

Makefile (0,5 puntos)

Crea un **Makefile** que permita generar todos los programas del enunciado a la vez y cada uno de ellos por separado. Añade una regla (clean) para borrar todos los binarios y/o ficheros objeto y dejar solo los ficheros fuentes. Los programas deben generarse si, y solo si, ha habido cambios en los ficheros fuentes.

Control de Errores y Usage (0,5 puntos)

Todos los programas deben incluir un control adecuado de los argumentos usando la función `Usage()` y deben controlar los errores en todas las llamadas al sistema.

Ejercicio 1 (2,5 puntos)

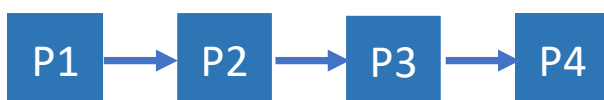
Implementa un código, en un fichero con nombre **ej1.c** que reciba como parámetro dos números enteros. El programa comenzará mostrando el valor del primer parámetro, a partir del cual a cada segundo lo irá incrementando en 1 y mostrando por pantalla. En caso de recibir un `SIGUSR1`, mostrará el mensaje: "RESTAN POR ACABAR XXX SEGUNDOS", siendo XXX el número de incrementos que faltan para mostrar tantos números como indica el segundo parámetro y continuar la ejecución.

El programa acaba cuando suceda primero alguna de las dos situaciones: escribe un total de números igual al valor del segundo parámetro o recibe una señal `SIGINT`. En cualquiera de los dos casos el programa debe mostrar el total de números mostrados por pantalla y utilizar este número como código de exit.

Los signals que tengas que reprogramar deben utilizar la misma única función, aunque distinguiendo en cada caso el signal causante del tratamiento.

Ejercicio 2 (2,0 puntos)

Implementa un código, en un fichero con nombre **ej2.c**, que sea capaz de crear la jerarquía de procesos que se muestra en el dibujo siguiente:



Cada proceso, una vez creado debe mostrar por pantalla su pid y la posición que ocupa en el orden de creación.

En el dibujo hay 4 procesos. Modifica el programa para que el número de procesos de la jerarquía sea dado por un parámetro en el programa.

Ejercicio 3 (3,5 puntos)

Implementa un código en un fichero con nombre **ej3.c** que cree en forma concurrente procesos. El número de procesos a crear se recibe como primer parámetro. Cada proceso hijo mutará para ejecutar el código **ej1** que has implementado en el *Ejercicio 1*. Si no lo has podido implementar correctamente, puedes utilizar el ejecutable adjunto en este control. Como primer parámetro de entrada utiliza el iterador del bucle de creación de procesos hijos y como segundo parámetro utiliza el número total de procesos.

Si transcurre un número de segundos superior al valor dado como segundo parámetro, el padre enviará un SIGKILL únicamente a aquellos procesos hijo que queden vivos. El tratamiento que tiene que hacer el padre para cada hijo que muere es mostrar en pantalla un mensaje indicando si ha finalizado normalmente, o si ha muerto por *signal* y entonces qué *signal* ha provocado su muerte.

Ejercicio 4 (1,0 puntos)

Responde, en el fichero **respuestas.txt**, razonando brevemente tus respuestas a las siguientes preguntas:

- A) ¿Como puedes comprobar el estado de un proceso en ejecución?
- B) Indica qué líneas de comandos has de introducir para que al poner como línea de comandos los caracteres **ll** muestre por pantalla el listado de ficheros en formato largo, y ordenado por fecha de creación de más antiguo a más reciente.

Qué se tiene que hacer

- El Makefile
- Los códigos de los programas en C
- La función Usage() para cada programa
- El fichero respuestas.txt con las respuestas a las preguntas

Qué se valora

- Que sigas las especificaciones del enunciado
- Que el uso de las llamadas a sistema sea el correcto
- Que se comprueben los errores de **todas** las llamadas al sistema
- Código claro y correctamente indentado
- Que el Makefile tenga bien definidas las dependencias y los objetivos
- La función Usage() que muestre en pantalla cómo debe invocarse correctamente al programa en caso que los argumentos recibidos no sean adecuados.
- El fichero respuestas.txt

Qué hay que entregar

Un único fichero tar.gz con el código fuente de los programas en C, el Makefile, y las respuestas en respuestas.txt:

CLAB1 grupos 11, 15 y 25

S01819Q1

```
tar zcvf clab1.tar.gz Makefile respuestas.txt ej1.c ej2.c ej3.c
```