

Lenguajes de Programación III

Trabajo Practico 1.

Prof.: Ing. Gustavo Sosa Cataldo

Planteamiento del T. P.:

Este trabajo involucra el desarrollo de dos implementaciones utilizando conceptos vistos en clase y laboratorio.

Cada tema vale 50% del primer parcial, por tanto en total corresponde al 100%.

Máxima cantidad de integrantes por grupo: 2 (se puede hacer individual)

Todos los integrantes del grupo deben estar presentes el día de la entrega/defensa, el que no esté presente lleva 0.

La fecha de entrega/defensa será en el día del primer parcial es decir el lunes 22 de marzo mediante meet. Les enviaré unos días antes el horario asignado a cada grupo. Además se debe compartir el código fuente con el profesor mediante GITHUB o GITLAB (que será revisado si se considera oportuno).

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) el mapeo de cada sublista de entrada de cada subprocesso

2) las dos sublistas ordenadas que recibe de sus hijos cada subprocesso no hoja y su lista final ordenada

3) el esquema gráfico en arbol

Ejemplo de mapeo de la siguiente lista de elementos en 7 procesos:

- Ejecucion del proceso:

./proceso 7 5,4,8,9,3,1,4,7,8,9,5,4,8,7,9,6

- Salida en consola:

===esquema de arbol===

```

                    proceso 0
                5,4,8,9,3,1,4,7,8,9,5,4,8,7,9,6
            proceso 1      proceso 2
        5,4,8,9,3,1,4,7  8,9,5,4,8,7,9,6
proceso 3      proceso 4      proceso 5      proceso 6
5,4,8,9      3,1,4,7      8,9,5,4
    8,7,9,6
```

===mapeos===

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

===procesamiento===

proceso 3 lista ordenada: {4,5,8,9}

proceso 4 lista ordenada: {1,3,4,7}

proceso 1: lista izquierda {4,5,8,9}, lista derecha {1,3,4,7} => {1,3,4,4,5,7,8,9}

proceso 5 lista ordenada: {4,5,8,9}

proceso 6 lista ordenada: {6,7,8,9}

proceso 2: lista izquierda {4,5,8,9}, lista derecha {6,7,8,9} => {4,5,6,7,8,8,9,9}

proceso 0: lista izquierda {1,3,4,4,5,7,8,9}, lista derecha {4,5,6,7,8,8,9,9} =>

{1,3,4,4,4,5,5,6,7,7,8,8,8,9,9,9}

Tema 2

Desarrolle dos programas

Programa 1: un proceso que envíe de forma planificada señales a los procesos detallados en un archivo de texto. En cada fila de este archivo se indicará:

ProcesoId	Señal	SegundosDelay
-----------	-------	---------------

Por ejemplo:

12345	9	22
2222	15	4
3333	3	11

En este caso, el programa, una vez que arranca su ejecución enviaría la señal 9 al proceso 12345 a los 22 segundos de haber arrancado, la señal 15 al proceso 2222 a los 4 segundos de arrancar y así sucesivamente. El programa termina cuando no queda ninguna planificación pendiente a enviar.

Programa 2: un proceso que implemente un handler que notifique cuando reciba las principales señales (SIGINT, SIGALRM, SIGUSR1, SIGUSR2, etc). Este proceso también será utilizado para validar el programa 1.