

Prácticas de Ampliación de Sistemas Operativos •

Tarea Semana 3

Descripción del programa `merge_files`

Escribe un programa llamado `merge_files` que acepte las siguientes opciones `./merge_files [-t BUFSIZE] [-o FILEOUT] FILEIN1 [FILEIN2 ... FILEINn]`. Por ejemplo:

```
$ ./merge_files -t 16 -o salida entrada1 entrada2
```

- El programa debe barajar los bytes de los ficheros que se proporcionan como entrada. De esta manera, en el ejemplo anterior, la salida tendrá el byte 0 del primer fichero, seguido del byte 0 del segundo fichero, seguido del byte 1 del primer fichero, seguido del byte 1 del segundo fichero, y así sucesivamente. Es decir, si `entrada1` contiene la cadena `ABCD\n` y `entrada2`, la cadena `123456\n`, la salida debería ser: `A1B2C3D4\n56\n`. El programa se comportará de igual manera con cualquier número de ficheros de entrada. Conforme cada fichero se vaya agotando, se dejará de usar, intercalando los bytes de los ficheros restantes de la misma forma.
- El parámetro `-t BUFSIZE` especifica el tamaño del buffer de lectura/escritura. Por defecto, el tamaño será de 1024 bytes. El buffer de lectura/escritura debe reservarse en memoria dinámica. El valor de `BUFSIZE` 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. Si no se proporciona ninguno, se mostrará un error. El programa puede admitir hasta un máximo de 16 ficheros de entrada.
- Las lecturas y escrituras parciales deben tratarse correctamente, tal y como se explicó en la sesión de prácticas.

Ejemplos de ejecución de `merge_files`

```
$ ./merge_files
Error: No hay ficheros de entrada.
Uso: ./merge_files [-t BUFSIZE] [-o FILEOUT] FILEIN1 [FILEIN2 ... FILEINn]
No admite lectura de la entrada estandar.
-t BUFSIZE  Tamaño de buffer donde 1 <= BUFSIZE <= 128MB
-o FILEOUT   Usa FILEOUT en lugar de la salida estandar

$ ./merge_files -t f1 f2
Error: Tamaño de buffer incorrecto.
Uso: ./merge_files [-t BUFSIZE] [-o FILEOUT] FILEIN1 [FILEIN2 ... FILEINn]
No admite lectura de la entrada estandar.
-t BUFSIZE  Tamaño de buffer donde 1 <= BUFSIZE <= 128MB
```

```

-o FILEOUT Usa FILEOUT en lugar de la salida estandar

$ ./merge_files f1 f2 ff3 f4 f5 f6 f7 f8 f9 f10 f11 f12 f13 f14 f15 16 f17
Error: Demasiados ficheros de entrada. Máximo 16 ficheros.
No admite lectura de la entrada estandar.
-t BUFSIZE Tamaño de buffer donde 1 <= BUFSIZE <= 128MB
-o FILEOUT Usa FILEOUT en lugar de la salida estandar

$ echo -e "ABCDEFGHIIJK" > f1; echo -e "123456" > f2; echo -e
"abcdefghijkl\nmnopq" > f3
$ ./merge_files -t 1 f1 f2
A1B2C3D4E5F6G
HIJK

$ ./merge_files -t 16 f1 f2 f3
A1aB2bC3cD4dE5eF6fG
gHhIiJjKk
l
mnopq

$ ./merge_files -o salida -t 16 f1 f2 nofile f3
Error: No se puede abrir 'nofile': No such file or directory
$ cat salida
A1aB2bC3cD4dE5eF6fG
gHhIiJjKk
l
mnopq

$ ls -l merge_files
-rwxrwxr-x 1 alumno alumno 69016 sep 17 16:20 ./merge_files
$ ./merge_files -o salida merge_files merge_files
$ ls -l salida
-rw----- 1 alumno alumno 138032 sep 17 16:30 salida
$ hexdump -C merge_files | head -4
00000000  7f 45 4c 46 02 01 01 00  00 00 00 00 00 00 00 00
|.ELF.....|
00000010  03 00 3e 00 01 00 00 00  20 12 00 00 00 00 00 00  |..>.....
.....|
$ hexdump -C salida | head -4
00000000  7f 7f 45 45 4c 4c 46 46  02 02 01 01 01 01 00 00
|..EELLFF.....|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
|.....|
00000020  03 03 00 00 3e 3e 00 00  01 01 00 00 00 00 00 00  |....>>.....|
00000030  20 20 12 12 00 00 00 00  00 00 00 00 00 00 00 00  |
.....|

```

Entrega

La entrega consistirá en un único archivo `merge_files.c`. El archivo debe compilar conforme a los archivos `tasks.json` o `Makefile` incluidos en `ASO-Semana1.tar.xz`, es decir, con `gcc -ggdb3 -Wall`

`-Werror -Wno-unused -std=c11 merge_files.c -o merge_files`. La primera línea de `merge_files.c` debe ser la directiva para compilar conforme al estándar POSIX: `#define _POSIX_C_SOURCE 200809L`. `merge_files.c` deberá pasar como mínimo los test indicados anteriormente. Por supuesto, tiene que funcionar tanto con archivos de texto como binarios, y con todos los tamaños de buffer entre 1 byte y 128 MBytes.

Evaluación

Para aprobar la práctica es imprescindible que los programas pasen las pruebas de ejecución que se muestran en este documento, sin que esto signifique que no se puedan detectar otros errores con pruebas adicionales que el profesor haga durante la evaluación.

Se tendrá en cuenta la correcta indentación del código, su estructuración, el manejo de los posibles errores de las llamadas al sistema o funciones de biblioteca, así como la correcta gestión de la memoria dinámica. Otro punto importante, y por eso lo volvemos a recordar, es que se penalizará severamente el no usar buffers para las lecturas/escrituras, es decir, leer o escribir byte a byte, pudiendo llegar a suspender la práctica, así como no contemplar lecturas o escrituras parciales si así se pide en el enunciado. Adicionalmente, las ineficiencias que se detecten en el código, también pueden ser objeto de penalizaciones.