Control Laboratorio 2 SO1920q1

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 sólo los ficheros fuente. Los programas deben generarse si, y solo si, ha habido cambios en los ficheros fuente.

Control de errores (0.5 puntos)

Para todos los programas que se piden a continuación deben comprobarse los errores de TODAS las llamadas a sistema (excepto el `write` por pantalla), controlar los argumentos de entrada y definir la función Usage().

test.c (3.0 puntos)

Implementa un programa ***test.c*** que reciba como parámetro dos números *X* y *N*, enteros mayores que cero (siendo N ≤ X ≤ 9999, ej: N=10 X=1000). A continuación, debe generar *X* números enteros aleatorios cuyos valores se encuentren dentro del rango *0* a *N-1*. Para cada número *i* generado, el programa debe incrementar en 1 el valor en la posición *i-ésima* de un fichero llamado "ocurrencias.bin". Este fichero debe inicializarse previamente a cero y debe contabilizar las ocurrencias de cada número aleatorio generado en formato máquina. El programa debe mantener las ocurrencias directamente en el fichero (NO puede utilizar un vector para contabilizarlas).

Nota: Para la generación de cada número *i* en el rango indicado, puedes utilizar la función *`rand()`* combinada con el operador de módulo: `int i = rand()%N;`

directorios.c (3.0 puntos)

Escribe un programa **directorios.c** que liste los directorios dentro del directorio home ordenados de forma descendente según el tamaño que ocupan en el sistema de ficheros. Para hacerlo, el programa debe crear un proceso hijo y luego mutar al programa "du", mientras que el proceso hijo debe mutar al comando "sort". La comunicación y entre padre e hijo debe hacerse a través de una pipe sin nombre, emulando al siguiente comando:

```
du -b path_a_home | sort -nr
```

Donde `path_a_home` es el path absoluto del directorio home, *`du -b`* escribe el tamaño de los directorios (ver *man du*) y *`sort -nr`* ordena en forma descendente las líneas que recibe como entrada (ver *`man sort`*).

directorios2.c (2.0 puntos)

Crea una copia de *directorios.c* en ***directorios2.c***. Modifica *directorios2.c* para resolver el mismo problema pero ahora utilizando una pipe con nombre, llamada "PIPE_DIR". La pipe ha de ser creada desde código. Si "PIPE_DIR" ya existe, el programa debe continuar ya que no se trata de ningún error.

respuestas.txt (1.0 punto)

Indica en ***respuestas.txt*** la secuencia de instrucciones de código necesarias para reservar en memoria dinámica (utilizando llamadas a sistema) un vector de N enteros e inicializar cada posición a 0 .

Control Laboratorio 2 SO1920q1

Qué hay que hacer

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

Qué se valora

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

Qué hay que entregar

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

```
tar zcvf clab2.tar.gz Makefile respuestas.txt *.c
```