

Piscina C C 10

Staff 42 pedago@42.fr

Resumen: Este documento corresponde a la evaluación del módulo C 10 de la piscina de 42.

Índice general

I.	Instrucciones	2
II.	Preámbulo	4
III.	Exercice 00 : display_file	5
IV.	Exercice 01 : cat	6
V.	Exercice 02 : tail	7
VI.	Exercice 03: hexdump	8

Capítulo I

Instrucciones

- Esta página será la única referencia: no se fíe de los rumores de pasillo.
- Vuelva a leer bien los enunciados antes de entregar sus ejercicios. Los enunciados pueden cambiar en cualquier momento.
- Tenga cuidado con los permisos de sus archivos y de sus directorios.
- Debe respetar el procedimiento de entrega para todos sus ejercicios.
- Sus compañeros de piscina se encargarán de corregir sus ejercicios.
- Además de por sus compañeros, también será corregido por un programa que se llama la Moulinette.
- La Moulinette es muy estricta a la hora de dar notas. Está completamente automatizada. Es imposible discutir con ella sobre su nota. Por lo tanto, sea extremadamente riguroso para evitar cualquier sorpresa.
- La Moulinette no tiene una mente muy abierta. No intenta comprender el código que no respeta la Norma. La Moulinette utiliza el programa norminette para comprobar la norma de sus archivos. Entienda entonces que es estúpido entregar un código que no pase la norminette.
- Los ejercicios han sido ordenados con mucha precisión del más sencillo al más complejo. En ningún caso le prestaremos atención ni tendremos en cuenta un ejercicio complejo si no se ha conseguido realizar perfectamente un ejercicio más sencillo.
- El uso de una función prohibida se considera una trampa. Cualquier trampa será sancionada con la nota -42.
- Solamente tendrá que entregar una función main() si le pedimos un programa.
- La Moulinette compila con los flags -Wall -Wextra -Werror y utiliza gcc.
- Si su programa no compila, tendrá 0.
- <u>No debe</u> dejar en su directorio <u>ningún</u> archivo que no se haya indicado de forma explícita en los enunciados de los ejercicios.

- ¿Tiene alguna pregunta? Pregunte a su vecino de la derecha. Si no, pruebe con su vecino de la izquierda.
- Su manual de referencia se llama Google / man / Internet /
- ¡No olvide participar en el foro Piscina de su Intranet o en el slack de su Piscina!
- Lea detenidamente los ejemplos. Podrían exigir cosas que no se especifican necesariamente en los enunciados...
- Razone. ¡Se lo suplico, por Odín! Maldita sea.

Capítulo II

Preámbulo

Para bien empezar el día, he aquí algunas preguntas muy sencillas:

¿Qué pasaría si metiésemos un secador alimentado por corriente continua en una caja hermética de un metro de lado?

¿Se habría detenido el accidente de Chernóbil arrojando antimateria en el reactor cuando se estaba fundiendo?

¿Es posible llorar hasta deshidratarse?

Si todos los humanos desaparecieran de la faz de la tierra, ¿cuánto tiempo pasaría antes de que la última fuente de luz artificial se apagara?

¿Hasta qué punto es peligroso meterse en una piscina durante una tormenta?

¿Desde qué altura habría que tirar un filete para que estuviera cocinado cuando llegara al suelo?

¿Cuándo -en caso de que ocurra alguna vez- el ancho de banda de Internet superará al servicio de mensajería de FedEx?

¿Cuántos tweets diferentes son posibles en nuestro idioma? ¿Cuánto tiempo tardaría la población mundial en leerlos todos en voz alta?

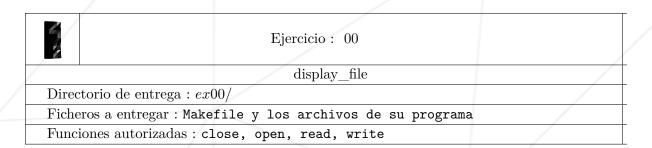
¿Cuál sería el resultado si todos los candidatos al carnet de conducir respondiesen al azar en el test del examen teórico?

¿Cuántos contestarían correctamente a todas las preguntas?

Preguntas sacadas les libro '¿Qué pasaría si... ?' de Randall Munroe.

Capítulo III

Exercice 00: display_file



- Escriba un <u>programa</u> que se llame ft_display_file y que muestre en la salida estándar únicamente el contenido del archivo pasado como argumento.
- El directorio de entrega tendrá un Makefile con una regla all, una regla clean y una regla fclean. El binario se llamará ft_display_file.
- La función malloc está prohibida. Puede hacer el ejercicio declarando únicamente una tabla de tamaño fijo.
- Todos los archivos pasados como parámetros serán válidos.
- Los mensajes de error tendrán que mostrarse en la salida que les haya sido reservada y seguidos de un salto de línea.
- Si no hay ningún argumento, su programa deberá mostrar

File name missing.

Si hay demasiados argumentos, su programa deberá mostrar

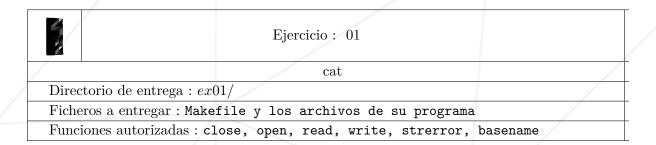
Too many arguments.

• Si el archivo no se puede leer, su programa deberá mostrar

Cannot read file.

Capítulo IV

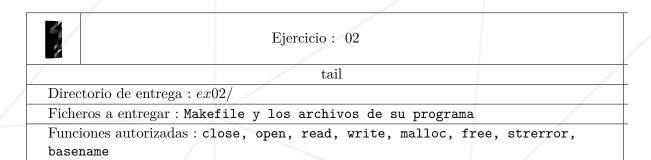
Exercice 01: cat



- Escriba un <u>programa</u> que se llame ft_cat y que realice lo mismo que el comando cat del sistema.
- No necesita gestionar las opciones.
- El directorio de entrega tendrá un Makefile con una regla all, una regla clean y una regla fclean.
- Puede utilizar la variable errno (ver el man de errno).
- Debería leer los man de todas las funciones autorizadas.
- Puede hacer el ejercicio declarando únicamente una tabla de tamaño fijo. Esta tabla tendrá un tamaño limitado a algo menos de unos 30 ko. Para probar esta limitación, utilice el comando ulimit de su shell.

Capítulo V

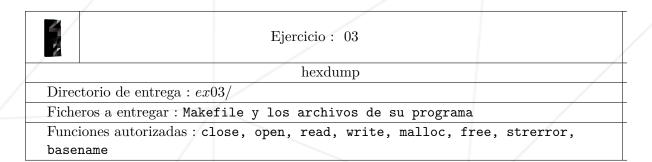
Exercice 02: tail



- Escriba un <u>programa</u> que se llame ft_tail y que realice lo mismo que el comando tail.
- \bullet Solo tiene que gestionar la opción $\neg c$, pero no tiene que gestionar el '+' y el '-'.
- Se realizarán todas las pruebas con la opción -c.
- El directorio de entrega tendrá un Makefile con una regla all, una regla clean y una regla fclean.
- Puede utilizar la variable errno.

Capítulo VI

Exercice 03: hexdump



- Escriba un <u>programa</u> que se llame ft_hexdump y que realice lo mismo que el comando hexdump del sistema, sin redirección.
- No necesita gestionar la opción -c.
- El directorio de entrega tendrá un Makefile con una regla all, una regla clean y una regla fclean.
- Puede utilizar la variable errno (ver el man de errno).