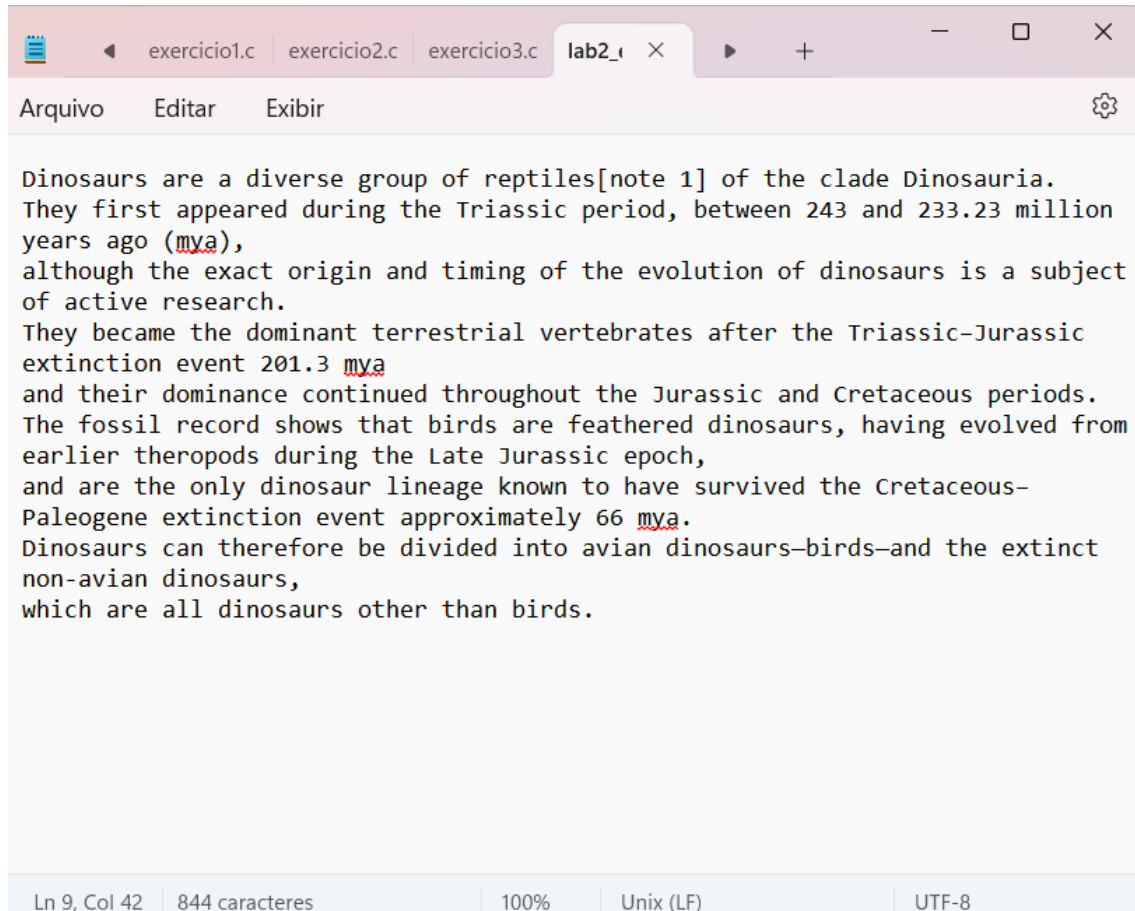


## LABORATÓRIO 2 – SYSCALLS – ALAN GLEIZER & CAIO CORSINI

Arquivo .txt usado para exercícios 1 e 2:



The screenshot shows a text editor window with a tab labeled 'lab2\_ex1'. The menu bar includes 'Arquivo', 'Editar', and 'Exibir'. The text content is as follows:

Dinosaurs are a diverse group of reptiles[[note 1](#)] of the clade Dinosauria. They first appeared during the Triassic period, between 243 and 233.23 million years ago ([mya](#)), although the exact origin and timing of the evolution of dinosaurs is a subject of active research. They became the dominant terrestrial vertebrates after the Triassic–Jurassic extinction event 201.3 [mya](#) and their dominance continued throughout the Jurassic and Cretaceous periods. The fossil record shows that birds are feathered dinosaurs, having evolved from earlier theropods during the Late Jurassic epoch, and are the only dinosaur lineage known to have survived the Cretaceous–Paleogene extinction event approximately 66 [mya](#). Dinosaurs can therefore be divided into avian dinosaurs–birds–and the extinct non-avian dinosaurs, which are all dinosaurs other than birds.

The status bar at the bottom indicates: Ln 9, Col 42 | 844 caracteres | 100% | Unix (LF) | UTF-8

### 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;

}

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```

#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;
}

```

Run
Ask AI
359ms on 16:04:57, 08/27

Dinosaurs are a diverse group of reptiles[not e 1] of the clade Dinosauria. They first appeared during the Triassic period, between 243 and 233.23 million years ago (mya), although the exact origin and timing of the evolution of dinosaurs is a subject of active research. They became the dominant terrestrial vertebrates after the Triassic-Jurassic extinction event 201.3 mya and their dominance continued throughout the Jurassic and Cretaceous periods. The fossil record shows that birds are feathered dinosaurs, having evolved from earlier theropods during the Late Jurassic epoch, and are the only dinosaur lineage known to have survived the Cretaceous-Paleogene extinction event approximately 66 mya. Dinosaurs can therefore be divided into avian dinosaurs—birds—and the extinct non-avian dinosaurs, which are all dinosaurs other than birds.

## 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;
```

```
C main.c
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5
6 #define BUFFER_SIZE 128
7
8 int main() {
9
10     // Nome do arquivo de entrada
11     const char *arquivo_entrada = "lab2_ex2.txt";
12     //const char *arquivo_entrada = "exemplo.txt";
13
14     // Descritores de arquivos
15     int fd_in; // Variavel para o arquivo em si
16     char buffer[BUFFER_SIZE]; // Variavel para armazenar cada linha do arquivo lido
17     ssize_t bytes_read;
18
19     int contagem = 1;
20
21     // Abrindo o arquivo de entrada no modo de leitura
22     fd_in = open(arquivo_entrada, O_RDONLY); //O_RDONLY = read only
23     if (fd_in == -1) {
24         perror("Não foi possível abrir o arquivo de entrada");
25     }
```

### 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 saída
```

```
    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 criação 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("\nCópia de %s para '%s' feita com sucesso!\n", arquivo_entrada, arquivo_saida);
}

return EXIT_SUCCESS;
}

```

```
37 // Lendo do arquivo de entrada e escrevendo no arquivo de saída
38 while ((bytes_read = read(fd_in, buffer, sizeof(buffer))) > 0) {
39     bytes_written = write(fd_out, buffer, bytes_read);
40     if (bytes_written != bytes_read) {
41         perror("Erro de escrita");
42         // Fechando os arquivos
43         close(fd_in);
44         close(fd_out);
45         return EXIT_FAILURE;
46     }
47 }
48
49 // Fechando os arquivos
50 close(fd_in);
51 close(fd_out);
52
53 if (bytes_read != -1) {
54     printf("\nCopia de %s para '%s' feita com sucesso!\n", arquivo_entrada,
arquivo_saida);
55 }
56
57 return EXIT_SUCCESS;
58 }
```

Run Ask AI 301ms on 16:07:03, 08/27 ✓

Arquivo 'lab2\_ex3\_destino.txt' criado!

Copia de lab2\_ex3\_origem.txt para 'lab2\_ex3\_destino.txt' feita com sucesso!

Arquivo de entrada e arquivo de saída:

lab2_ex3_destino.txt	lab2_ex3_origem.txt
1 A Arquitetura de von Neumann (de John von Neumann, pronunciado Nóimánn) é uma	1 A Arquitetura de von Neumann (de John von Neumann, pronunciado Nóimánn) é uma
2 arquitetura de computador que se caracteriza pela possibilidade de uma máquina digital	2 arquitetura de computador que se caracteriza pela possibilidade de uma máquina digital
3 armazenar seus programas no mesmo espaço de memória que os dados, podendo assim	3 armazenar seus programas no mesmo espaço de memória que os dados, podendo assim
4 manipular tais programas. Esta arquitetura é um projeto modelo de um computador digital	4 manipular tais programas. Esta arquitetura é um projeto modelo de um computador digital
5 de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de	5 de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de
6 armazenamento ("memória") para comportar, respectivamente, instruções e dados.	6 armazenamento ("memória") para comportar, respectivamente, instruções e dados.