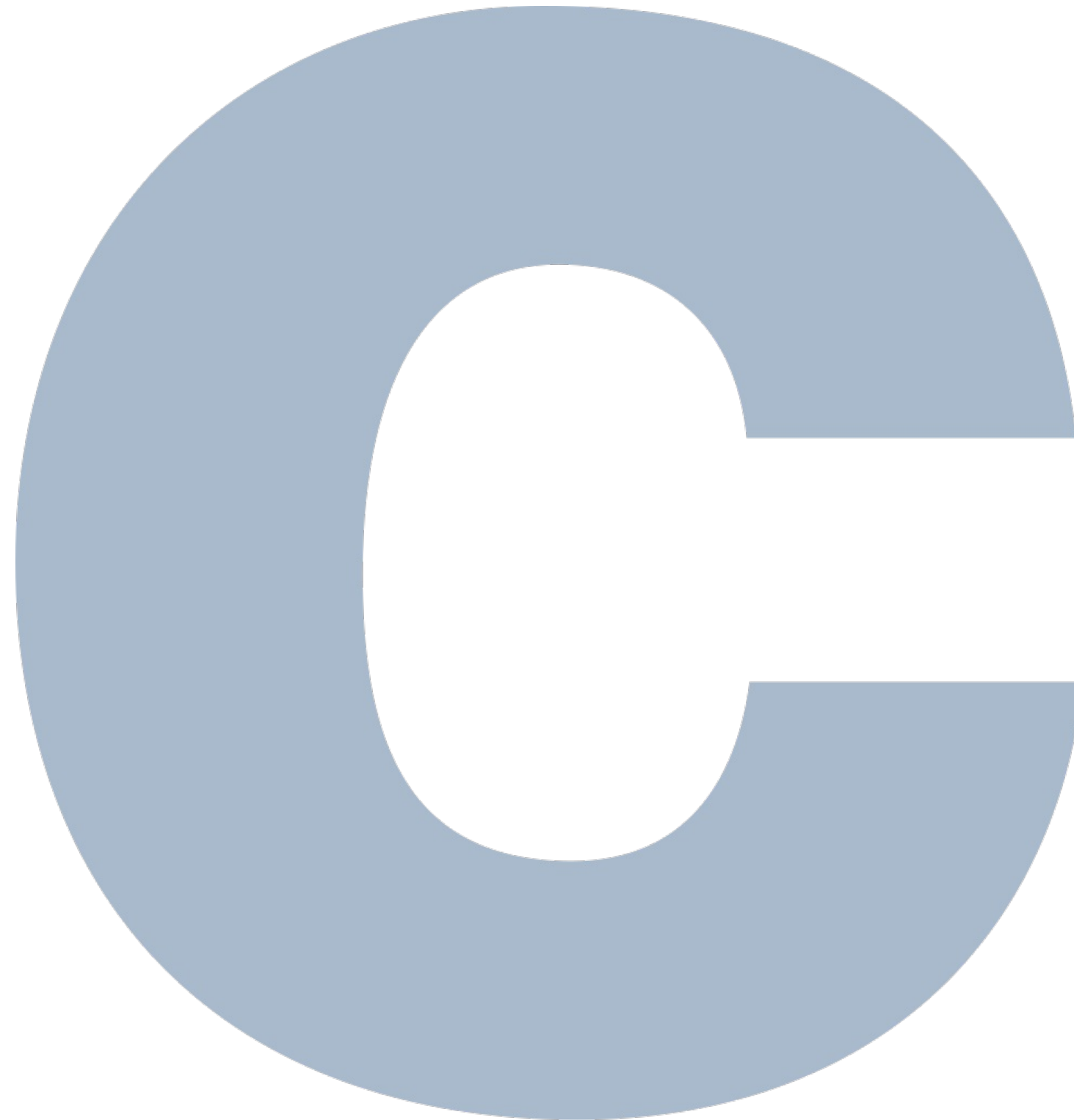


Leitura de linha em arquivo - parte 1



Objetivo e motivação

Objetivo:

- Criar uma "mini-biblioteca" C com funções para a leitura de uma linha de caracteres a partir de um arquivo de texto, com funcionalidade semelhante à `getline()` (POSIX).

Motivação:

- Dificuldade que os alunos encontram em:
 - Definição/implementação de interfaces
 - Uso de funções mais "avançadas" de C
 - Uso de arquivos e streams em C
 - Tratamento de erros
 - "Baixo nível"
- Incorporação futura na `CRpaic.h`
(github.com/computacaoraiz/CRpaic)

Pré-requisitos

- Programação C básica/intermediária
- Ponteiros (e ponteiros para ponteiros)
- Alocação de memória
 - stdlib.h**: malloc(), realloc(), free()
- Controle de buffers
 - tamanho alocado, tamanho efetivo, posição
- Tipos de dados "especiais"
 - size_t, ssize_t, FILE ***
- Bibliotecas e funções:
 - stdio.h**: fopen(), fclose(), fgetc(), ungetc(),
feof(), ferror(), perror()
 - string.h**: strerror(), memset()
 - errno.h**: errno
 - stdbool.h, sys/types.h, CRpaic.h**
- Linux man pages online (<https://man7.org>)

Projeto inicial

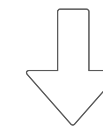
Visão geral das funcionalidades:

- **Leitura de 1 linha de caracteres de um arquivo texto, até a quebra final de linha (sem incluir a quebra), com adição do caractere NUL (\0) que delimita a string lida**
- **Tratamento de diferentes tipos de quebras de linha:**
 - \n** (UNIX, Linux, Apple atual, Android, etc.)
 - \r\n** (DOS, Windows)
 - \r** (Sistemas legados, Apple antigos, etc.)

Esta é uma linha de texto que, por causa de seu tamanho, é exibida no monitor em quatro linhas. \n

\n

Aqui começa outra linha de texto, exibida em três linhas no monitor. \n



Esta é uma linha de texto que, por causa de seu tamanho, é exibida no monitor em quatro linhas. \0

- **Proteção contra linhas gigantes e buffer overflow**
- **Tratamento de diversos tipos de erros**
- **Retornar a string lida (mesmo se for vazia: "")**
- **Retornar a quantidade de bytes lidos (sem contar a quebra de linha e o caractere NUL), em caso de sucesso**
- **Retornar indicador -1 em caso de falha, e ajustar errno**
- **A leitura pode ser de um arquivo ou de stdin**

Retornos:

- a string terminada por NUL
- 97 bytes (não conta NUL)

Obs.: 95 letras + NUL ?

Projeto inicial

Fluxo a ser seguido:

