Tema 1:
Desarrolle un programa que tome como entrada dos argumentos:
arg1: número de procesos a lanzar en paralelo (debe representar un arbol binario completo: 1,3,7,15,etc)
arg2: array de enteros separados por “,” (ejemplo: 61,5,72,8,6,4,82,7,1,2,4,7,8,4,4)
y realice el algoritmo de ordenacion mergesort tanto para la ordenación interna como para la ordenacion de los elementos resultantes de los hijos utilizando tantos procesos como indica el arg1, siguiendo un mapeo de tipo arbol binario completo (lógicamente la ejecucion en cada nivel del arbol debe darse en paralelo). Debe utilizar las funciones de manejo de procesos (fork, wait, waitpid, etc).
Como salida, el programa debe imprimir
1) la sublista de entrada de cada subproceso
2) las dos sublistas ordenadas de cada subproceso no hoja
2) la lista ordenada final

Ejemplo de mapeo de la siguiente lista de elementos en 7 procesos:
5,4,8,9,3,1,4,7,8,9,5,4,8,7,9,6

proceso 0
5,4,8,9,3,1,4,7,8,9,5,4,8,7,9,6

proceso 1
5,4,8,9,3,1,4,7

proceso 2
8,9,5,4,8,7,9,6

proceso 3
5,4,8,9

proceso 4
3,1,4,7

proceso 5
8,9,5,4

proceso 6
8,7,9,6

salida:
proceso 0: 5,4,8,9,3,1,4,7,8,9,5,4,8,7,9,6
proceso 1: 5,4,8,9,3,1,4,7
proceso 2: 8,9,5,4,8,7,9,6
proceso 3: 5,4,8,9
proceso 4: 3,1,4,7
proceso 5: 8,9,5,4
proceso 6: 8,7,9,6
proceso 1: lista izquierda {4,5,8,9}, lista derecha {1,3,4,7}
proceso 2: lista izquierda {4,5,8,9}, lista derecha {6,7,8,9}
proceso 0: lista izquierda {1,3,4,4,5,7,8,9}, lista derecha {4,5,6,7,8,8,9,9}
lista ordenada: 1,3,4,4,4,5,5,6,7,7,8,8,8,9,9,9

Tema 2:
Escriba un programa que setee un cronómetro de X segundos recibido como primer argumento. Entonces cada vez que se cumpla los X segundos un handler asociado a dicho cronómetro debe aumentar en 1 un contador global. Cuando este contador llegue a un límite máximo también recibido como argumento (segundo argumento) el handler debe avisar que se llegó al límite máximo y preguntar al usuario si se quiere terminar el programa o empezar de nuevo el cronómetro. En caso de presentarse una señal SIGINT, el programa deberá imprimir el valor del contador y finalizar.

Tema 3:
Escriba un programa que lance X procesos (donde X se recibe como primer argumento). Los procesos hijos deberán ignorar la señal SIGINT (el padre no debe ignorar, por tanto los hijos quedaran huerfanos como hijos del proceso init en caso que el padre reciba SIGINT). El Proceso padre deberá proporcionar la capacidad de manejar tres comandos (quedará bloqueado en espera de comandos):
1) lista: imprime la lista de los pid de los procesos hijos
2) exit: envía la señal kill a cada proceso hijo y termina
3) kill <pid>: envía la señal kill al proceso hijo con id <pid>