

# SISTEMAS OPERATIVOS

## Certamen #1

### Pauta de Corrección

Wenceslao Palma <wenceslao.palma@ucv.cl>

1. Responda las sgtes preguntas. Justifique cada una de sus respuestas.

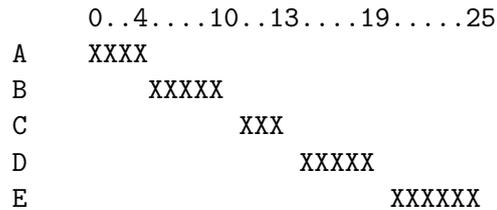
- (a) (5 ptos.) Cuál es la diferencia entre `|` y `>` ?  
R.: `|` permite que la salida de un proceso se tome como entrada de otro proceso y `>` redirecciona la salida de un proceso hacia un archivo.
- (b) (5 ptos.) Cuáles son las tareas que realiza el kernel de Unix al momento de crear un proceso usando `fork()`?  
R.:  
(a) asignar al nuevo proceso espacio en la tabla de procesos.  
(b) asignar un PID al proceso.  
(c) realizar una copia del contexto del proceso padre.  
(d) incrementar contador de archivos del padre.  
(e) poner al hijo en estado ready.  
(f) retornar el PID al padre.
- (c) (5 ptos.) Cuál es la similitud y diferencia entre los algoritmos de planificación SRT y SPN?  
R.: en ambos algoritmos el itinerador de procesos escoge aquel con el menor tiempo restante y la diferencia está en que SRT es una versión expulsiva de SPN.
- (d) (5 ptos.) Cuál es la diferencia entre semáforos generales y binarios?  
R.: ambos tiene el mismo poder de expresión, sin embargo es más sencilla la utilización de semáforos generales cuando se necesita que varios procesos accedan al mismo tiempo a la sección crítica.

2. (20 pts.) Considere los siguientes datos:

Proceso	Llegada	Tiempo de Servicio
A	0	4
B	2	6
C	4	3
D	10	6
E	13	6

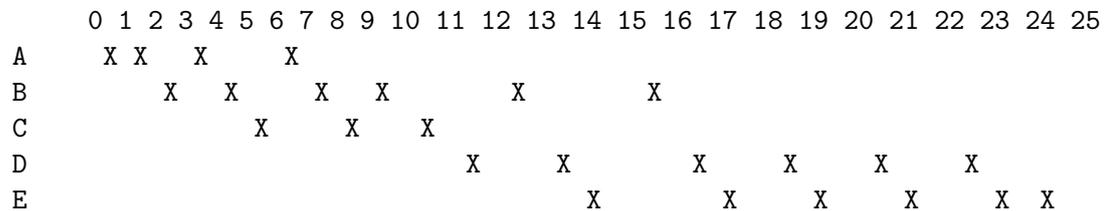
Muestre como los algoritmos FCFS y RR( $q=1$ ). Realice una tabla con los valores de Tiempo de Finalización,  $T_{retorno}$  y  $T_{retorno}/T_{servicio}$ . Para cada algoritmo realice el gráfico correspondiente. Con todo lo realizado comente.

FCFS (9 pts.)



	A	B	C	D	E	Media
Tfinalizacion	4	10	13	19	25	
Tretorno	4	8	9	9	12	8.4
Tr/Ts	1	1.3	3	1.5	2	1,76

RR ( $q=1$ ) (9 pts.)



	A	B	C	D	E	Media
Tfinalizacion	7	16	11	23	25	
Tretorno	7	14	7	13	12	10.6
Tr/Ts	1.75	2.33	2.33	2.16	2	2.11

comentario (2 pts)

3. (20 pts.) Escriba un script usando bash y awk que elimine todos los archivos que se encuentran a partir del directorio actual y cuyo tamaño sea superior a 2MB. El script debe generar un reporte en un archivo de texto llamado out.txt que contenga la cantidad de archivos eliminados y la ubicación de cada uno de ellos dentro del árbol de directorio.

```
#!/usr/bin/bash

eliminar () {
  for archivo in `ls`; do
    if [ -d "$archivo" ]; then
      cd "$archivo"
      eliminar
      cd ..
    else
      size=`wc -c $archivo | awk '{print $1}'`
      if [ "$size" -gt "2097152" ]; then
        echo "eliminando.....$PWD/$archivo">>$salida
        i=`expr $i + 1`
      fi
    fi
  done
}
dir=$PWD
salida="$dir/out.txt"
i=0
eliminar
echo "nro de archivos eliminados: $i">>$salida
```