

# 2023 (2S) - EXERCÍCIO 2 - COMPILADORES

---

## Instruções:

1. Compile o seu programa usando o compilador GCC em linha de comando;
2. Utilize o Valgrind<sup>1</sup> para verificar vazamentos de memória e se toda memória alocada foi desalocada;
3. Apresente o resultado do seu programa na linha de comando;
4. Implemente e aplique todos os testes necessários para garantir que seu programa funcione corretamente.

## Exercício 1.

Projete um programa que contenha um *buffer*. Desta vez, o tamanho do *buffer* é fixo, por exemplo 256. A estrutura de dados que conterá o *buffer* também deve guardar a posição do próximo *char* a ser lido do *buffer* e o número da linha desse *char* no arquivo original. Caso um *char* seja lido do *buffer* e não tenha sido imediatamente utilizado em uma computação, o *buffer* deve permitir receber essa informação e retroceder para a posição do *char* que não foi utilizado. Seu programa deve ser dividido em três arquivos: *main.c*, *funcs.h* e *funcs.c*.

No arquivo *main.c* você deve apenas:

1. verificar se o número de parâmetros passado para o programa na linha de comando está correto, caso contrário encerre o programa. O único parâmetro é o nome do arquivo que será lido.
2. abrir o arquivo e verificar se ele foi aberto corretamente, caso contrário encerre o programa;
3. alocar a estrutura de dados do *buffer* chamando a função *allocate\_buffer* do arquivo *funcs.c*;
4. interagir com a função *get\_next\_char*;
5. chamar a função *deallocate\_buffer* que desaloca a memória do *buffer*;
6. fechar o arquivo.

No arquivo *funcs.c*, você deve implementar as funções: *allocate\_buffer*, *deallocate\_buffer*, *get\_next\_char* e qualquer outra função necessária para o funcionamento do *buffer*. Na função *get\_next\_char*, você deve carregar o *buffer* com um bloco de *chars* que deve ter um número de *chars* igual ao tamanho do *buffer* ou um bloco que seja menor que o tamanho do *buffer* mas contenha uma linha toda, incluindo o caractere de quebra de linha. Em cada chamada, a função *get\_next\_char* retorna um *char* do *buffer*. O caractere de fim do arquivo também deve ser retornado.

No arquivo *funcs.h* você deve declarar a estrutura de dados que conterá o *buffer* e o que julgar necessário para o funcionamento do seu programa.

---

<sup>1</sup><https://www.ic.unicamp.br/~rafael/materiais/valgrind.html>