

1. Sombrea la respuesta correcta (0,25 c/u).

1) ¿Con qué comando se accede a los procesos en Windows /Linux?

- a) Processlist/process
- b) Tasklist/ps
- c) Tasklist/process
- d) Process/ps

2) ¿Qué información nos muestra PID?

- a) La ID del proceso padre
- b) ID del proceso
- c) El procesador que tiene asignado el proceso
- d) Ninguna de las anteriores

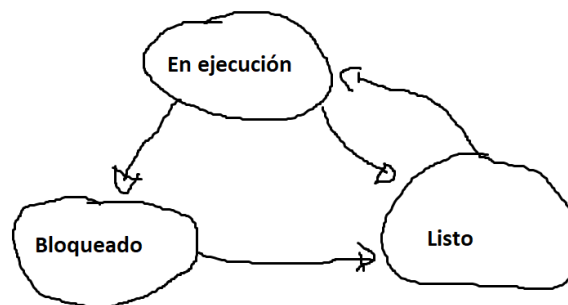
3) ¿Qué información nos da PPID?

- a) La ID del proceso padre
- b) ID del proceso
- c) El procesador que tiene asignado el proceso
- d) Ninguna de las anteriores

4) ¿Qué información nos da STIME

- a) Tiempo que lleva ejecutándose un proceso
- b) Hora a la que empezó a ejecutarse el proceso
- c) Nos permite hacer un Set Time
- d) Ninguna de las anteriores

2. Explica el estado de los procesos, sus posibles transacciones y haz el gráfico. (1 punto).



En ejecución a Bloqueado: el proceso se toma un momento a que ocurra el evento.

Bloqueado a Listo. Realiza el evento que el proceso estaba esperando.

Listo a Ejecución: El sistema operativo le da tiempo al CPU

Ejecución a Listo: Se le ha terminado el tiempo y el sistema operativo realiza el proceso.

3. Escribe un código en Java que permita abrir Chrome. (2 puntos).

```
package ejercicio3;  
  
import java.io.IOException;  
  
public class Principal {  
  
    public static void main(String[] args) throws IOException {  
        // TODO Auto-generated method stub  
        ProcessBuilder pb = new ProcessBuilder("cmd", "/c", "chrome");  
        //ProcessBuilder pb = new ProcessBuilder("cmd", "/c", "tasklist");  
  
        Process p = pb.start();  
        InputStream is=p.getInputStream();  
        int c;  
        while ((c=is.read()) != -1) {  
            System.out.print((char)c);  
        }  
        is.close();  
    }  
}
```

4. Escribe un código en Java que nos muestre por la consola de Eclipse los procesos que tengamos abiertos. (2 puntos).

```
1 package ejercicio4;
2
3 import java.io.IOException;
4
5
6 public class Principal {
7
8     public static void main(String[] args) throws IOException {
9
10         ProcessBuilder processBuilder = new ProcessBuilder("cmd", "/c", "taskl
11
12         Process proceso = processBuilder.start();
13         InputStream is=proceso.getInputStream();
14         int c;
15         while ((c=is.read()) != -1) {
16             System.out.print((char)c);
17         }
18         is.close();
19     }
20 }
```

Problems @ Javadoc Declaration Console ×

erminated> Principal (35) [Java Application] C:\Users\Emperador\p2\pool\plugins\org.eclipse.justj.openjdk.h

isHost.exe	3036	Services	0	5.636 KB
rhost.exe	2852	Services	0	173.920 KB
rhost.exe	2696	Services	0	6.112 KB
rhost.exe	3084	Services	0	7.996 KB
ieclxx.exe	3092	Console	1	14.560 KB
emory Compression	3172	Services	0	3.280 KB
rhost.exe	3216	Services	0	8.620 KB
rhost.exe	3240	Services	0	8.712 KB
rhost.exe	3248	Services	0	11.404 KB
rhost.exe	3452	Services	0	15.024 KB
rhost.exe	3552	Services	0	18.344 KB
rhost.exe	3576	Services	0	7.112 KB
rhost.exe	3584	Services	0	10.620 KB

5. Crea el cuadro de Bernstein e indica qué instrucciones son concurrentes. (2 puntos).

Instrucción 1: $p1 = a * \text{square}$;

Instrucción 2: $p2 = b * x$;

Instrucción 3: $\text{square} = x * x$;

Instrucción 4: $z = m1 + m2$;

Instrucción 5: $y = z + c$;

Las instrucciones que son concurrente son Conjunto I1 e I2, Conjunto I1 e I4, Conjunto I1 e I5, Conjunto I2 e I3, Conjunto I2 e I4, Conjunto I2 e I5, Conjunto I3 e I4 y Conjunto I3 e I5

La tabla la tengo subida aparte en formato png

6. Desarrolla las ventajas de la programación concurrente en los monoprocesadores. (1 punto).

La multiprogramación da la sensación de estar haciendo varios procesos a la vez.

- La multiprogramación maximiza el uso de la CPU, mientras que el multiproceso ahorra tiempo al ejecutar varios procesos.
- En multiprogramación los procesos no están diseñados para trabajar entre sí, mientras que en multiproceso se permite la comunicación.
- En la multiprogramación los recursos se comparten, mientras que en el multiproceso cada proceso tiene sus recursos asignados

7. Desarrolla los dos problemas inherentes a la programación concurrente. (1 punto).

- Exclusión mutua. Solo se permite a un programa de realizar su respectivo trabajo porque si no se satura.
- Condición de sincronización. Tiene la habilidad de cuando bloquear y cuando desbloquear el proceso mediante condiciones.