72 |

**c)** SJF (*Shortest Job First*)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | 5 | 4 | 3 | 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **B** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| **C** |  |  |  |  |  |  |  |  |  |  |  |  | 5 | 4 | 3 | 2 | 1 |  |  |  |  |  |  |  |
| **D** |  |  |  |  |  | 3 | 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **E** |  |  |  |  |  |  |  |  | 4 | 3 | 2 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| Tiempo | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Process** | **End time** | **Turnaround** | **Execution time** | **Wait time** | **Turnaround normalizado** |
| A | 5 | 5-0=5 | 5 | 5-5=0 | 5/5=1 |
| B | 24 | 24-1=23 | 7 | 23-7=16 | 23/7=3,3 |
| C | 17 | 17-2=15 | 5 | 15-5=10 | 15/5=3 |
| D | 8 | 8-4=6 | 3 | 6-3=3 | 6/3=2 |
| E | 12 | 12-6=6 | 4 | 6-4=2 | 6/4=1,5 |
| **Valores medios** | | 11 | 4,8 | 6,2 | 2,16 |