

# Practica-julio-2025.pdf



Mega\_0904



Sistemas Operativos



2º Grado en Ingeniería Informática



Escuela Politécnica Superior  
Universidad de Alicante



[Accede al documento original](#)



Escuela de  
Organización  
Industrial

Contigo que evoluciones.  
Contigo que lideras. Contigo que transformas.

**Esto es EOI.  
Mismo propósito,  
nueva energía.**



Descubre más aquí



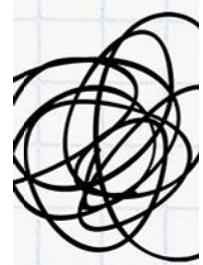
**EOI** Escuela de  
Organización  
Industrial

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato  
→ Planes pro: más coins

pierdo  
espacio



Necesito  
concentración

ali ali ooooh  
esto con 1 coin me  
lo quito yo...

wuolah

## **Sistemas Operativos**

### **2024-2025**

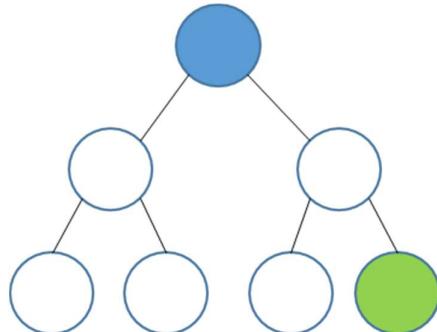
Práctica extraordinaria de julio

wuolah

# EVALUACIÓN

## Ejercicio1. Gestión básica de procesos. (3 puntos)

- a) Realizar un programa llamado **arbolB.c** que reciba un argumento. El programa tendrá que generar un árbol binario de procesos, de tantos niveles como indique el argumento. En el ejemplo siguiente se muestra un árbol binario de tres niveles. (1 punto)



- b) Realizar un programa llamado **señales.c** que, partiendo de un árbol binario de tres niveles, como el que muestra la figura anterior, implemente la funcionalidad que se describe a continuación. Primero se construirá todo el árbol de procesos. Después, el proceso verde, transcurridos 10 segundos, le enviará una señal al proceso azul. Antes de enviar la señal, el proceso verde imprimirá “soy el proceso XXX y envío la señal YY”, donde XXX será su identificador de proceso e YY será el número de señal. Cuando el proceso azul recibe la señal imprimirá “soy el proceso XXX y he recibido la señal YY”, donde XXX será su identificador de proceso e YY será el número de señal. Además, el proceso azul debe ejecutar el comando “pstree”. (2 puntos)

## Ejercicio2. Comunicación entre procesos: tuberías. (3 puntos)

Realizar un programa llamado **hacha.c** que divida un archivo en varios trozos con el mismo nombre y extensión h00, h01, ... teniendo en cuenta el formato y las siguientes consideraciones:

```
$ hacha <archivo> <tamaño>  
archivo a dividir  
tamaño en bytes de los archivos divididos
```

- Para dividir, el proceso hacha (proceso padre) generará tantos procesos hijos como archivos se tengan que crear. Mediante tuberías el proceso padre enviará la información a los hijos y serán estos últimos los que creen los archivos de destino y escriban en ellos.

```
$ hacha at _madrid.mp3 50000
```

```
$ ls
```

[at\\_madrid.mp3.h00](#)

[at\\_madrid.mp3.h01](#)

[at\\_madrid.mp3.h02](#)

### Consideraciones

- Las entradas y salidas a los archivos se realizarán con llamadas al sistema estudiadas en la asignatura (no se puede utilizar printf, scanf, etc.)
- El padre creará tantos hijos como fragmentos del archivo a realizar, siendo decisión del alumno si los hijos se lanzan secuencial o concurrentemente.
- El padre lee el archivo origen y mediante tuberías le pasa la información del archivo a los hijos.
- Cada hijo crea el archivo de destino y escribe la información que le ha pasado el padre.

### Ejercicio3. Comunicación entre procesos: sockets. (4 puntos)

Implementar una aplicación cliente-servidor mediante la utilización de sockets cuya funcionalidad será la transferencia de texto entre el cliente y el servidor. El ejercicio constará de dos partes (procesos): el cliente, que leerá el texto introducido por el usuario desde la entrada estándar y lo enviará al servidor, y el servidor, que recibirá los datos del cliente y los mostrará por la salida estándar.

La finalización de la transferencia se producirá cuando el texto introducido por el usuario sea la palabra “FIN”.

El cliente se ejecutará en la máquina local y su función es establecer la comunicación con el servidor y enviar al servidor el texto introducido por el usuario. El servidor se ejecutará en la máquina remota y, después de establecer la comunicación, mostrará el texto recibido desde el cliente.

El cliente se lanzará mediante la siguiente instrucción:

#### **ClienteRemoto IP\_Servidor**

El servidor se lanzará con la orden **ServidorRemoto** y debe estar en todo momento escuchando por el puerto 9999.

La puntuación del ejercicio será la siguiente:

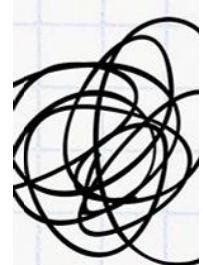
- Implementación correcta del cliente y su estructura. **(1 punto)**
- Implementación correcta del servidor y su estructura. **(1 punto)**
- La correcta implementación de la transferencia de texto. **(2 puntos)**

Importante

Puedo eliminar la publi de este documento con 1 coin

¿Cómo consigo coins? → Plan Turbo: barato  
→ Planes pro: más coins

pierdo  
espacio



Necesito  
concentración

ali ali ooooh  
esto con 1 coin me  
lo quito yo...

wuolah

Fecha de entrega: Hasta el 9 de julio.

Para la corrección de la práctica se debe entregar un informe (en formato PDF) donde se expliquen los pasos principales que se han realizado. Se deben incluir capturas de pantalla, que muestren la ejecución correcta de los distintos ejercicios. Además, se deben entregar los códigos fuente de los distintos ejercicios.

La entrega se realizará a través de la sección **Tutorías de UACloud**, enviando la tutoría al profesor de prácticas correspondiente. Se debe adjuntar a la tutoría un archivo comprimido con el **informe de la práctica** y los **códigos fuente**.