
 <p>Universidad Carlos III de Madrid</p>	<p>Departamento de Informática Grado en Ingeniería Informática Sistemas Operativos</p> <p>Prueba de Evaluación Continua 19 de Marzo de 2013-B</p>	
---	---	---

**ATENCIÓN:**

- Lea atentamente todo el enunciado antes de comenzar a contestar.
- Dispone de 90 minutos para realizar la prueba.
- No se podrán utilizar libros ni apuntes, ni calculadoras de ningún tipo.
- Los teléfonos móviles deberán permanecer desconectados durante la prueba ( apagados, no silenciados).
- Solamente se corregirán los ejercicios contestados con bolígrafo. Por favor no utilice lápiz.

**APELLIDOS:**

**NOMBRE:**

**NIA:**

**GRUPO:**

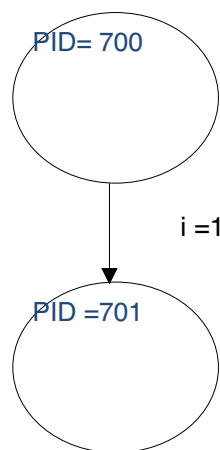
**Ejercicio 1: Responda a las siguientes preguntas de teoría (4 puntos)**

- [1 punto] Explique tres medidas de valoración en la planificación de procesos.
  - Utilización de CPU:  
Porcentaje de tiempo que se usa la CPU.  
Objetivo: Maximizar.
  - Productividad:  
Número de trabajos terminados por unidad de tiempo.  
Objetivo: Maximizar.
  - Tiempo de retorno ( $T_q$ )  
Tiempo que está un proceso en el sistema. Instante final ( $T_f$ ) menos instante inicial ( $T_i$ ).  
Objetivo: Minimizar.
  - Tiempo de servicio ( $T_s$ ):  
Tiempo dedicado a tareas productivas (cpu, entrada/salida).  $T_s = T_{CPU} + T_{E/S}$   
Tiempo de espera ( $T_e$ ):  
Tiempo que un proceso pasa en colas de espera.  
 $T_e = T_q - T_s$
  - Tiempo de retorno normalizado ( $T_n$ ):  
Razón entre tiempo de retorno y tiempo de servicio.  $T_n = T_q / T_s$   
Indica el retardo experimentado.
- [1 punto] Cuando se produce un cambio de contexto voluntario por parte de un proceso.

Se produce un cambio de contexto de forma voluntaria cuando un proceso hace un `exit()`. También cuando se realiza una llamada bloqueante como un `read()`.

- [1 punto] Suponiendo que el PID de este proceso es 700, indicar la jerarquía de procesos generada identificando cada proceso con un valor PID consecutivo al del padre, y escribir los mensajes que aparecerían en pantalla

```
for (i=1; i<4; i++){
    pid=fork();
    if (pid==0){ //Hijo
        printf ("Hijo %d, pid =%d, ppid=%d\n",i, getpid(),getppid());
        exit(0);
    }
    else
        printf ("El padre ha creado %d hijos \n",i);
        exit (0);
}
```



Hijo 1, pid=701, ppid=700

El padre ha creado 1 hijos

Esta es una de las posibles soluciones, el orden de los mensajes puede variar

- [1 punto] Indique en qué situaciones un proceso puede realizar las siguientes transiciones de estado:  
Listo en 2º plano -> Bloqueado en 1º plano  
Ejecución -> Bloqueado.

Listo en 2º plano -> Bloqueado en 1º plano

Esta transición no puede darse, si un proceso está listo en 2º plano la única transición posible es a listo en 1º plano.

Ejecución -> Bloqueado

Si el proceso necesita un recurso del cual no dispone para continuar ejecutándose pasa al estado bloqueado hasta que dicho recurso está disponible.

## Ejercicio 2 [3 puntos]

Un sistema operativo utiliza un planificador. En un instante determinado no hay ningún trabajo en ejecución y se desean ejecutar trabajos cuyos tiempos de llegada al sistema son los siguientes:

Proceso	Tiempo de llegada al sistema	Tiempo de ejecución	Prioridad
A	0	2	1
B	0,999999999	5	0
C	1,999999999	6	0
D	1,999999999	2	1
E	3,999999999	4	1

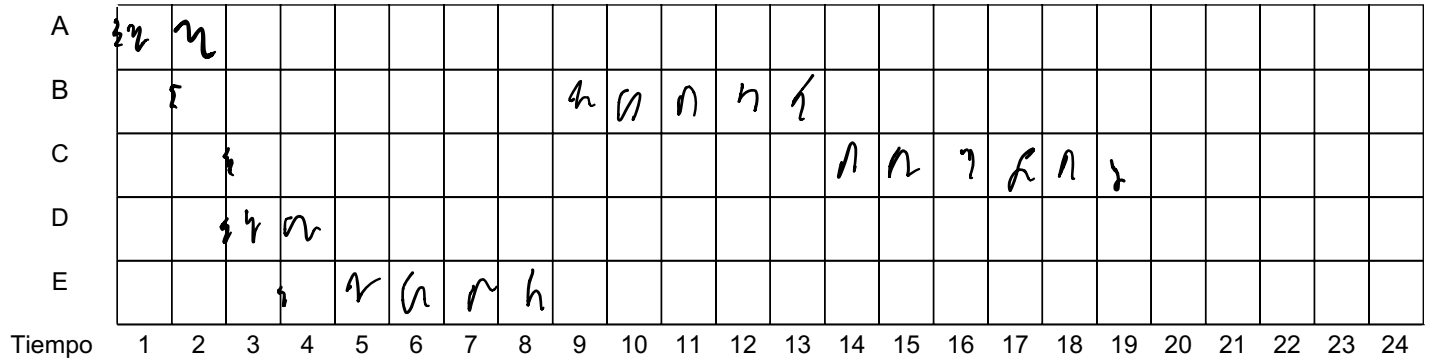
Para los cálculos redondear los tiempos de llegada al valor inmediatamente superior. Sabiendo que la prioridad más alta es 1. Se pide rellenar las tablas en los siguientes casos:

- Política de planificación FCFS (First to Come First to Server)
- Política de planificación Round-Robin con rodaja de 3.

Para las dos posibilidades, se pide:

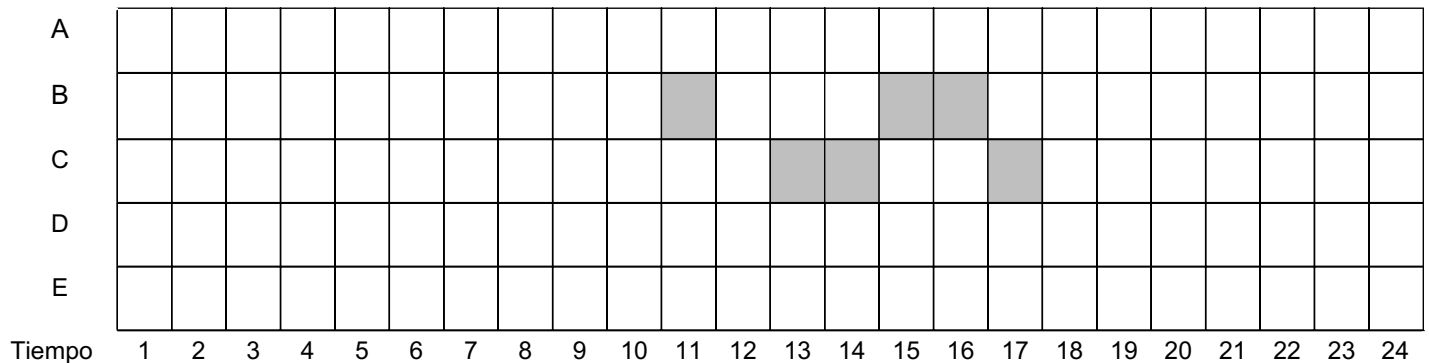
- Determine el momento de finalización de cada proceso.
- Determine el tiempo que cada proceso ha estado en el sistema (tiempo de retorno).
- Determine el tiempo de servicio y el tiempo de espera de cada proceso.

a) Política de planificación FCFS.





Proceso	Periodo de finalización	Tiempo de retorno	Tiempo de servicio	Tiempo de espera
A	2	2	2	0
B	13	11	5	6
C	19	17	6	11
D	4	2	2	0
E	8	4	4	0

b) Política de planificación Round-Robin con rodaja de 3.



Proceso	Momento de finalización	Tiempo de retorno	Tiempo de servicio	Tiempo de espera
A	2	2	2	0
B	16	15	5	8
C	19	17	6	11
D	4	2	2	0
E	8	4	4	0

 <p>Universidad Carlos III de Madrid</p>	<p>Departamento de Informática Grado en Ingeniería Informática Sistemas Operativos</p> <p>Prueba de Evaluación Continua 19 de Marzo de 2013-B</p>	
---	---	---

### Ejercicio 3 [3 puntos]:

Dibuje el esquema de procesos y codifique , usando el lenguaje C, un programa que permita ejecutar el siguiente mandato: `ls | cat > f1`

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
Int main (int argc, char *argv[])
{
    Int fd[2];
    Int fid;
    pipe(fd);
    pid=fork();
    switch (pid)
    {
        case -1:
            perror (error al crear proceso hijo);
            break;
        case 0:
            fid= open (f1, O_WRONLY | O_CREAT | O_TRUNC, 0751);
            close (1);
            dup(fid);
            close (0);
            dup(fd[0]);
            close(fd[0]);
            close(fd[1]);
            execlp(cat,cat,NULL);
            exit(-1)
        default:
            close (1);
            dup(fd[1]);
            close(fd[0]);
            close(fd[1]);
            execlp(ls,ls,NULL);
            exit(-1)
    }
}
```