Turma 3WA - INF1316 - 2025.1 - 9h-11h

Gabriel Patrício de Oliveira - 2310806
Gabriel Martins Mendes - 2311271

Objetivo:

1) Programar funcionalidades dos utilitários do unix - “echo” dando o nome de “meuecho” ao seu programa

Ex: $meuecho bom dia /* exibe os parâmetros de meuecho */

$bom dia

2) Programar funcionalidades do utilitário do unix “cat” dando o nome de “meucat” ao seu programa

Ex: $meucat echo.c cat.c /* exibe os arquivos echo.c e cat.c */

3) Programar as funcionalidades da shell e executar os seus programas meuecho, meucat

4) Com a sua shell executar o utilitário “ls” do Unix

Estrutura do programa:

O nosso programa é composto por 3 arquivos .c.
O principal é o "minha_shell.c" que contém o código de uma shell própria.
O segundo arquivo é o "meucat.c" que tem o código da função "cat" do utilitário do Unix.
O terceiro arquivo é o "meuecho.c" que tem o código da função "echo" do utilitário do Unix.

Solução:

Nós fizemos o código do meucat.c simplesmente abrindo um arquivo e imprimindo o conteúdo dele. O segundo código, o meuecho.c nós simplesmente printamos com printf o conteúdo que foi passado como agrv. O principal fica num while(TRUE) lendo a entrada que serão os comandos e, de acordo com o que ele ler, chama os arquivos binários gerados dos outros módulos.

Observações e Conclusões:

Não tivemos muitas dificuldades.

Sugerimos compilar com:

```
gcc -o minha_shell minha_shell.c
gcc -o meucat meucat.c
gcc -o meuecho meucho.c
```

O nosso programa funcionou em todos os casos testados.

meuecho.c

```
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
	for (int i = 1; i < argc; i++)
	{
	printf("%s\n", argv[i]);

	if (i < argc - 1)
	{
	printf("\n");
	}
	}
	return 0;
}
```

meucat.c

```
#include <stdlib.h>
#include <stdio.h>

void read_file(char* nome)
{
	FILE* f = fopen(nome, "r");

	if (!f)
	{
	printf("O arquivo nao pode ser aberto.\n");
	exit(1);
	}
	char c;
	while( (c = fgetc(f)) != EOF)
	printf("%c", c);

	fclose(f);
}
```

```
int main(int argc, char** argv)
{
	if (argc < 2)
	{
	printf("Deve ser passado ao menos um arquivo --> ./meucat <arquivo1> <arquivo2> ...
<arquivoN> ...\n");
	exit(1);
	}
	for(int i = 1; i < argc; i++)
	{
	read_file(argv[i]);
	}
	return 0;
}
```

minha_shell.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

#define MAX_INPUT 100
#define MAX_ARGS 10
#define TRUE 1

int main(void)
{
	char comando[MAX_INPUT];
	char* args[MAX_ARGS];

	while(TRUE)
	{
	printf("$ ");
	if (fgets(comando, MAX_INPUT, stdin) == NULL)
	{
	break;
	}

	comando[strcspn(comando, "\n")] = 0; // pra tirar o \n so

	int i = 0;
```

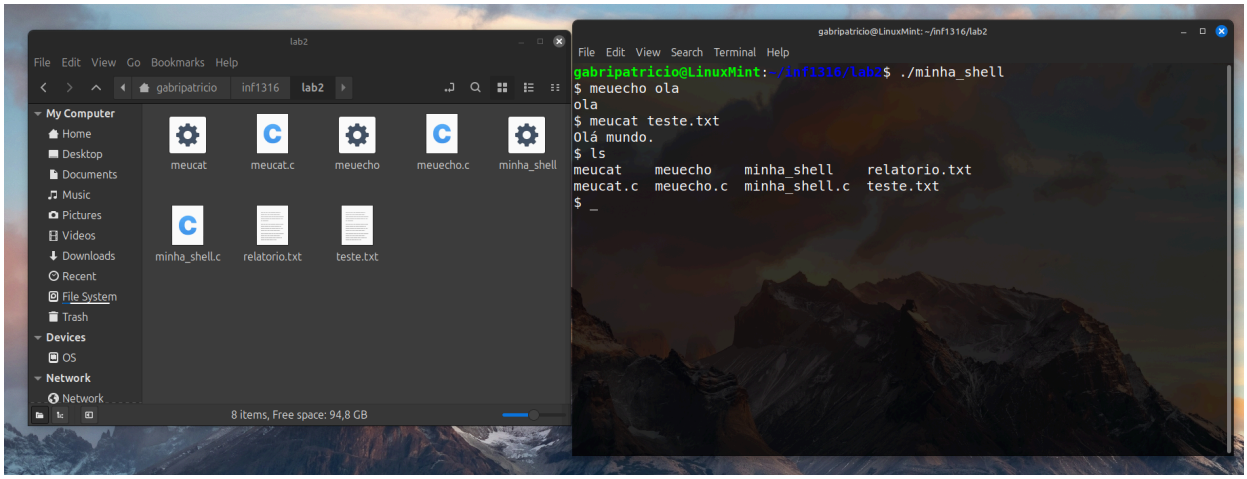
```
        char* params = strtok(comando, " ");

        while(params != NULL && i < MAX_ARGS - 1)
        {
        args[i++] = params;
        params = strtok(NULL, " ");
        }

        args[i] = NULL; // tem que ser NULL pq o execv pede

        if (fork() != 0)
        {
        wait(NULL);
        }
        else
        {
        if (strcmp(args[0], "meucat") == 0)
        {
                execv("./meucat", args);
        }
        else if (strcmp(args[0], "meuecho") == 0)
        {
                execv("./meuecho", args);
        }

        else if (strcmp(args[0], "ls") == 0)
        {
                execvp("ls", args);
        }
        else
        {
                printf("Comando não reconhecido.\n");
        }
        }
        }
        return 0;
}
```


lab2
File Edit View Go Bookmarks Help
gabripatricio
inf1316
lab2
My Computer
Home
Desktop
Documents
Music
Pictures
Videos
Downloads
Recent
File System
Trash
Devices
OS
Network
Network
meucat
meucat.c
meuecho
meuecho.c
minha_shell
minha_shell.c
relatorio.txt
teste.txt
8 items, Free space: 94,8 GB
gabripatricio@LinuxMint: ~/inf1316/lab2
File Edit View Search Terminal Help
gabripatricio@LinuxMint:~/inf1316/lab2$ ./minha_shell
$ meuecho ola
ola
$ meucat teste.txt
Olá mundo.
$ ls
meucat meuecho minha_shell relatorio.txt
meucat.c meuecho.c minha_shell.c teste.txt
$ _