**LABORATÓRIO 2 – SYSCALLS – ALAN GLEIZER & CAIO CORSINI**

Arquivo .txt usado para exercícios 1 e 2:

A screenshot of a computer

Description automatically generated

**Exercício 1:**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdlib.h>

#define BUFFER\_SIZE 128

int main() {

// Nome do arquivo de entrada

const char \*arquivo\_entrada = "lab2\_ex1.txt";

// Descritores de arquivos

int fd\_in; // Variavel para o arquivo em si

char buffer[BUFFER\_SIZE]; // Variavel para armazenar cada linha do arquivo lido

ssize\_t bytes\_read;

// Abrindo o arquivo de entrada no modo de leitura

fd\_in = open(arquivo\_entrada, O\_RDONLY); //O\_RDONLY = read only

if (fd\_in == -1) {

perror("Não foi possível abrir o arquivo de entrada");

return EXIT\_FAILURE;

} else {

while ((bytes\_read = read(fd\_in, buffer, sizeof(buffer))) > 0) {

buffer[bytes\_read] = '\0';

printf("%s", buffer);

}

}

return EXIT\_SUCCESS;

}

A screenshot of a computer

Description automatically generated

**Exercício 2:**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdlib.h>

#define BUFFER\_SIZE 128

int main() {

// Nome do arquivo de entrada

const char \*arquivo\_entrada = "lab2\_ex2.txt";

//const char \*arquivo\_entrada = "exemplo.txt";

// Descritores de arquivos

int fd\_in; // Variavel para o arquivo em si

char buffer[BUFFER\_SIZE]; // Variavel para armazenar cada linha do arquivo lido

ssize\_t bytes\_read;

int contagem = 1;

// Abrindo o arquivo de entrada no modo de leitura

fd\_in = open(arquivo\_entrada, O\_RDONLY); //O\_RDONLY = read only

if (fd\_in == -1) {

perror("Não foi possível abrir o arquivo de entrada");

return EXIT\_FAILURE;

} else {

while ((bytes\_read = read(fd\_in, buffer, sizeof(buffer)-1)) > 0) {

for(int i=0; i<sizeof(buffer);i++){

if(buffer[i] == '\n') contagem++;

}

}

}

printf("Numero de linhas: %d", contagem);

return EXIT\_SUCCESS;

A screenshot of a computer

Description automatically generated

**Exercício 3:**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <stdlib.h>

#define BUFFER\_SIZE 128

int main() {

// Nome do arquivo de entrada

const char \*arquivo\_entrada = "lab2\_ex3\_origem.txt";

// Nome do arquivo de saída

const char \*arquivo\_saida = "lab2\_ex3\_destino.txt";

// Descritores de arquivos

int fd\_in; // Variavel do arquivo de entrada

int fd\_out; // Variavel do arquivo de saida

char buffer[BUFFER\_SIZE]; // Variavel que vai armazenar cada linha

ssize\_t bytes\_read, bytes\_written;

// Abrindo o arquivo de entrada no modo de leitura

fd\_in = open(arquivo\_entrada, O\_RDONLY); // Read only

if (fd\_in == -1) {

perror("Erro ao carregar arquivo de entrada");

return EXIT\_FAILURE;

}

// Abrindo o arquivo de saída no modo de escrita (e criar se não existir)

fd\_out = open(arquivo\_saida, O\_WRONLY | O\_CREAT | O\_TRUNC, 0644);

if (fd\_out == -1) {

perror("Erro de criacao do arquivo de saída");

close(fd\_in); // Fechando arquivo de entrada

return EXIT\_FAILURE;

}

printf("Arquivo '%s' criado!\n", arquivo\_saida);

// Lendo do arquivo de entrada e escrevendo no arquivo de saída

while ((bytes\_read = read(fd\_in, buffer, sizeof(buffer))) > 0) {

bytes\_written = write(fd\_out, buffer, bytes\_read);

if (bytes\_written != bytes\_read) {

perror("Erro de escrita");

// Fechando os arquivos

close(fd\_in);

close(fd\_out);

return EXIT\_FAILURE;

}

}

// Fechando os arquivos

close(fd\_in);

close(fd\_out);

if (bytes\_read != -1) {

printf("\nCopia de %s para '%s' feita com sucesso!\n", arquivo\_entrada, arquivo\_saida);

}

return EXIT\_SUCCESS;

}

A screenshot of a computer

Description automatically generated

Arquivo de entrada e arquivo de saída:

A screenshot of a computer

Description automatically generated