

Tareas

Prácticas de Ampliación de Sistemas Operativos - Tarea Semana 3 - Corregida

Borrador - En proceso

Enviada

Corregida

Título	Prácticas de Ampliación de Sistemas Operativos - Tarea Semana 3
Estudiante	ELENA PEREZ GONZALEZ-TABLAS
Fecha de envío	18-oct-2020 21:29
Calificación	5,00 (máx 10,00)

Instrucciones

Descripción del programa tr_mem_din

Escribe un programa llamado tr_mem_din que acepte las siguientes opciones ./tr_mem_din -s SRC -d DST [-t SIZE] [-o FILEOUT] [FILEIN1 FILEIN2 ... FILEINn]. Por ejemplo:

```
$ ./tr_mem_din -s rq -d RQ -t 4096 -o salida entrada1 entrada2
```

- El programa debe sustituir cada carácter de la cadena de caracteres SRC por el carácter en la misma posición de la cadena de caracteres 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, tr_mem_din emitirá un mensaje de error.
- El parámetro -t SIZE especifica el tamaño del buffer de lectura/escritura. Por defecto, el tamaño es de 1024 bytes. El buffer de lectura/escritura debe reservarse en memoria dinámica. El valor de SIZE no puede ser menor que 1 byte, ni mayor que 128 MB (134.217.728 bytes).
- El parámetro -o FILEOUT indica el fichero donde se escribirá la versión modificada. Si no se proporciona un fichero de salida, se usará la salida estándar.
- Los parámetros FILEIN son los ficheros de entrada a modificar. Si no se proporciona ninguna, se usará la entrada estándar.
- Las escrituras parciales deben tratarse correctamente, tal y como se explicó en la sesión de prácticas.

Ejemplos de ejecución de tr_mem_din

```
$ ./tr_mem_din
```

Uso: ./tr_mem_din -s SRC -d DST [-t SIZE] [-o FILEOUT] [FILEIN1 FILEIN2 ... FILEINn]

```
$ ./tr_mem_din -s "" -d ""
```

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

```
$ ./tr_mem_din -s "ab" -d "ABC"
```

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

```
$ ./tr_mem_din -s "ab" -d "AB" -t 1000000000
```

Error: SIZE debe ser un valor entre 1 y 134217728

```
$ ./tr_mem_din -s abcd -d ABCD -t 4096
```

caso

CAso

(Ctrl+D)

```
$ echo qwertyuiop > entrada1 ; echo qawsedrftg > entrada2
```

```
$ ./tr_mem_din -t 4096 -s rq -d RQ -o salida entrada1 entrada2 ; cat salida
```

QweRtyuiop

QawsedRftg

```
$ base64 /dev/urandom | head -c 1000000 | ./tr_mem_din -t 4096 -s a -d A | grep a
```

(no hay salida)

Entrega

La entrega consistirá en un único fichero **tr_mem_din.tgz** que a su vez contendrá dos ficheros: `tr_mem_din.c` y [tr_mem_din.ods](#).

El fichero `tr_mem_din.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 tr_mem_din.c -o tr_mem_din`. La primera línea de `tr_mem_din.c` debe ser la directiva para compilar conforme al estándar POSIX: `#define _POSIX_C_SOURCE 200809L`.

El fichero [tr_mem_din.ods](#) es una plantilla para medir el rendimiento de `tr_mem_din` que debe cumplimentarse. La orden a ejecutar para obtener los tiempos sería la siguiente:

```
$ base64 /dev/urandom | head -c 1000000000 > entrada ; $(which time) -f "%e segundos" ./tr_mem_din -t T -s a -d A -o /dev/null entrada ; rm entrada
```

(no hay salida)

Recursos adicionales para la tarea

No hay adjuntos todavía

Adjunto enviado

-  tr_mem_din.tgz (18 KB; 18-oct-2020 21:29)

Comentarios adicionales del profesor

- El programa se cuelga con el test:

```
base64 /dev/urandom | head -c 1000000 > random ; tr -d -c 'a' < random | wc -c
```

```
./tr_mem_din -t 4096 -s aA -d Aa random > random2 ; tr -d -c 'A' < random2 | wc -c
```

- La salida del programa NO es correcta con -s aA -d Aa porque la implementación de tr es incorrecta.
- La implementación de tr se basa en buscar '\0' en lugar de utilizar el número de bytes leídos. Utilizar el carácter '\0' no tiene sentido pues podría no aparecer nunca. El programa podría fallar generando una violación de segmento.
- Se puede implementar tr correctamente con O(n).

[Volver a la lista](#)