

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

Sistemas Operativos

Práctica de control



UNIVERSIDAD DE BURGOS

1º curso – GRADO EN INGENIERÍA INFORMÁTICA

Asignatura: Sistemas Operativos

Alumno: Miguel Díaz Hernando

Curso: 2022/2023

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

Índice

1.- Introducción.....	3
1.1- Descripción del algoritmo RR FNI Peor.....	3
1.2- Gestión de memoria con particiones fijas no iguales, peor ajuste.....	4
1.3- Objetivo.....	4
2.- Ejemplos manuales.....	5
2.1- Ejemplo 1.....	5
3.- Ejemplo mediante el script.....	8
3.1- Ejemplo 1.....	9
3.2- Ejemplo 2.....	17
4.- Conclusiones.....	23
5.- Bibliografía.....	24

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

1.- Introducción

En ésta práctica se ha desarrollado un algoritmo de gestión de procesos a través de Round Robin, con una gestión de memoria por particiones fijas y no iguales, con el criterio de peor ajuste.

1.1- Descripción del algoritmo RR FNI Peor

Round Robin es un algoritmo sencillo y fácil de aplicar, pensado con sistemas de uso compartido en mente.

Se trata de un algoritmo apropiativo¹, en el que el sistema operativo asigna un quantum² o tiempo de ejecución a los procesos, con un valor constante y común a todos los procesos. Al completar dicho quantum, si el proceso no ha acabado, es desalojado de CPU para dar paso a otro.

El orden de ejecución de los procesos se decide mediante una cola FIFO³. Los procesos se van encolando por orden de llegada y esperando a ejecutarse durante su quantum asignado. Si no finalizan en su quantum, son expulsados al terminar éste y vuelven al final de la cola a la espera de un siguiente turno de uso de la CPU, que es ocupada por el siguiente proceso de la cola. Si el proceso finaliza, de igual manera accede a la CPU el siguiente proceso de la cola con un quantum entero.

-
- 1 Un sistema apropiativo es aquel en el que el sistema operativo puede expulsar a la fuerza a procesos en ejecución de la CPU para alojar a otros.
 - 2 El quantum es un valor que representa la cantidad de tiempo de ejecución que le corresponde a un proceso antes de ser expulsado de CPU para ceder la ejecución a otro proceso. Se define al comienzo del algoritmo y permanece constante y común a todos los procesos.
 - 3 Una cola FIFO es aquella en la que se da la prioridad al elemento que antes ha sido añadido a la misma.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

1.2.- Gestión de memoria con particiones fijas no iguales, peor ajuste

Este método de gestión de memoria sigue un esquema de gestión sin memoria virtual⁴, todo el proceso a ejecutar debe estar cargado en memoria, y una asignación contigua, todo el proceso debe cargarse en posiciones de memoria consecutivas.

La memoria se divide en particiones de tamaños diferentes, que permanecerán inmutables durante la ejecución del algoritmo. Se dispone de una única cola, con funcionamiento FIFO como viene definido en el uso de Round Robin. Ésto puede producir una contrapartida como desaprovechamiento de la memoria.

Se emplea un ajuste al peor⁵, cuando un proceso accede a memoria se ubica en la partición que más espacio libre deje tras su alojamiento.

1.3.- Objetivo

Explicado éstos conceptos, el objetivo de la práctica es desarrollar un script en bash que simule la gestión de un sistema con un algoritmo Round Robin, con particiones fijas no iguales al peor ajuste, pudiendo elegir distintos valores para cada parámetro en cada ejecución.

4 En los esquemas de gestión con memoria virtual solo es necesario que una parte del proceso en ejecución esté cargada en memoria.

5 Otros ajustes son al mejor, el que menos espacio libre deje, y al primero que se encuentre libre.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

2.- Ejemplos manuales

A continuación veremos algunos ejemplos cubriendo los casos límite de ejecución del algoritmo desarrollado.

2.1.- Ejemplo 1

- Se escogen las siguientes particiones:

Partición:	P01	P02	P03	P04	P05
Tamaño:	2	5	3	7	4

Con estas particiones podremos comprobar fácilmente el peor ajuste.

- Se escoge un quantum de 2.
- Se escogen los procesos:

Proceso:	T Llegada:	Ráfaga CPU:	Memoria:
P01	20	1	5
P02	5	2	5
P03	4	1	3
P04	1	5	2
P05	3	3	2
P06	6	2	2

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

Al introducir estos procesos, lo primero que podremos comprobar es la correcta ordenación de procesos debido la introducción desordenada de los tiempos de llegada. Resultarán en este orden:

Proceso:	T Llegada:	Ráfaga CPU:	Memoria:
P04	1	5	2
P05	3	3	2
P03	4	1	3
P02	5	2	5
P06	6	2	2
P01	20	1	5

A continuación se representa el desarrollo del ejercicio:

	T = 0	T = 1	T = 3	T = 4	T = 5	T = 6	T = 7	T = 8	T = 9	T = 11	T = 12	T = 14	T = 20	T = 21
Ejecución	-	P04 (5) Part 04	P04 (3) Part 04	P04 (2) Part 04	P05 (3) Part 02	P05 (2) Part 02	P03 (1) Part 05	P04 (1) Part 04	P06 (2) Part 03	P05 (1) Part 02	P02 (2) Part 04	-	P01 (1) Part 04	-
Memoria	-	-	P05 (3) Part 02	P05 (3) Part 02 P03 (1) Part 05	P03 (1) Part 05 P04 (1) Part 04	P03 (1) Part 05 P04 (1) Part 04 P06 (2) Part 03	P04 (1) Part 04 P06 (2) Part 03 P05 (1) Part 02	P06 (2) Part 03 P05 (1) Part 02	P05 (1) Part 02 P02 (2) Part 04	P02 (2) Part 04	-	-	-	-
En espera	-	-	-	-	P02 (2)	-	P02 (2)	P02 (2)	-	-	-	-	-	-
Fuera del Sistema	P04 (5) P05 (3) P03 (1) P02 (2) P06 (2) P01 (1)	P05 (3) P03 (1) P02 (2) P06 (2) P01 (1)	P03 (1) P02 (2) P06 (2) P01 (1)	P02 (2) P06 (2) P01 (1)	P06 (2) P01 (1)	-	P01 (1)	P01 (1)	P01 (1)	P01 (1)	P01 (1)	-	-	-
Finalizado	-	-	-	-	-	-	-	P03	P03 P04	P03 P04 P06	P03 P04 P06 P05 P02	P03 P04 P06 P05 P02	P03 P04 P06 P05 P02	P03 P04 P06 P05 P02 P01

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

Se representa en una columna cada uno de los posibles estados de los procesos, y en el eje horizontal se aprecia el transcurso del tiempo. Cada proceso viene acompañado por su tiempo restante de ejecución, y se pintan en morado si están en memoria después de haber recibido tiempo de ejecución.

- En $T=0$ vemos todos los procesos fuera del sistema puesto que el primero llega en $T=1$.
- En $T=1$ el P04 llega al sistema, entra en memoria a la partición 4 cumpliendo el peor ajuste, y entra a ejecución ya que no tiene competencia por la CPU.
- En $T=3$ P04 acaba su quantum, y simultáneamente llega P05 al sistema. En esta situación, P04 entra a la cola antes de comprobar la llegada de P05, de forma que es P04 el que vuelve a disfrutar de la CPU, quedando P05 en espera alojado en la partición 2 cumpliendo el peor ajuste.
- En $T=4$ P03 llega al sistema y se aloja en la partición 5, la siguiente libre más grande. P04 continúa con su quantum de ejecución.
- En $T=5$ P04 acaba su quantum y vuelve a la cola, entrando a ejecución P05. Llega P02 al sistema, Pero al no haber en ninguna partición libre permanece a la espera.
- En $T=6$ P06 llega al sistema y entra a memoria a la partición 3 a pesar de seguir P02 a la espera, que no bloquea a procesos posteriores. P05 sigue en su quantum.
- En $T=7$ P05 acaba su quantum y vuelve a la cola, entrando a ejecución P03.
- En $T=8$ P03 finaliza su ejecución en mitad de su quantum, y entra a ejecución P04 de nuevo.
- En $T=9$ P04 finaliza su ejecución y al dejar libre la partición 4 P02 entra a memoria y se pone en la cola. Entra a CPU P06.
- En $T=11$ P06 finaliza su ejecución y entra a CPU P05.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

- En T=12 P05 finaliza su ejecución, y entra P02.
- En T=14 P02 finaliza su quantum y su ejecución, dejando la CPU vacía.
- En T=20 P01 entra al sistema y se aloja en la partición 4, entrando también a ejecución al no encontrar competencia.
- En T=21 P01 finaliza su ejecución.

3.- Ejemplo mediante el script

Ahora veremos el mismo ejercicio resuelto por el script que he programado:

Lo primero que pregunta el script es la forma de introducción de los datos. En este caso podemos introducirlos manualmente (1) o mediante un fichero de datos (3). Para este ejemplo usaré un fichero de datos que he generado previamente, ya que el objetivo actual no es ver toda la interacción y opciones del script si no comprobar la resolución del ejercicio.

```
R.R. Fijas no iguales, peor ajuste.  
Mario Juez Gil, Omar Santos, Alvaro Urdiales Santidria,  
Gonzalo Burgos de la Hera, Lucas Olmedo Díez  
Miguel Díaz Hernando  
Versión Junio 2023
```

```
1- Entrada Manual  
2- Fichero de datos de última ejecución (DatosLast.txt)  
3- Otros ficheros de datos  
4- Rangos manuales para valores aleatorios  
5- Fichero de rangos de última ejecución (DatosRangosLast.txt)  
6- Otros ficheros de rangos  
7- Rangos manuales para rangos aleatorios (prueba de casos extremos)  
8- Fichero de rangos aleatorios de última ejecución (DatosRangosAleatoriosLast.txt)  
9- Otros ficheros de rangos para rangos aleatorios  
  
Elija una opción: 3
```

Figura 1: Ejercicio 1; Introducción de datos

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

```
R.R. Fijas no iguales, peor ajuste.  
Mario Juez Gil, Omar Santos, Alvaro Urdiales Santidria,  
Gonzalo Burgos de la Hera, Lucas Olmedo Díez  
Miguel Díaz Hernando  
Versión Junio 2023  
  
1 - DatosDeRangos1.txt  
2 - DatosDefault.txt  
3 - EjemploInforme1.txt  
Introduce uno de los ficheros del listado: 3
```

Figura 2: Ejercicio 1; Selección de fichero

Después pregunta por el modo de ejecución, elegiremos por eventos para ver cada paso.

```
R.R. Fijas no iguales, peor ajuste.  
Mario Juez Gil, Omar Santos, Alvaro Urdiales Santidria,  
Gonzalo Burgos de la Hera, Lucas Olmedo Díez  
Miguel Díaz Hernando  
Versión Junio 2023  
  
1- Por eventos (Intro)  
2- Automática (Tras n segundos)  
3- Completa (Todo seguido)  
Modo de ejecución: 1
```

Figura 3: Ejercicio 1; Modo de ejecución

3.1.- Ejemplo 1

A continuación comienza la ejecución del ejercicio, en la que se irán viendo los momentos clave que fueron explicados en la resolución manual.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

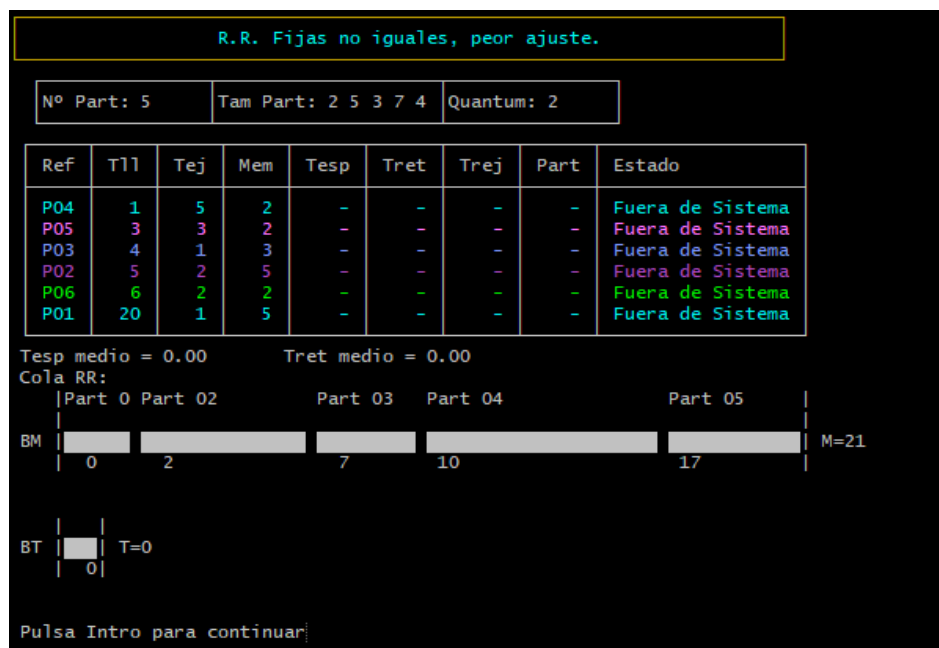


Figura 4: Ejercicio 1; T=0

Inicialmente podemos ver los datos de los procesos ya ordenados, la distribución de la memoria en las particiones, y el T=0 sin procesos.

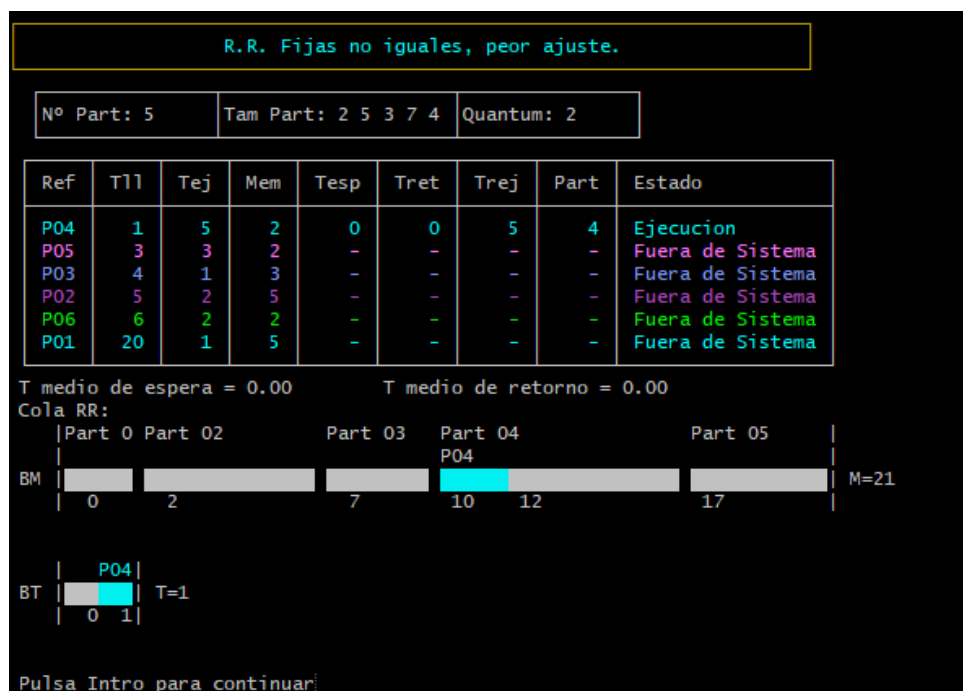


Figura 5: Ejercicio 1; T=1

En T=1 se puede ver la llegada de P04 al sistema y cómo cumple el peor ajuste encontrando la partición libre mayor.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

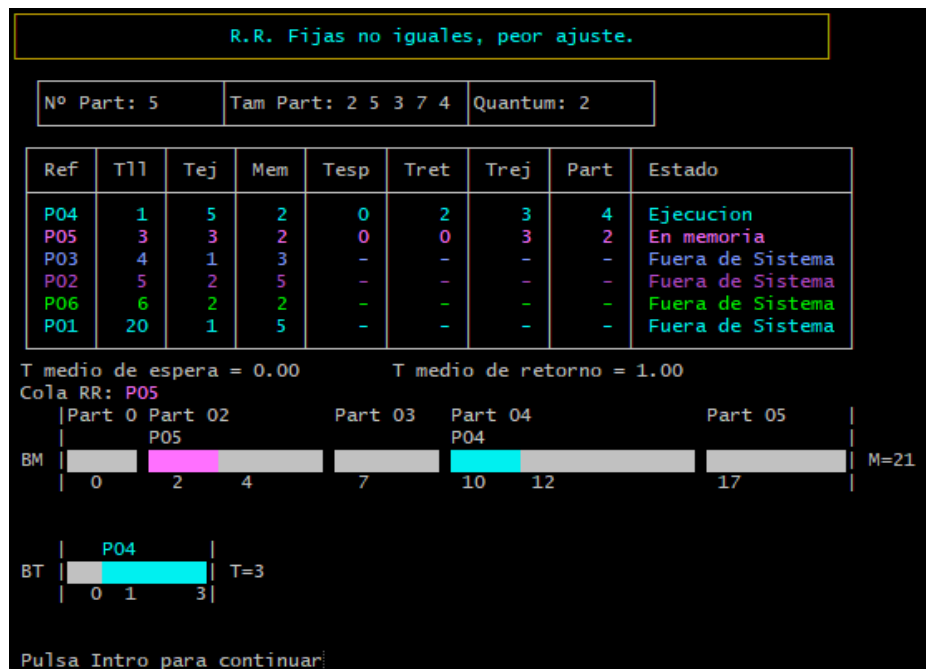


Figura 6: Ejercicio 1; T=3

En T=3 se comprueba cómo P04 entra a la cola antes que P05 cuando simultáneamente el primero acaba su quantum y el segundo entra al sistema.



Figura 7: Ejercicio 1; T=4

En T=4 P03 llega al sistema y se pone en la cola.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

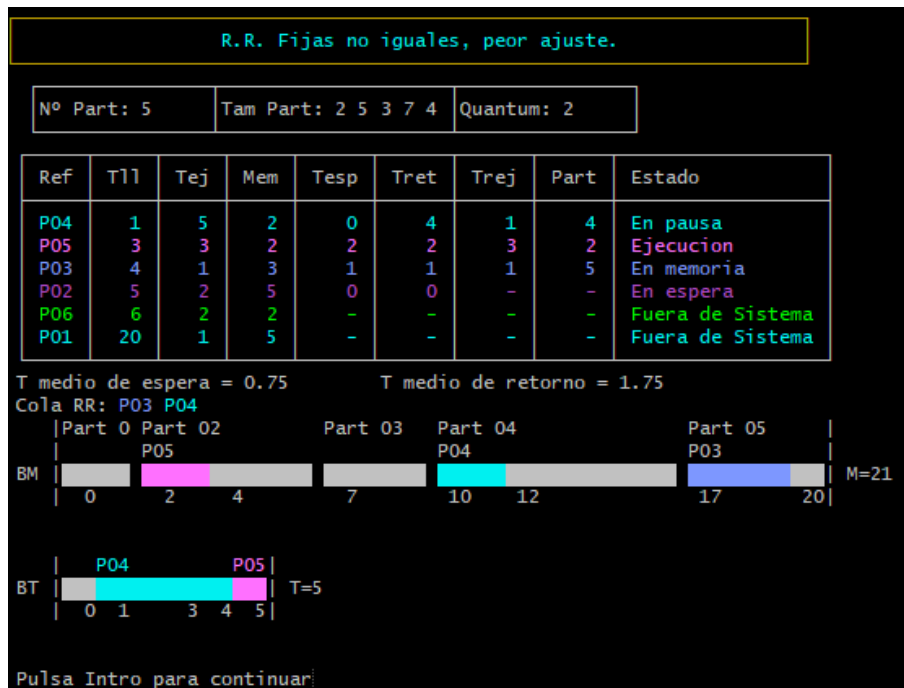


Figura 8: Ejercicio 1; T=5

En T=5 se ve cómo P02 llega al sistema, pero queda en espera al no existir una partición libre que pueda alojarlo.

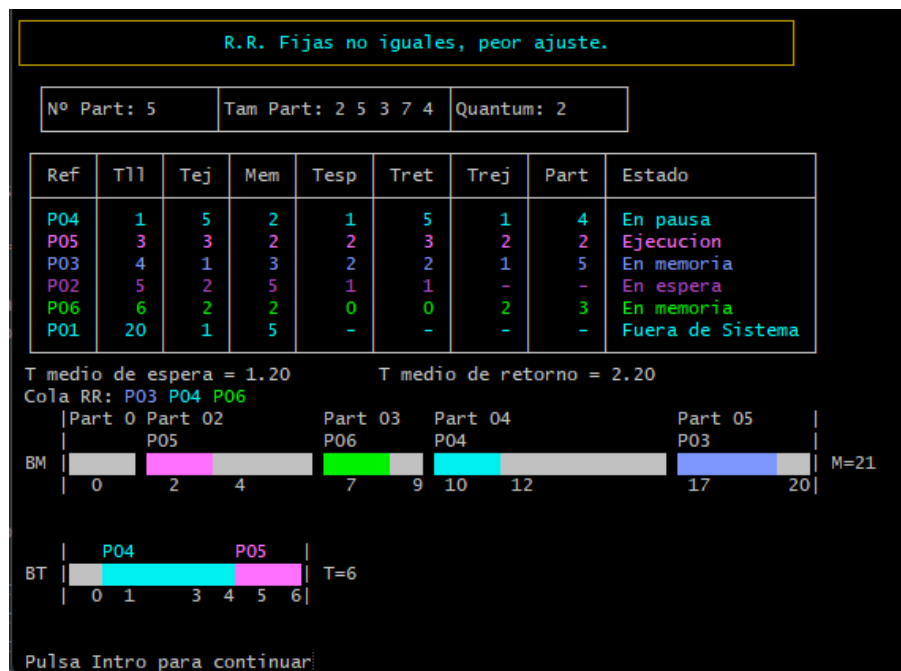


Figura 9: Ejercicio 1; T=6

En T=6 P06 entra al sistema y se comprueba cómo entra a memoria en la partición 3, no siendo bloqueado por estar P02 en espera.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

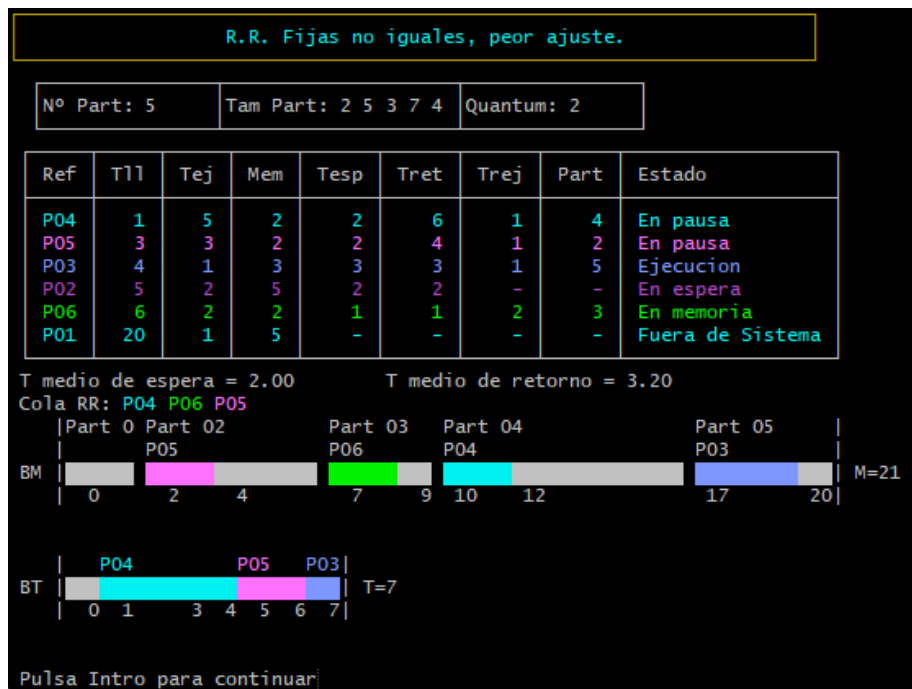


Figura 10: Ejercicio 1; T=7

En T=7 continúa la ejecución de procesos, finalizando su quantum P05 y ejecutándose P03.

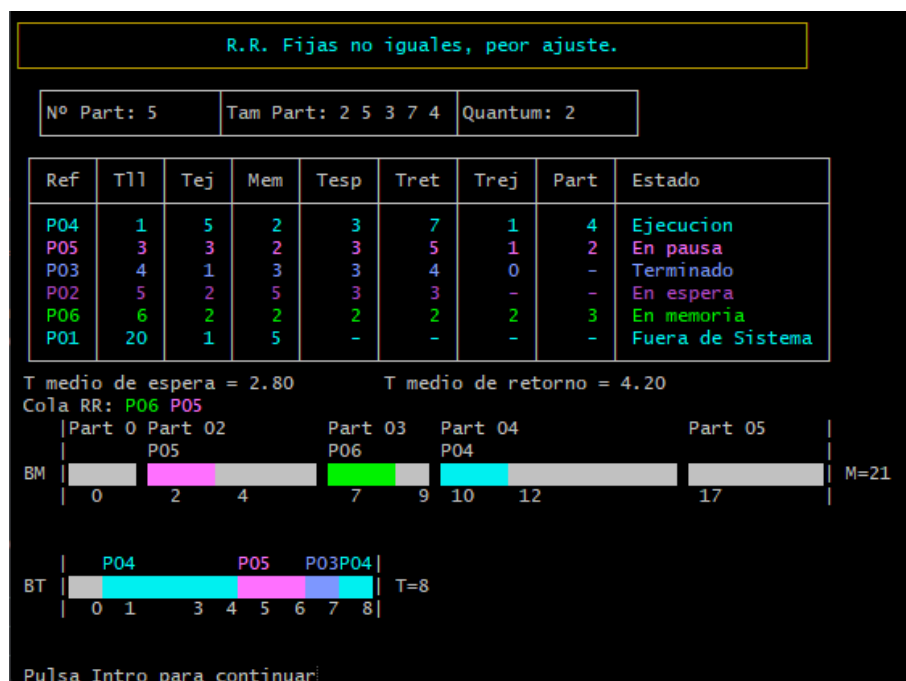


Figura 11: Ejercicio 1; T=8

En T=8 P03 finaliza su ejecución y continúa ejecutándose P04.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

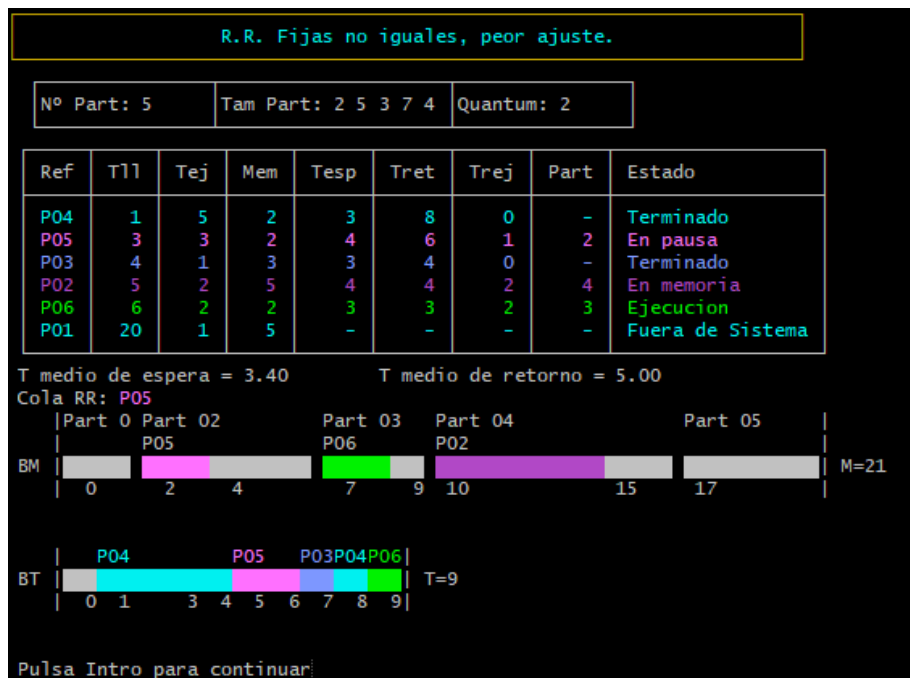


Figura 12: Ejercicio 1; T=9

En T=9 P04 finaliza su ejecución dejando libre la partición 4, pudiendo entonces entrar en memoria P02.

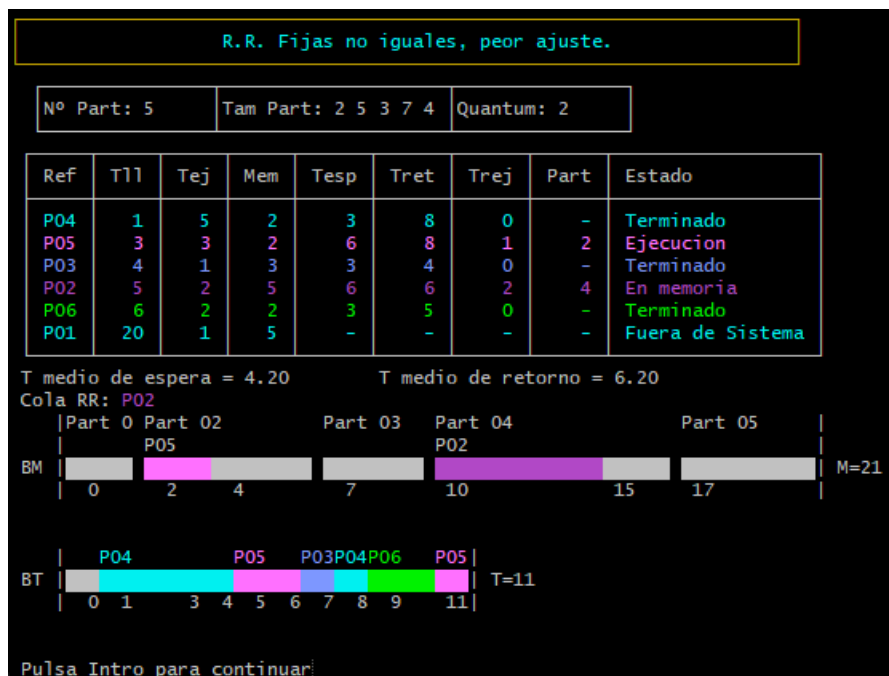


Figura 13: Ejercicio 1; T=11

En T=11 P06 finaliza su ejecución.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

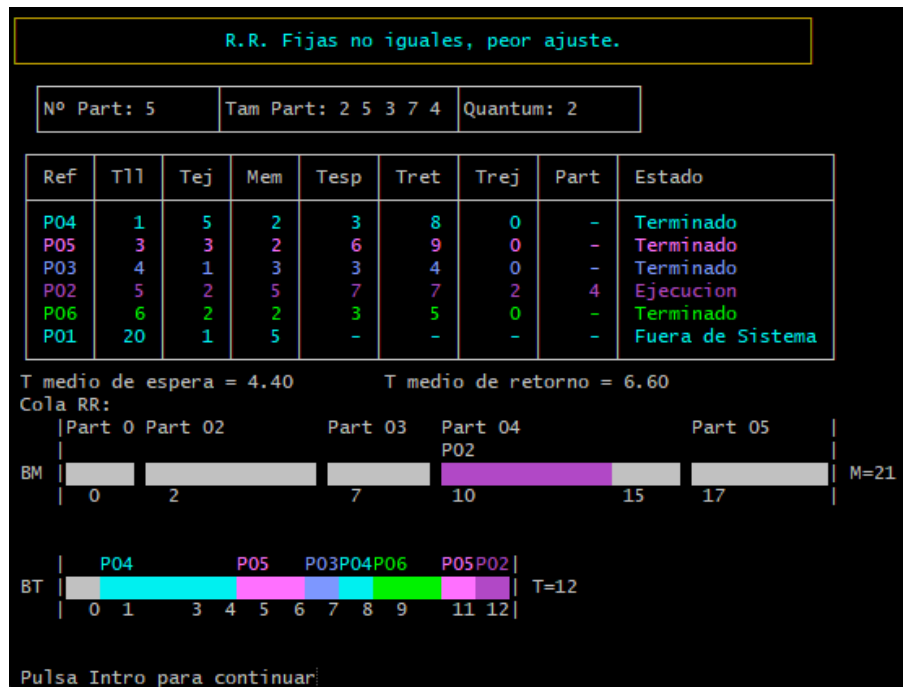


Figura 14: Ejercicio 1; T=12

En T=12 P05 finaliza su ejecución y P02 entra a CPU.

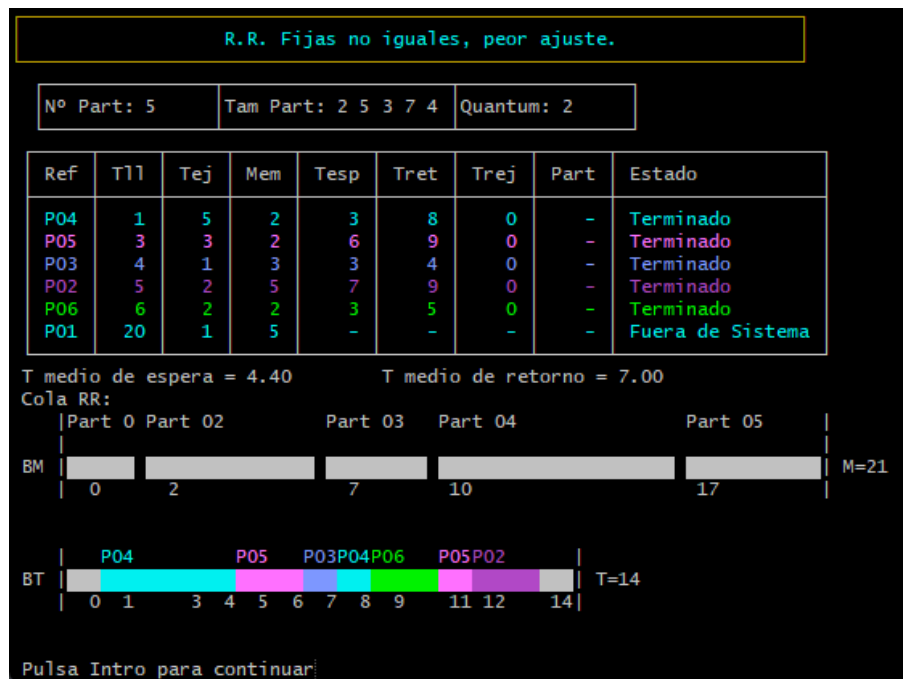


Figura 15: Ejercicio 1; T=14

En T=14 P02 finaliza su quantum y su ejecución, y se aprecia en la barra de tiempo cómo la CPU queda vacía.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

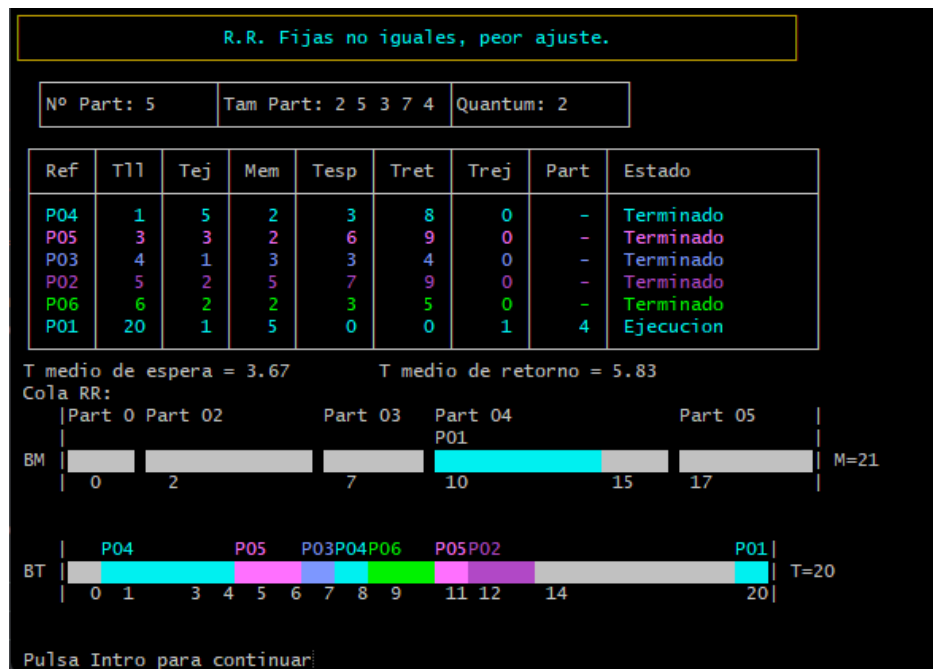


Figura 16: Ejercicio 1; T=20

Después no ocurre nada hasta T=20, cuando entra P01 al sistema.

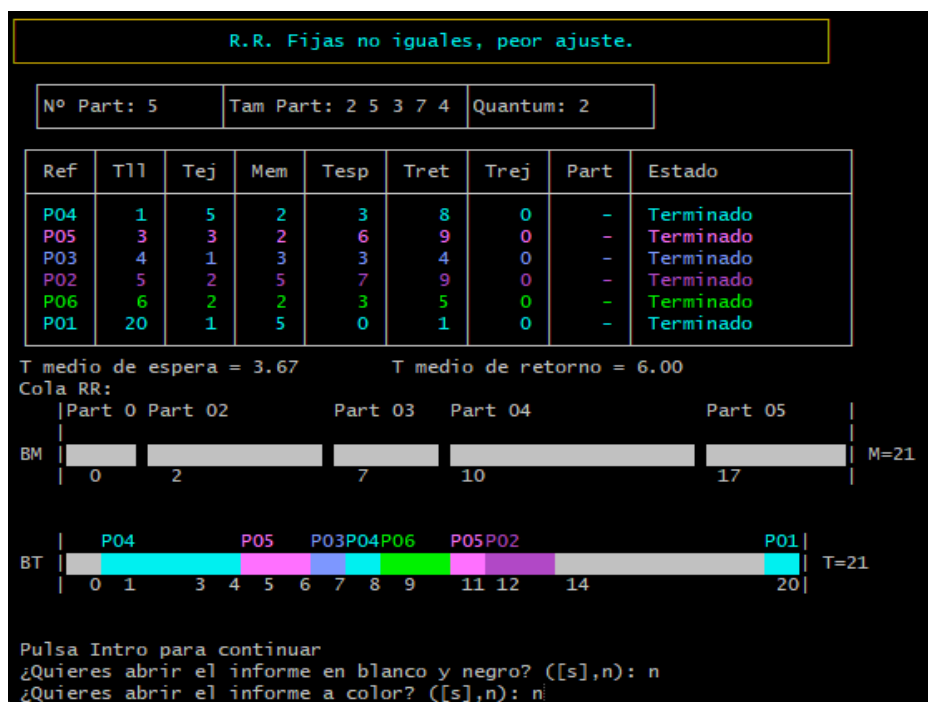


Figura 17: Ejercicio 1; T=21

En T=21 P01 finaliza y concluye el ejercicio.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

3.2.- Ejemplo 2

Ahora que ya hemos comprobado que el algoritmo se comporta como se espera, vamos a testear otros aspectos como el ajuste del tamaño de las unidades y los saltos de línea de las barras. Para ello ejecutaremos con los siguientes datos:

Partición:	P01	P02	P03	P04	P05	P06
Tamaño:	4	3	9	5	6	3

Proceso:	T Llegada:	Ráfaga CPU:	Memoria:
P01	2000000	1	5
P02	4	5	3
P03	5	7	5
P04	1	6	2

La entrada de datos se puede realizar manualmente y seleccionar que se guarde en el fichero EjemploInforme2.txt.

Ahora veremos la ejecución del script con éstos datos:

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

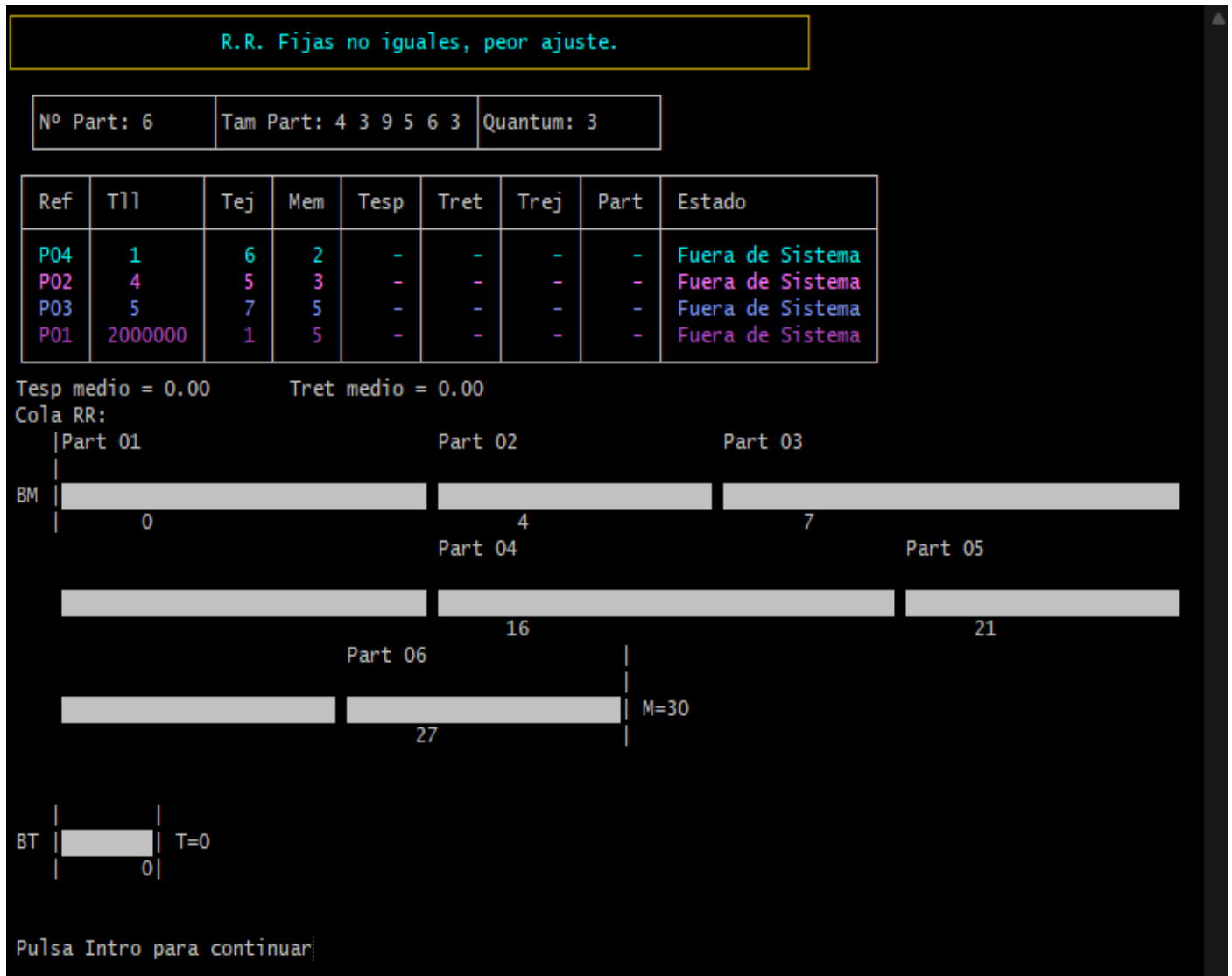


Figura 18: Ejercicio 2; T=0

Al iniciar la ejecución, se aprecia directamente cómo la barra de memoria ocupa varias líneas correctamente, y cómo el tamaño de cada unidad ha crecido para acomodarse al tiempo de llegada de P01 con 7 cifras.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

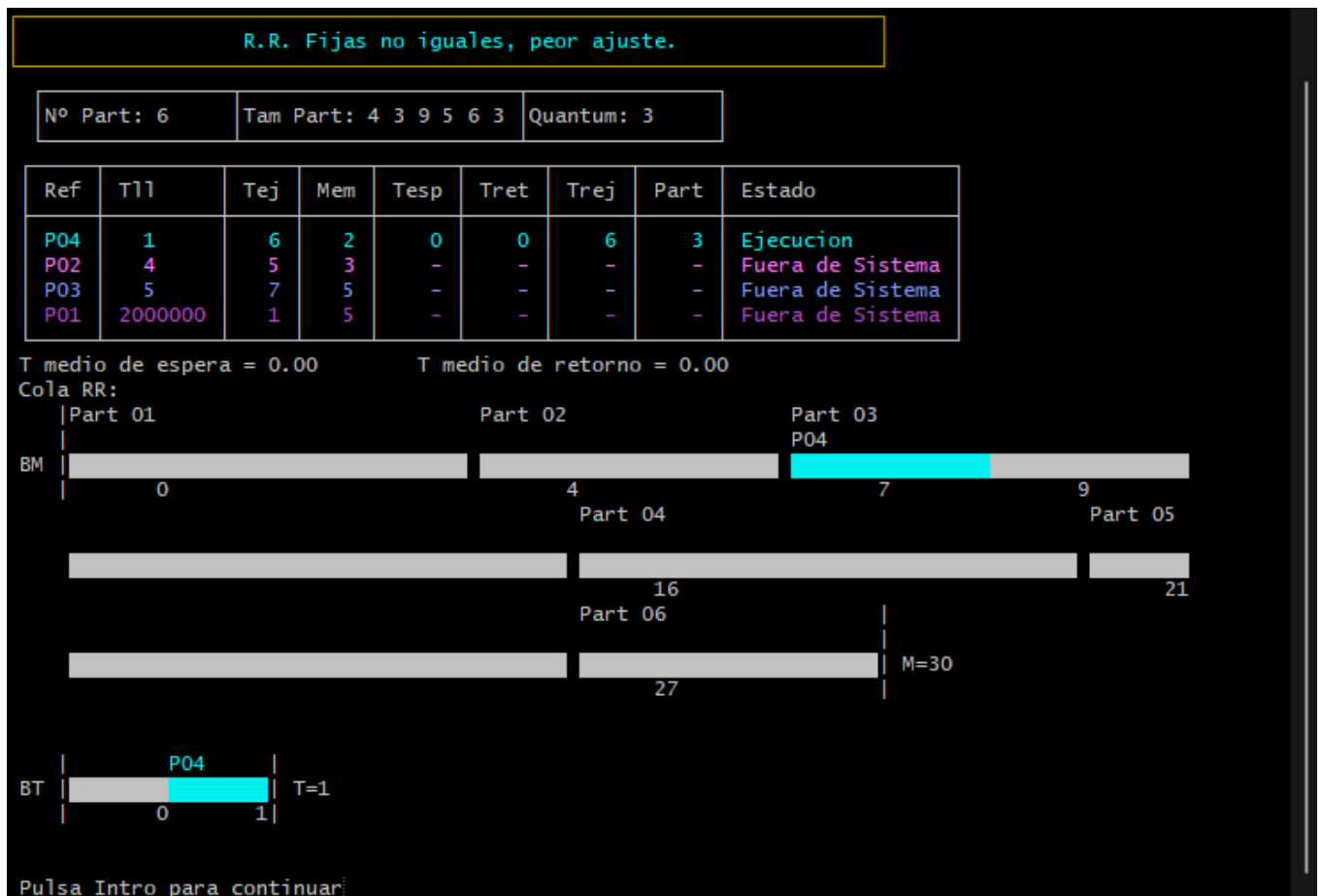


Figura 19: Ejercicio 2; T=1

Ahora he movido el límite de la ventana justo hasta el borde de memoria, y se ve cómo ha movido a la siguiente línea las últimas unidades de cada línea que no cabían por no poder poner un espacio a la derecha antes del final de pantalla.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.



Figura 20: Ejercicio 2; T=2

Recorto algo más la ventana y ahora se aprecia cómo todo el bloque del total de memoria pasa a la siguiente línea al no caber entero.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

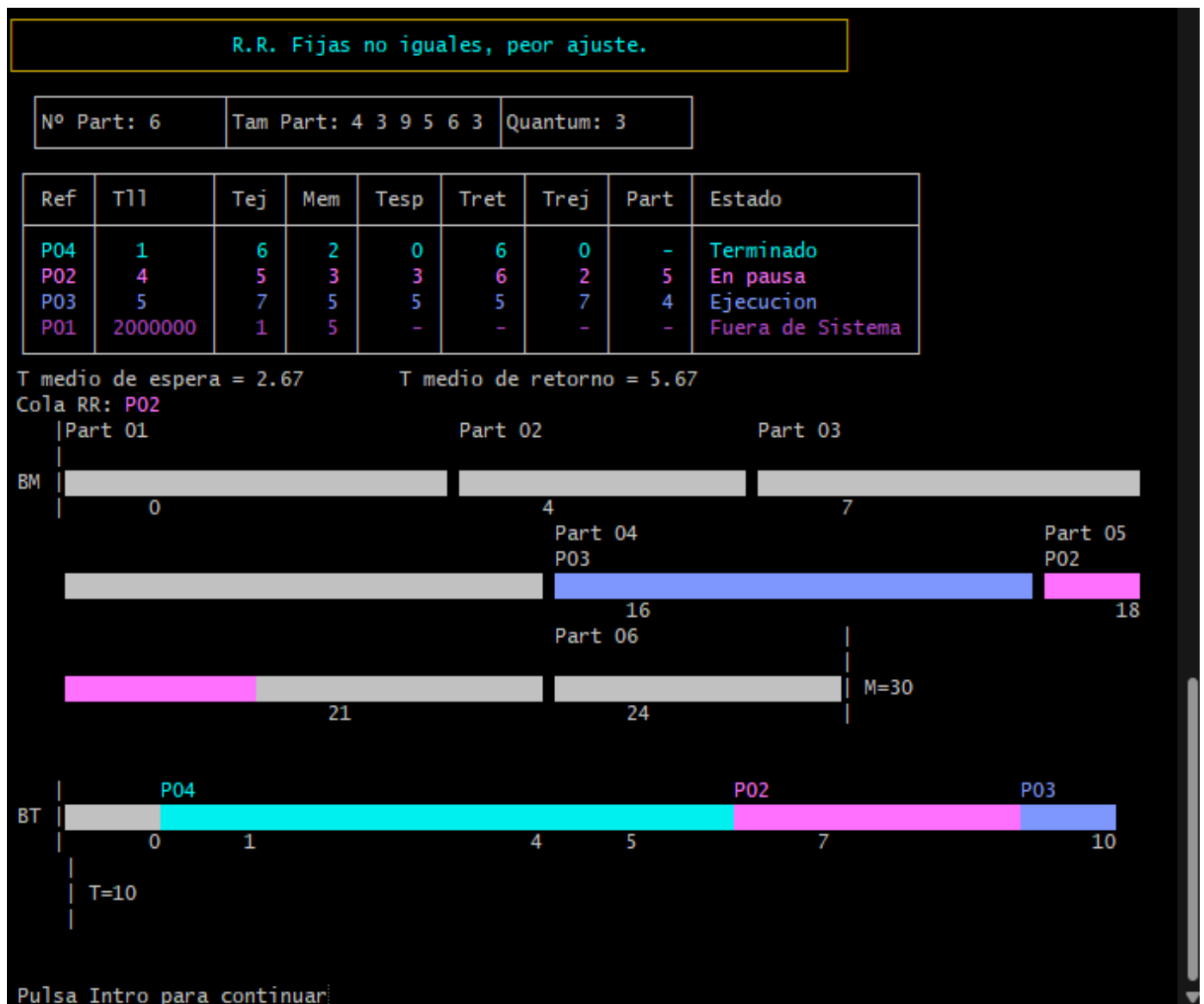


Figura 21: Ejercicio 2; T=10

Ajusto nuevamente el ancho de la ventana para cortar justo en medio del total de tiempo, y vemos como también pasa todo el bloque a la siguiente línea.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.



Figura 22: Ejercicio 2; T=18

Por último, comprobamos cómo la barra de tiempo se corta correctamente a la siguiente línea cuando una unidad no cabe entera.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

4.- Conclusiones

Queda comprobado el correcto funcionamiento del algoritmo al aplicar la gestión de procesos Round Robin, así como la gestión de memoria al peor ajuste.

También se comprueba la adecuación del script a las necesidades visuales de ajuste al entorno de ejecución.

En cuanto al algoritmo, ha supuesto un proceso de aprendizaje muy relevante. No un aprendizaje teórico, puesto que los conceptos necesarios, si bien son reforzados al implementar el algoritmo, es imprescindible conocerlos bien antes de su implementación. Pero sí un gran aprendizaje práctico sobre el lenguaje y la interfaz bash, con una gran utilidad si interesa continuar aprendiendo sobre Linux en cualquier ámbito, no sólo en gestión de sistemas.

Cabe destacar que la aplicación del peor ajuste es puramente teórica, dado que está más pensado para sistemas con particiones dinámicas, en los que el mayor espacio sobrante puede ser aprovechado por otros procesos. En un sistema real con particiones fijas, el peor ajuste no parece aportar ninguna ventaja, más bien al contrario.

Esto se pudo ver en el ejercicio 1, $T=5$, cuando el P02 tuvo que permanecer a la espera de una partición con tamaño suficiente, situación que se habría evitado con el mejor ajuste. Los primeros procesos habrían ocupado las particiones más pequeñas, permitiendo así la entrada a memoria de todos los procesos a su llegada al sistema y resultando en una mayor eficiencia temporal.

Práctica de control: Round Robin; Particiones Fijas no iguales, peor ajuste.

5.- Bibliografía

- Documentación académica correspondiente a la asignatura de Sistemas Operativos, relacionada con la gestión de procesos, gestión de memoria y bash.
- Diversas prácticas de alumnos de cursos anteriores de la asignatura, facilitadas en la plataforma virtual para tal efecto.
- Códigos de color y formato en bash:
<https://gist.github.com/JBlond/2fea43a3049b38287e5e9cefc87b2124>
- Manejo y representación de variables en bash:
<https://atareo.es/tutorial/scripts-en-bash/variables-en-bash/>
- Manejo y representación de arrays en bash:
<https://atareo.es/tutorial/scripts-en-bash/arrays-en-bash/>
- Manipulación de ficheros y archivos en bash:
https://www.zeppelinlinux.es/mover-copiar-renombrar-y-borrar-directorios-y-archivos-en-linux/#an_indice
- Información sobre el comando tput:
<https://www.slayerx.org/2020/05/08/tput-para-mejorar-tus-scripts-en-bash/>
- Información sobre el comando printf y sus opciones:
<https://linuxize.com/post/bash-printf-command/>
- Uso y ejemplos del comando awk:
<https://geekland.eu/uso-del-comando-awk-en-linux-y-unix-con-ejemplos/>