Tareas

Prácticas de Ampliación de Sistemas Operativos - Tarea Semana 4 - Corregida

Borrador - En proceso
Enviada
Corregida

Título	Prácticas de Ampliación de Sistemas Operativos - Tarea Semana 4
Estudiante	ELENA PEREZ GONZALEZ-TABLAS
Fecha de envío	08-nov-2020 22:16
Calificación	**9,50 (máx 10,00)**

Instrucciones

Descripción del programa `cat_pipe_tr_pipe_wc`

Escribe un programa llamado `cat_pipe_tr_pipe_wc` que acepte las siguientes opciones `./cat_pipe_tr_pipe_wc [-e] -s SRC -d DST [FILEIN1 FILEIN2 ... FILEINn]`. Por ejemplo:

```
$ ./cat_pipe_tr_pipe_wc -s rq -d RQ entrada1 entrada2
```

- El programa deberá simular la ejecución de la siguiente línea de órdenes, pero sin ejecutar las utilidades del sistema operativo, es decir, sin usar las llamadas al sistema `exec*()`.

  ```
  $ cat FILEIN1 FILEIN2 ... FILEINn | tr -s SRC -d DST | wc -lc
  ```

  En su lugar se implementarán tres funciones, una por cada una de las aplicaciones, llamadas respectivamente `catfd`, `trfd` y `wcfd`, que recibirán los parámetros necesarios para su correcto funcionamiento (las dos primeras funciones se han implementado en una tarea anterior). Los parámetros `FILEINi` son los ficheros de entrada. Si no se proporciona ninguno, se usará la entrada estándar.

- El programa deberá crear tres procesos hijos, cada uno de los cuales ejecutará una de las funciones. Estos procesos estarán conectados entre sí por tuberías, de manera que cada uno pueda proporcionar al siguiente los datos que genere. El proceso padre esperará la terminación de los hijos sin un orden determinado.
- El primer hijo se comportará como una orden `cat` de los ficheros de entrada, usando la entrada estándar si no se proporciona ningún fichero.
- El segundo hijo se comportará como una orden `tr` que tomará lo que le proporcione el anterior proceso y traducirá cada carácter de la cadena `SRC` por el carácter en la misma posición de la cadena `DST`. En consecuencia, ambas cadenas deben ser **no nulas y tener la misma longitud**, que tiene que ser mayor o igual que 1. En caso contrario, se mostrará un mensaje de error y se saldrá proporcionando un valor de terminación adecuado.
- El parámetro `-e` indicará que los caracteres que componen las cadenas `SRC` y `DST` de la función `trfd` admiten caracteres de control. Esto se verá mejor con un ejemplo. La primera de las siguientes ordenes sustituirá cada letra `A` por un retorno de carro, mientras que la segunda, sustituirá cada letra `A` por una barra invertida y cada letra `B` por una letra `n`. La diferencia está en que el parámetro `-e` hará que tomemos `\n` como un único carácter en lugar de dos.

  - `$ cat_pipe_tr_pipe_wc -e -s 'A' -d '\n' file1 file2`

    - `$ cat_pipe_tr_pipe_wc -s 'AB' -d '\n' file1 file2`

      Por simplicidad, el único carácter de control que se debe reconocer es `'\n'`.

- El tercer y último hijo se comportará como una orden `wc -lc` que saca como salida el número de líneas y de caracteres que tiene el fichero que se le proporciona por la entrada estándar. Esta función tendremos que implementarla nosotros y no usar la que proporciona el sistema operativo.
- Recordemos que queremos simular el comportamiento de las órdenes y no ejecutarlas. Por tanto, al no ser aplicaciones externas que hay que ejecutar sino funciones, la redirección de las entradas y salidas estándar es opcional, ya que existe una solución más simple utilizando directamente las tuberías.
- Cada uno de los procesos deberá definir un buffer adecuado para su funcionamiento, cuyo tamaño se determinará gracias a la gráfica de rendimiento que obtuvimos en la anterior tarea para conseguir un rendimiento adecuado sin un excesivo consumo de memoria.
- Las lecturas y escrituras parciales deben tratarse correctamente, tal y como se explicó en las sesiones de prácticas.

Ejemplos de ejecución de `cat_pipe_tr_pipe_wc`

```
$ ./cat_pipe_tr_pipe_wc

Uso: ./cat_pipe_tr_pipe_wc [-e] -s SRC -d DST [FILEIN1 FILEIN2 ... FILEINn]

$ ./cat_pipe_tr_pipe_wc -s '' -d ''

Error: SRC y DST deben ser cadenas de caracteres de la misma longitud

$ ./cat_pipe_tr_pipe_wc -s -d

Uso: ./cat_pipe_tr_pipe_wc [-e] -s SRC -d DST [FILEIN1 FILEIN2 ... FILEINn]

$ ./cat_pipe_tr_pipe_wc -s ab -d ABC

Error: SRC y DST deben ser cadenas de caracteres de la misma longitud

$ echo 'hola mundo' | ./cat_pipe_tr_pipe_wc -s 'ao' -d 'AO'

1 11

$ echo 'hola mundo' | ./cat_pipe_tr_pipe_wc -e -s 'ao' -d '\n\n'

4 11

$ echo qwertyuiop > entrada1 ; echo qawsedrftg > entrada2 ; ./cat_pipe_tr_pipe_wc -
e -s 'rq' -d '\nQ' entrada1 entrada2

4 22

$ base64 /dev/urandom | head -c 1000000 | ./cat_pipe_tr_pipe_wc -e -s '\n' -d '-'

0 1000000
```

Entrega

La entrega consistirá en un único fichero `cat_pipe_tr_pipe_wc.c`. El fichero `cat_pipe_tr_pipe_wc.c` debe compilar conforme a los ficheros `tasks.json` o `Makefile` incluidos en los ficheros `.tgz`, es decir, con `gcc -ggdb3 -Wall -Werror -Wno-unused -std=c11 cat_pipe_tr_pipe_wc.c -o cat_pipe_tr_pipe_wc`. La primera línea de `cat_pipe_tr_pipe_wc.c` debe ser la directiva para compilar conforme al estándar POSIX: `#define _POSIX_C_SOURCE 200809L`.

Recursos adicionales para la tarea

No hay adjuntos todavía

Adjunto enviado

- cat_pipe_tr_pipe_wc.c (10 KB; 08-nov-2020 22:16)

Comentarios adicionales del profesor

- La implementación de '-e' es ineficiente. El buffer auxiliar es innecesario.

Volver a la lista