2022

UNIVERSIDAD DE BURGOS

Práctica de Control

SISTEMAS OPERATIVOS

HUGO GÓMEZ MARTÍN

UNIVERSIDAD DE BURGOS | 7 July 2022

Contenido

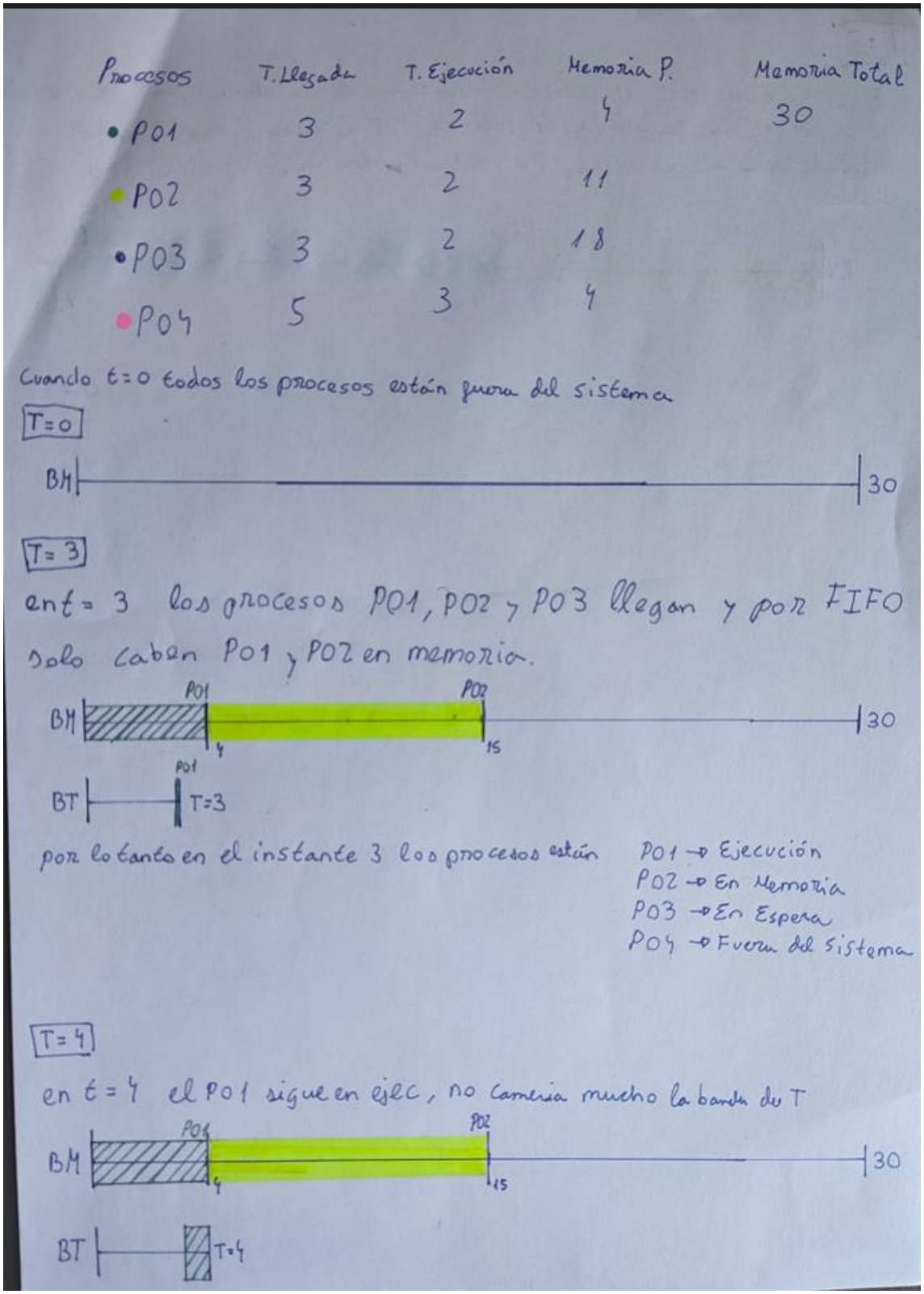
Algoritmo SRPT

En la Práctica de Control trabajamos con SRPT (Shortest Remaining Process Time first) donde las particiones de memoria son según necesidades, además la memoria es no continua y reubicable.

El algoritmo es SRPT y como vimos en el temario, es el modelo apropiativo de SJF, es decir, que el procesador (CPU) ejecuta el proceso más corto que está en memoria principal. Si llega a memoria otro proceso más corto (en proceso de duración), saca el que estaba ejecutando e introduce ese proceso mas corto en CPU y así sucesivamente.

Cuando ese proceso que sale de la CPU sin terminar para cederle el puesto al otro proceso que tiene preferencia (ya que es mas corto en duración), el proceso que sale pasará de tener un estado de **ejecución** a estado de **pausa** y el que entra pasará a estar en estado de **ejecución**, los demás estados que estén en memoria estarán en **espera** y los que están fuera del sistema tendrán el estado de **fuera del sistema**.

Ejemplo a mano

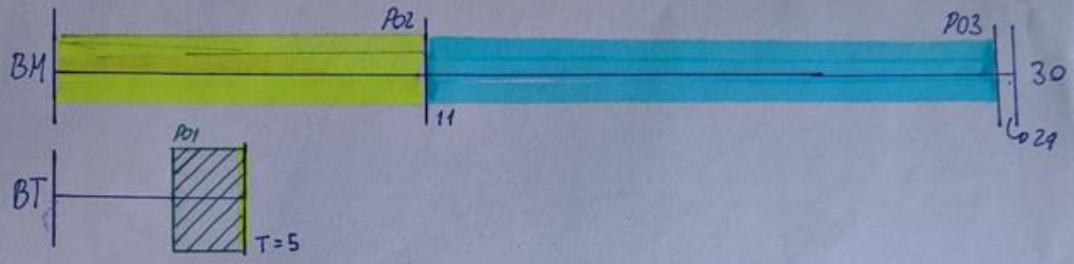


T=5

En el instante T=5 el proceso P01 acaba de ejecutarse y sale de la memoria.

Para evitar la fragmentación interna P02 se reubica y deja espacio libre suficiente para que el proceso P03 entre en moria.

P04 entra en el sistema

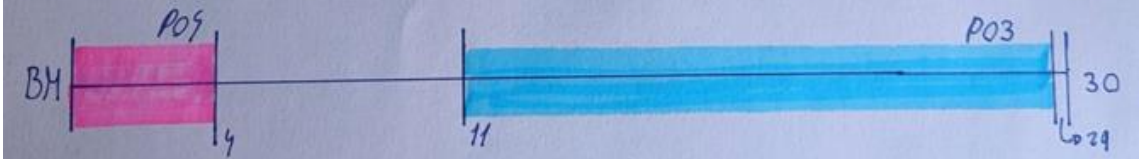


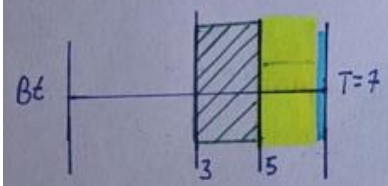
por lo tanto en el instante 5 los procesos están

P01 → Terminado
P02 → En ejecución
P03 → En memoria
P04 → En Espera

T=7

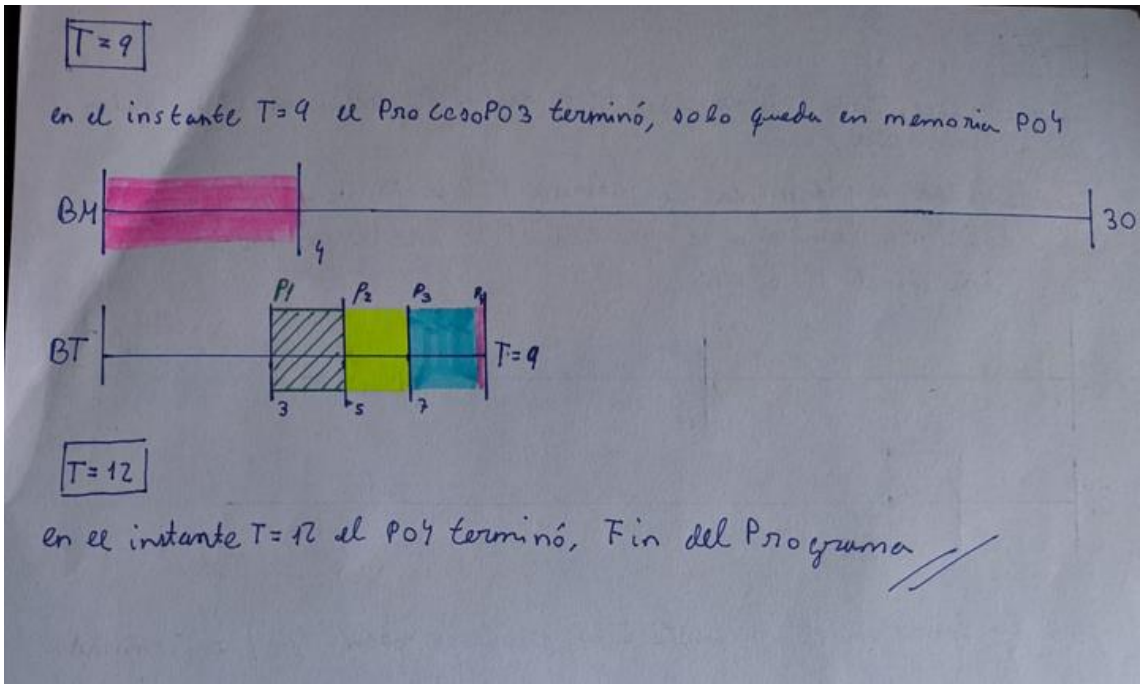
En el instante T=7 el proceso P02 terminó, pasará a ejecutarse P03 y P04 como cabe en el espacio que dejó P02 no hace falta reubicar





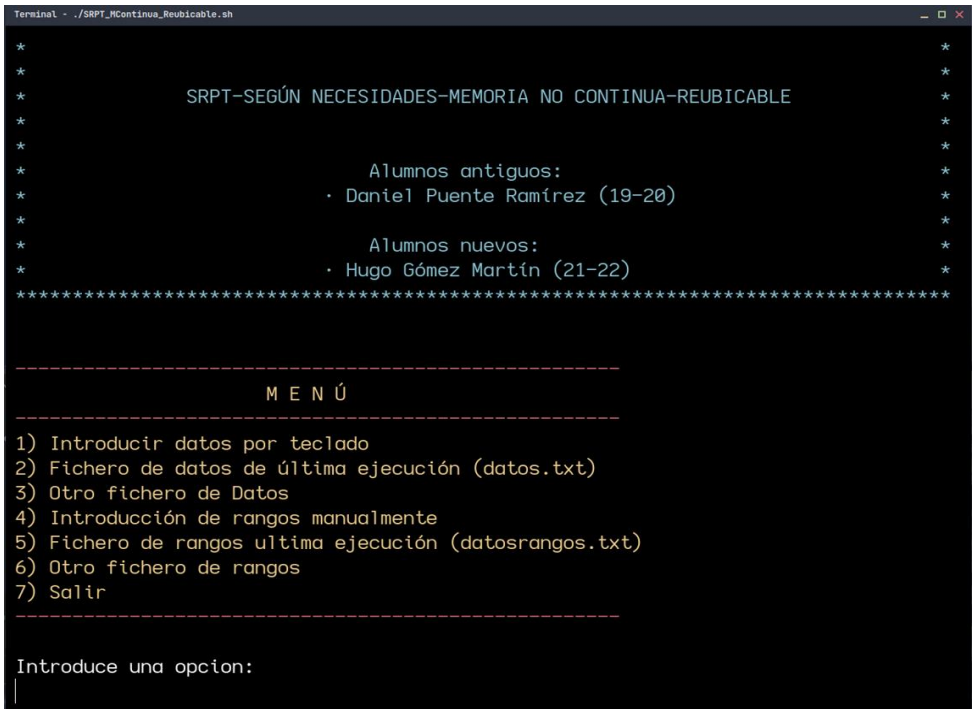
por lo tanto en el instante 7 los procesos están

P01 → Terminado
P02 → Terminado
P03 → En ejecución
P04 → En Memoria

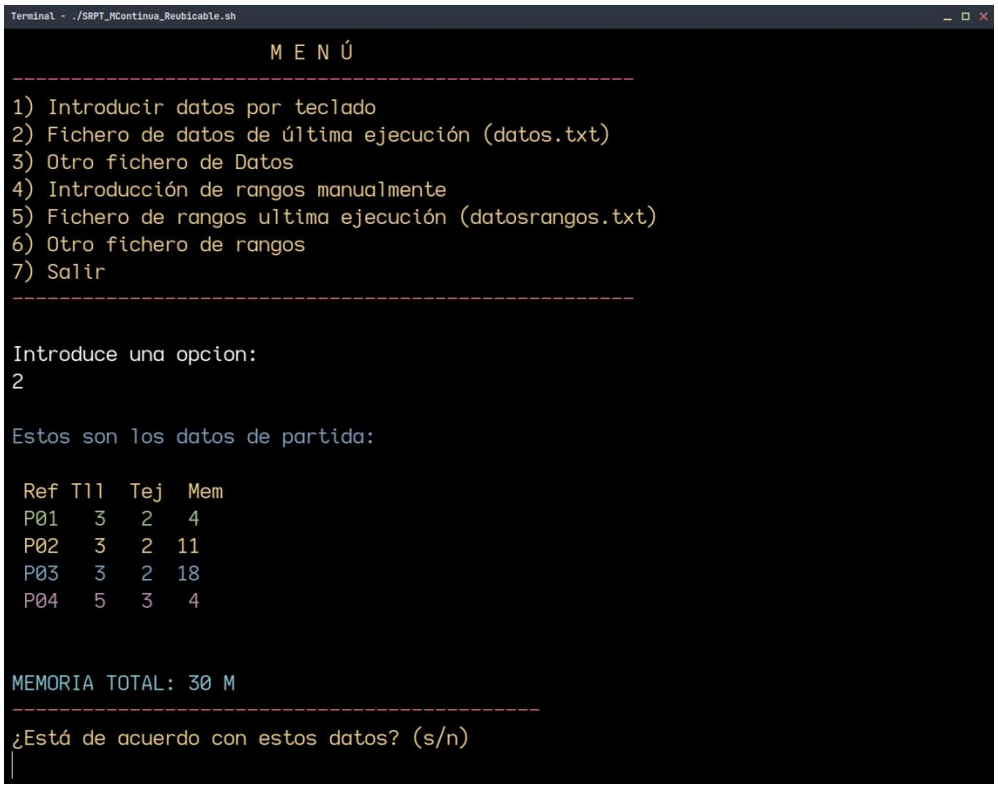

T=9
en el instante T=9 el Proceso P03 terminó, solo queda en memoria P04
BM
4
30
P1
P2
P3
P4
BT
T=9
3
5
7
T=12
en el instante T=12 el P04 terminó, Fin del Programa

Ejemplo con capturas de Linux

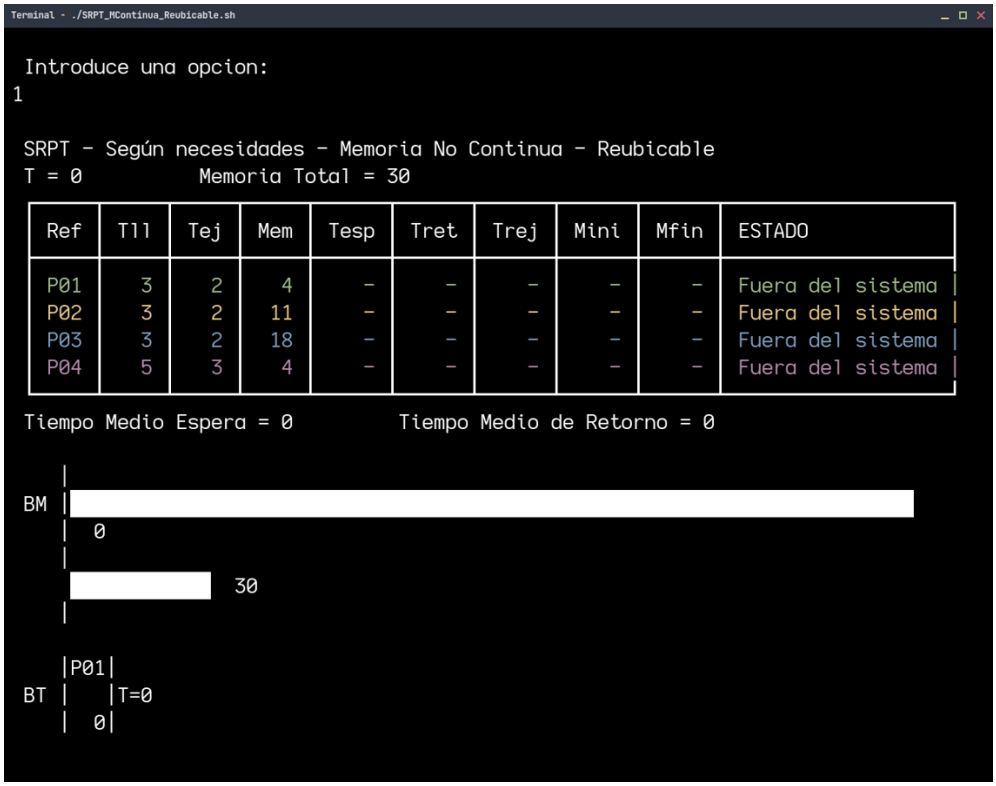
A continuación, se muestran capturas de la ejecución del ejercicio hecho a mano anterior, por el script en Linux, se comentan los sucesos más importantes.



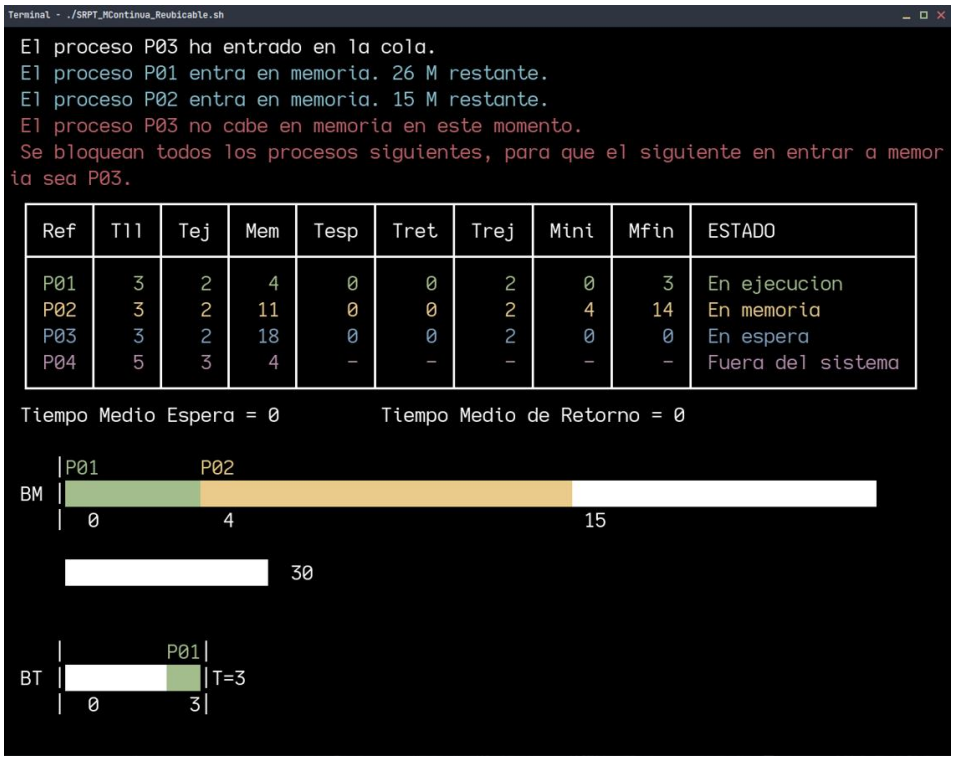
Al iniciar el programa nos aparecerá un menú donde tenemos todas las opciones permitidas por el programa



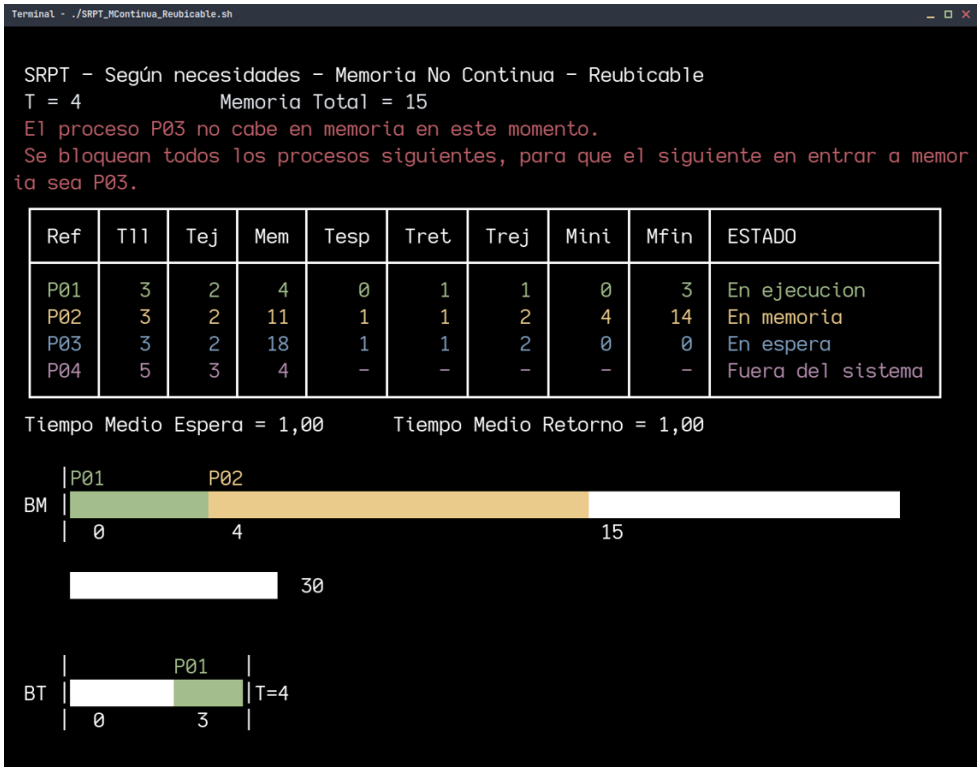
Selecciono la opción donde tenia guardados los datos en datos.txt



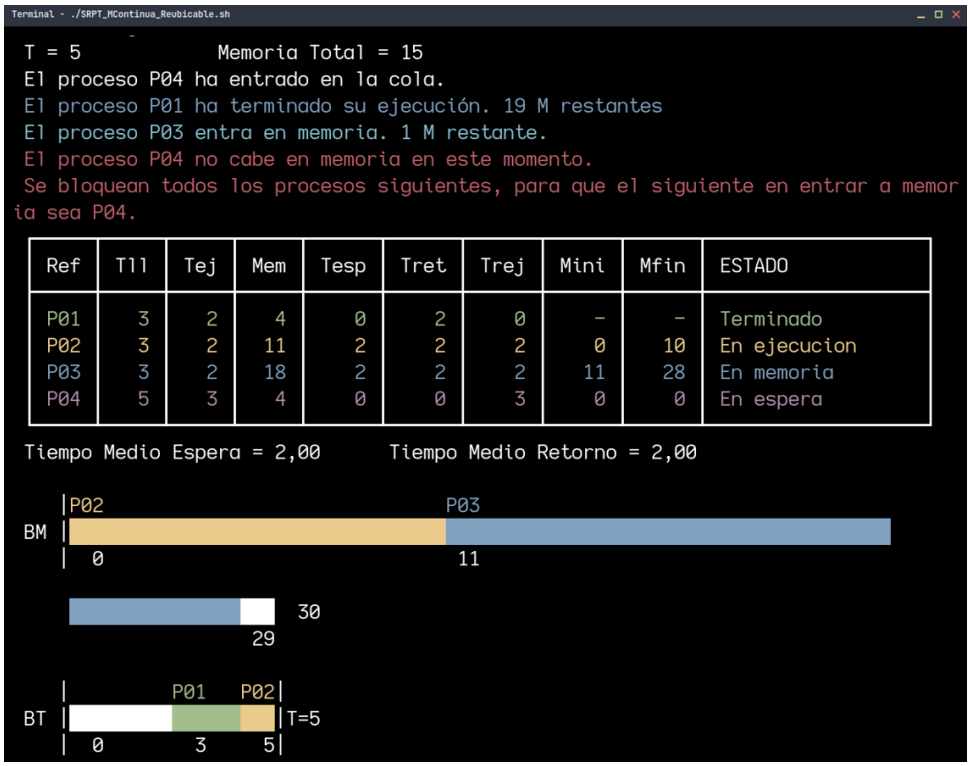
Aquí podemos ver los datos en el instante 0 antes de iniciar, todos los procesos fuera del sistema



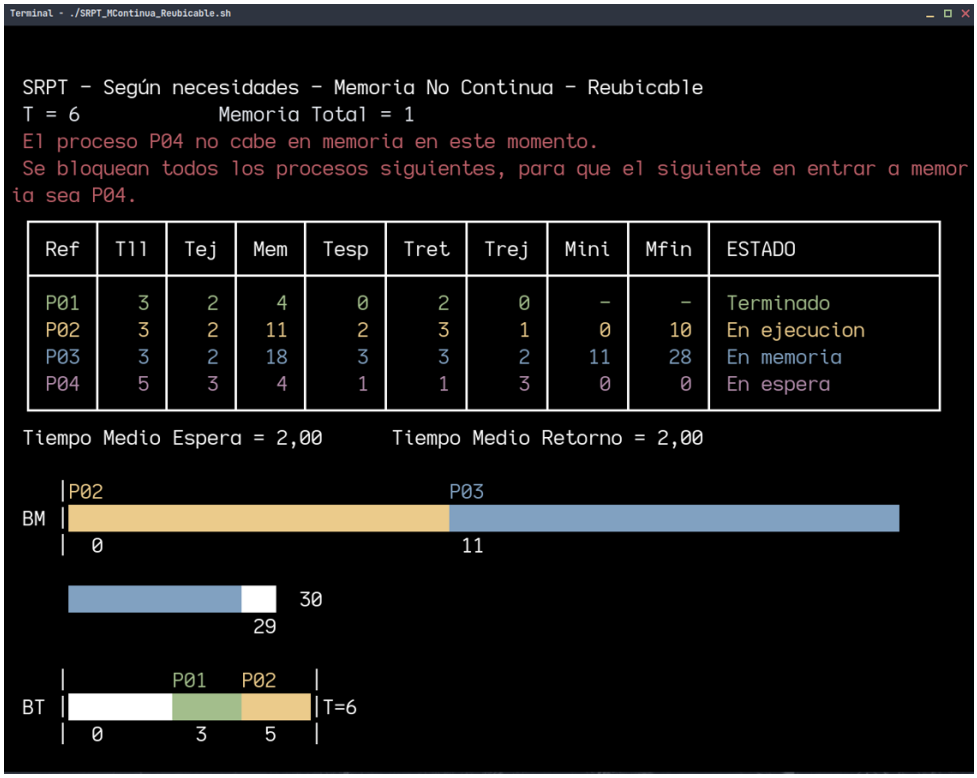
Podemos observar que en el instante 3 el proceso 01 y 02 entran en memoria, P03 no ya que no cabe y al ser FIFO entran en orden de entrada

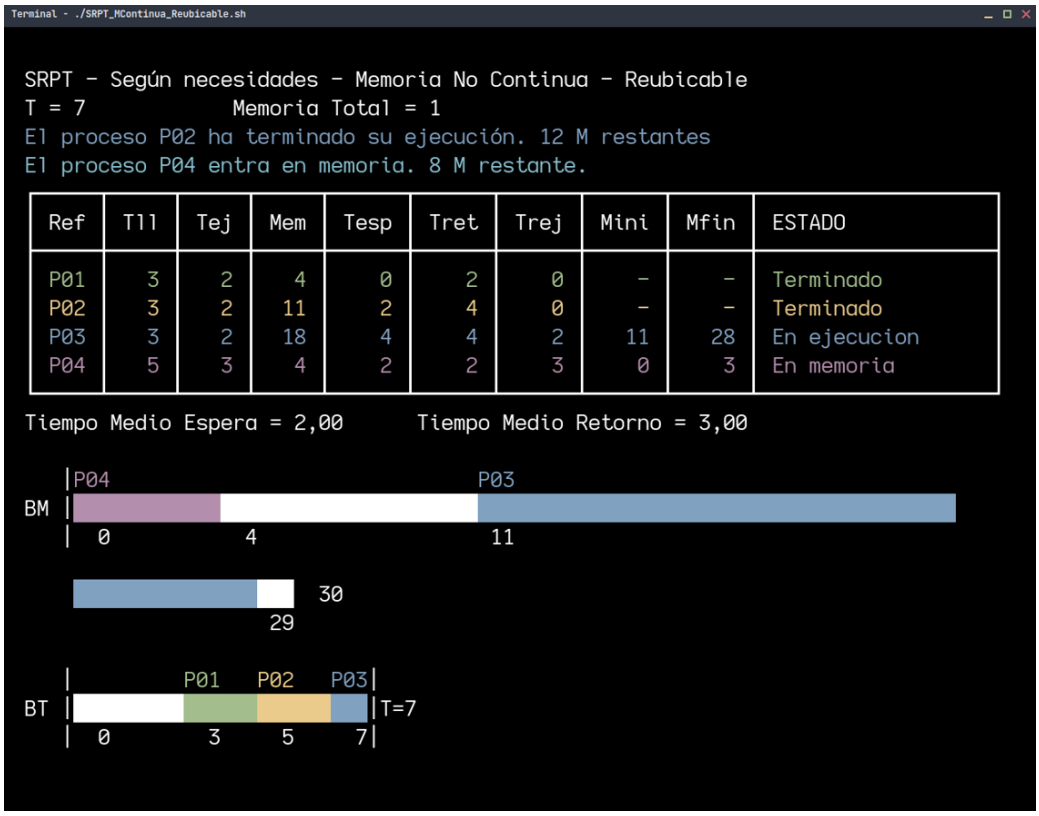


En el instante T=4 el P01 está en ejecución mientras P02 sigue en memoria, no cambió mucho en una unidad de tiempo

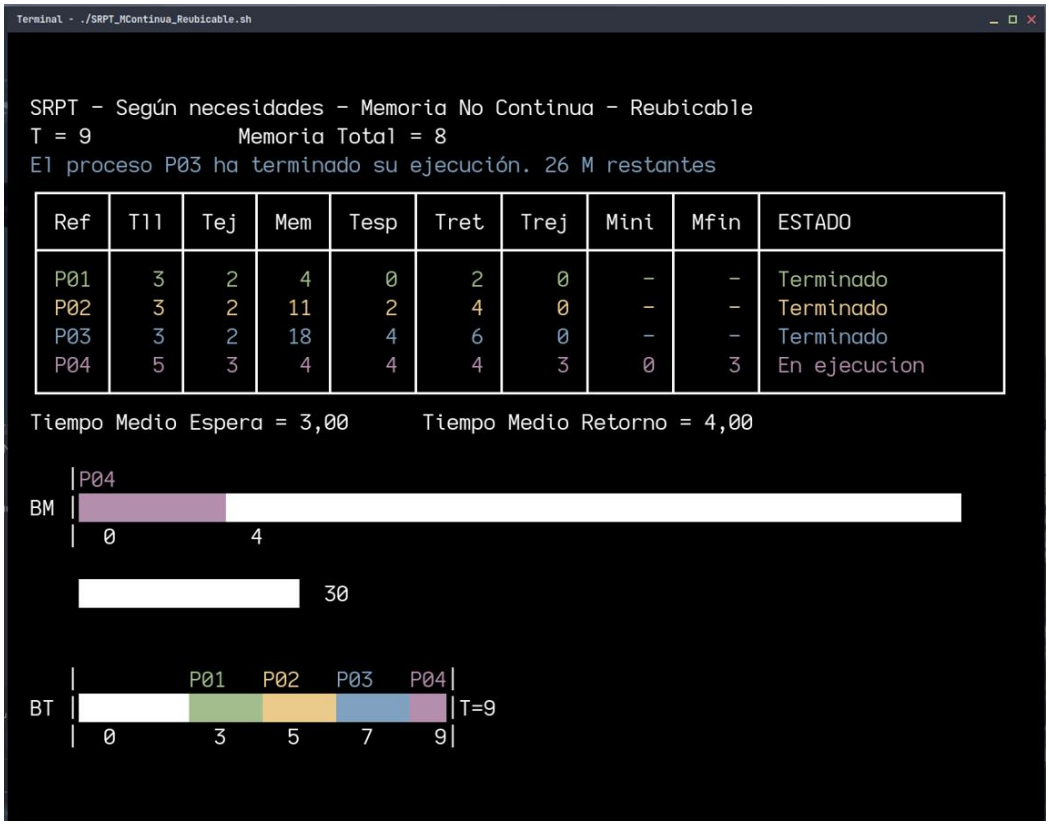


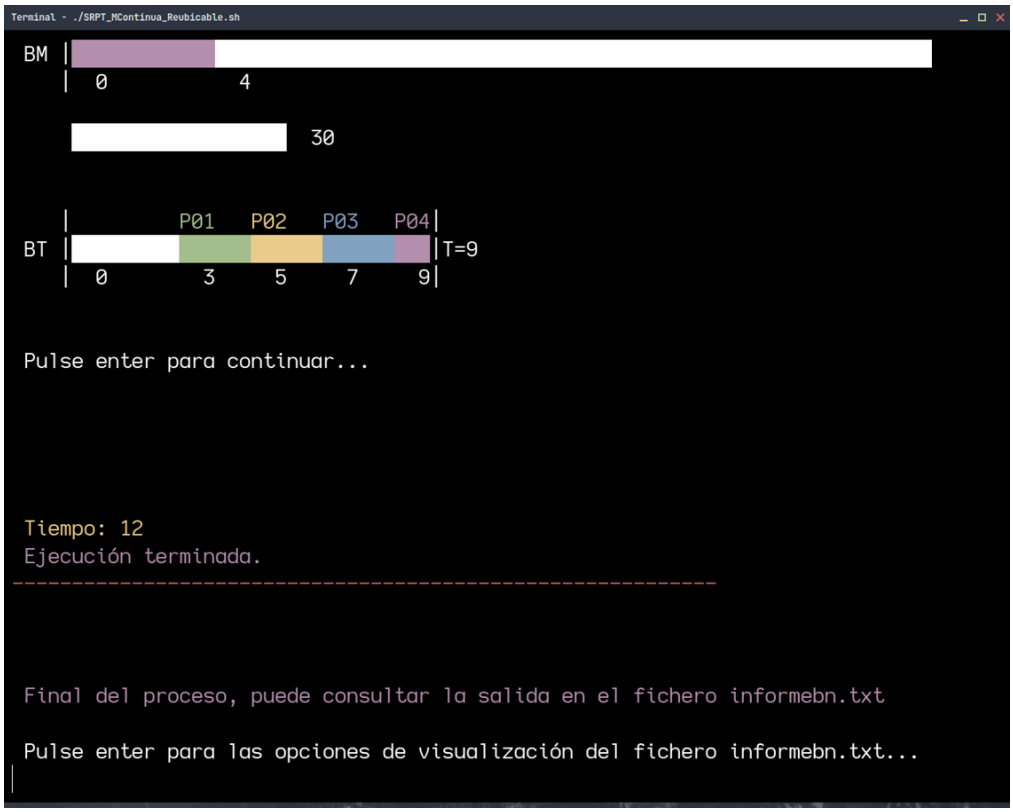
Con el proceso P01 ya finalizado P02 se reubica para dejar espacio a P03. P02 se ejecuta y P03 entra en memoria





Una vez el proceso P02 ha acabado P03 no hace falta que se reubique ya que P04 cabe en el espacio que dejo el proceso P02





En el instante 12 termina la ejecución, no hay mas procesos para ejecutar.

Modificaciones hechas

Se exponen las modificaciones hechas por el alumno en no siguiendo un orden de relevancia concreto.

1. Cambiar los colores de los caracteres del programa dependiendo si el fondo de la terminal es blanco o negro o cualquier otro color para garantizar así un alto contraste y que sea así perfectamente legible por parte del usuario.

2. Encuadrar la tabla de datos de los procesos mientras se ejecuta el algoritmo con sus respectivos iconos específicos para hacer la tabla totalmente lisa y perfecta.

3. Establecer antes de ejecutar el algoritmo un menú de ejecución con 3 opciones y sus respectivas validaciones:
    a. 1º opción: Por eventos (pulsando enter)
    b. 2º opción: Por eventos automático (estableciendo el usuario la unidad de tiempo que desee)
    c. 3º opción: Completa (ejecuta el algoritmo de golpe todo el programa hasta finalizar)

4. Establecer después de las bandas tanto de tiempo como de memoria una línea vertical compuesta de tres "|" (uno encima del otro) que sirve para delimitar la banda.

5. Establecer después de la línea que limita las dos bandas la información del total de memoria (en la banda de memoria) ej: |M=20 y la información del tiempo (modificándose cada vez que se aumente ese tiempo) |T=3.

6. Corrección de memoria total además de la implementación del nombre del algoritmo antes de la tabla de los procesos.

7. Implementar todos los .sh externos que dependían del script principal a funciones (ya el script no depende de ningún .sh externo).

8. Cambiar el nombre a los ficheros del informe a informebn.txt e informecolor.txt.

9. Realizar correctamente todas las opciones del fichero correctamente:
    a. Opción 1: además de introducir los datos nos pide donde guardar los datos (si en la opción 2 o la 3 del menú, si se guarda en la 2 se guarda e datos.txt y si se guarda en la 3 nos pedirá en qué fichero queremos guardarlo
    b. Opción 2: nos lee los datos de datos.txt
    c. Opción 3: nos lee los datos del fichero que le introduzcamos
    d. Opción 4: realiza lo mismo que la opción 1 pero con los ficheros aleatorios y esta vez dándonos como opciones donde guardar los rangos en la 5 o la 6 (opciones). Una vez guardadas nos pregunta donde queremos guardar los datos sacados de los rangos (datos.txt, datos2.txt, datos3.txt)
    e. Opción 5: lee los rangos de datos.txt y saca los datos de esos rangos para preguntar dónde queremos guardarlos (datos.txt, datos2.txt, datos3.txt)

f. Opción 6: lee los rangos del archivo que pasó al usuario, saca los datos de esos rangos y pregunta dónde queremos guardarlos
g. Opción 7: Sale del programa

10. Resumen rangos (opción 4) con los datos de los rangos.

11. He quitado unos cuantos read enters que realmente no servían para nada y molestaban a la hora de ejecutar el algoritmo.

12. He validado opciones de los menús

13. Si has escrito mal el nombre del fichero en las opciones 1 y 4 se crea un fichero nuevo con el nombre que pusiste automáticamente .

14. Pasar todos los ficheros .dat que había anteriormente a .txt

15. Corrección de bugs y errores menores.

Conclusiones y opinión personal

Esta ha sido la competencia académica más compleja y larga a la cual me he enfrentado. Que manera de subir y bajar de las nubes a la hora de realizar esta práctica, realizaba una mejora y me cargaba otras dos que totalmente no tenían nada que ver.

Empecé esta práctica sabiendo algo de C pero mis conocimientos a la hora de programar en bash eran muy escasos, dediqué unas 10 horas solamente a revisar todo el código y al intentar aprender lo básico de bash (es verdad que al revisar el código terminas aprendiendo por ejemplo a realizar bucles o por ejemplo saber cómo se una una variable). Las siguientes 60 horas las dediqué a realizar todas las mejoras, muchas veces me quedé hasta las 3 de la mañana para tener ese momento de lucidez y conseguir avanzar en este trabajo.

José Manuel tenia razón al decir “La practica de control os va a llevar por el camino de la amargura” o “empezad cuanto antes o os arrepentiréis”, totalmente es así, recomiendo empezarla nada más elegir el script y no hagáis como yo para evitar enloquecer. En muchos momentos quise tirar la toalla, cuando por ejemplo no me di cuenta de que no puse un continuar=NO al final de las opciones del menú, más de 1 hora y media estuve buscando que era lo que fallaba, miré en las copias de seguridad etc y lo único que me faltaba era eso, cuando lo vi me quería insultar a mí mismo.

Como conclusión gracias a este trabajo he adquirido grandes conocimientos de bash y de comandos de Unix, también he aprendido un poco mas sobre mi algoritmo.