

Piscina C C 10

 $Sum{\'a}rio:\ Este\ documento\ \'e\ o\ tema\ do\ m\'odulo\ C\ 10\ da\ Piscina\ C\ da\ 42.$

Conteúdo

Ι	Instruções	6
II	Preâmbulo	4
III	Exercício 00 : display_file	ţ
IV	Exercício 01 : cat	(
V	Exercício 02 : tail	,
VI	Exercício 03: hexdump	

Capítulo I

Instruções

- Somente esta página servirá de referência, não confie nos boatos.
- Releia bem o tema antes de entregar seus exercícios. A qualquer momento o tema pode mudar.
- Atenção aos direitos de seus arquivos e suas pastas.
- Você deve seguir o procedimento de entrega para todos os seus exercícios.
- Os seus exercícios serão corrigidos por seus colegas de piscina.
- Além dos seus colegas, haverá a correção de um programa chamado Moulinette.
- A Moulinette é muito rigorosa na sua avaliação. Ela é completamente automatizada. É impossível discutir sua nota com ela. Tenha um rigor exemplar para evitar surpresas.
- A Moulinette não tem a mente muito aberta. Ela não tenta entender o código que não respeita a Norma. A Moulinette utiliza o programa norminette para verificar a norma dos seus arquivos. Entenda que é uma tolice entregar um código que não passe na norminette.
- Os exercícios estão rigorosamente ordenados do mais simples ao mais complexo. Em nenhum caso daremos atenção, nem levaremos em conta um exercício complexo se outro mais simples não tiver sido perfeitamente realizado.
- A utilização de uma função proibida é um caso de fraude. Qualquer fraude é punida com nota de -42.
- Você não deve entregar uma função main() se nós pedirmos um programa.
- A Moulinette compila com as sinalizações -Wall -Wextra -Werror, e utiliza gcc.
- Se o seu programa não compila, você terá 0.

- Você <u>não deve</u> deixar em sua pasta <u>nenhum</u> outro arquivo além daqueles explicitamente especificados pelos enunciados dos exercícios.
- Você tem alguma dúvida? Pergunte ao seu vizinho da direita. Ou tente também perguntar ao seu vizinho da esquerda.
- Seu manual de referência se chama Google / man / Internet /
- Considere discutir no fórum Piscina do seu Intra, assim como no slack da sua Piscina!
- Leia atentamente os exemplos. Eles podem muito bem pedir coisas que não estão especificadas no tema...
- Reflita. Por favor, por Odin! Por tudo que é mais sagrado.

Capítulo II

Preâmbulo

Para começar bem seu dia, veja a seguir algumas questões muito simples:

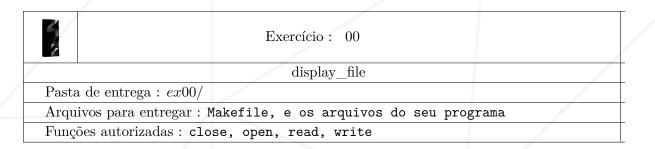
- O que aconteceria se eu deixasse um secador de cabelo ligado com eletricidade contínua em um cubo hermético de um metro de lado?
- Jogar antimatéria no reator de Chernobyl quando ele estava derretendo teria impedido sua fusão?
- É possível chorar até desidratar?
- Se todos os seres humanos desaparecessem da face da Terra, depois de quanto tempo a última fonte de luz artificial se apagaria?
- É realmente perigoso nadar em uma piscina durante uma tempestade?
- De que altura seria preciso soltar um bife para que ele esteja cozido ao chegar no chão?
- Quando se é que um dia isso vai acontecer a largura de banda da Internet vai superar a da FedEx?
- Quantos tweets diferentes são possíveis na nossa língua? E quanto tempo a população mundial levaria para ler todos em voz alta?
- Qual seria o resultado se todos os candidatos ao exame para tirar carteira de mot respondessem aleatoriamente ao questionário de múltipla escolha?

 Quantos acertariam todas as perguntas?

Questões extraídas do livro 'E se?' de Randall Munroe.

Capítulo III

Exercício 00: display_file



- Escreva um <u>programa</u> chamado ft_display_file que exiba na saída padrão somente o conteúdo do arquivo passado como argumento.
- A pasta de entrega deverá ter um Makefile com uma regra all, uma regra clean e uma regra fclean. O binário vai se chamar ft_display_file.
- A função malloc é proibida. Você pode fazer o exercício somente declarando uma matriz de tamanho fixo.
- Todos os arquivos passados como parâmetro serão válidos.
- As mensagens de erro deverão ser exibidas na saída que lhes está reservada e sendo seguidas por uma quebra de linha.
- Se não houver nenhum argumento, seu programa deverá mostrar

File name missing.

• Se houver muitos argumentos, seu programa devera mostrar

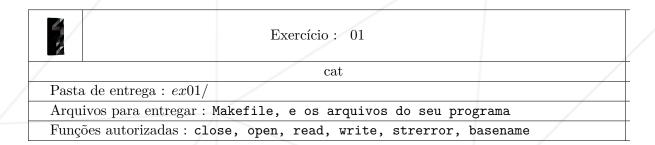
Too many arguments.

• Se o arquivo não for legivel, seu programa devera mostrar

Cannot read file.

Capítulo IV

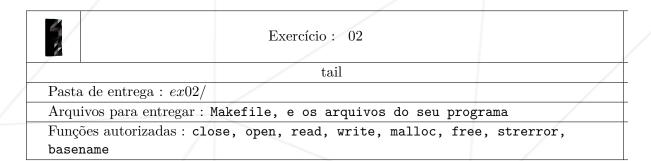
Exercício 01: cat



- Escreva um <u>programa</u> chamado ft_cat que execute o mesmo trabalho que o comando cat do sistema.
- Você não terá que lidar com as opções.
- A pasta de entrega deverá ter um Makefile com uma regra all, uma regra clean e uma regra fclean.
- Você pode utilizar a variável errno (ver o man de Errno).
- Você deve ler os man de todas as funções autorizadas
- Você pode fazer o exercício somente declarando uma matriz de tamanho fixo. Essa matriz terá um tamanho limitado a um pouco menos de aproximadamente 30 ko. Para testar essa limitação, utilize o comando ulimit no seu shell.

Capítulo V

Exercício 02: tail



- Escreva um <u>programa</u> chamado ft_tail que execute o mesmo trabalho que o comando tail.
- Você terá que lidar somente com a opção -c, e não o '+' e o '-'.
- Todos os testes serão realizados com a opção -c
- A pasta de entrega deverá ter um Makefile com uma regra all, uma regra clean e uma regra fclean.
- Você pode utilizar a variável errno

Capítulo VI

Exercício 03: hexdump

		Exercício: 03			
		hexdump			
	Pasta de entrega : $ex03/$				
Arquivos para entregar : Makefile, e os arquivos do seu programa					
	Funções autorizadas: close, open, read, write, malloc, free, strerror,				
	basename				

- Escreva um <u>programa</u> chamado ft_hexdump que execute o mesmo trabalho que o comando hexdump do sistema sem redirecionamento.
- $\bullet\,$ Você terá que lidar somente com a opção -C.
- A pasta de entrega deverá ter um Makefile com uma regra all, uma regra clean e uma regra fclean.
- Você pode utilizar a variável errno (ver o man de errno).