Desenvolvimento de um Mini Terminal Linux

Bárbara Zamperete Oliveira, Matheus Fellype de Moura Silva

Departamento de Ciência da Computação - Universidade Federal de Roraima (UFRR)  
Boa Vista – RR – Brasil.

{Bárbara, Matheus}bahzamp25@gmail.com, matheusfellype772@gmail.com

**Abstract.** This article aims to present the functionality of the mini terminal shell, such as, what it can do, it’s implementation for how it can perform, among others.

**Resumo.** Este artigo tem como finalidade apresentar as funcionalidades do mini terminal shell, tais como, o que ele consegue fazer sua implementação para como ele consegue executar entre outros.

# 1. O que é o Shell?

Para começarmos, falaremos brevemente sobre o que é o shell. Ele é o interpretador de comandos e que irá funcionar como a interface entre o usuário e o sistema operacional. Como falamos, essa interface intermediará o usuário e o sistema operacional através de linhas de comandos. Com isso, sua funcão é ler a linha de comando, interpretar, executá-lo e devolver o resultado pelas saídas.

Para esse trabalho, criamos um mini terminal shell chamado shellso.

# 2. Inicialização e Captura do comandos do usuário

Partindo para explicação das funções do shell, vamos começar pela inicialização do shell. No main, declaramos um char chamado *user\_imput* de tamanho *MAX*, este definido como 100, ela irá armazenar o comando utilizado pelo usuário. Logo seguida a função *init\_shell* é chamada.

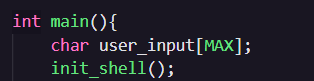
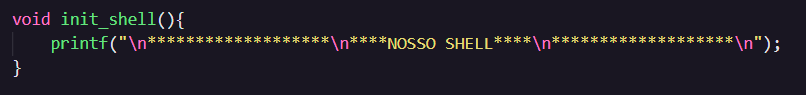


Figure 1. Main-1

A função *init\_shell* é uma simples função que mostrará um cabeçalho de início do shell.

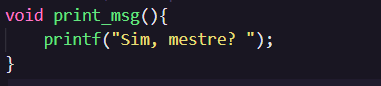
**Figure 2. InitShell**

Para a captura dos comandos digitados pelo usuário, utilizamos a função *get\_input*, que irá receber como argumento o *user\_input*  que já foi comentado. Mas antes disso, uma função chamada *print\_msg*  que irá mostrar a inicialização do prompt com a mensagem “Sim, mestre?”.



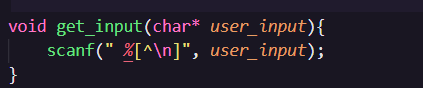
**Figure 3. Chamadas**

A função *print\_msg* é uma função que irá mostrar uma mensagem toda vez que o usuário for digitar um comando.



**Figure 4. PrintMsg**

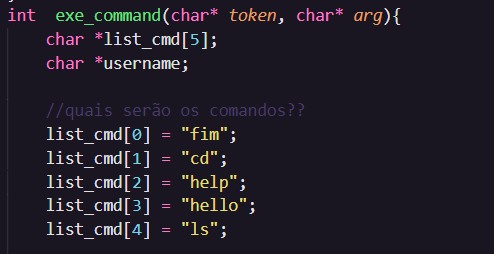
A função *get\_input* irá capturar o comando que o usuário irá digitar e ele usá como argumento o char *user\_input.*



**Figure 5. GetInput**

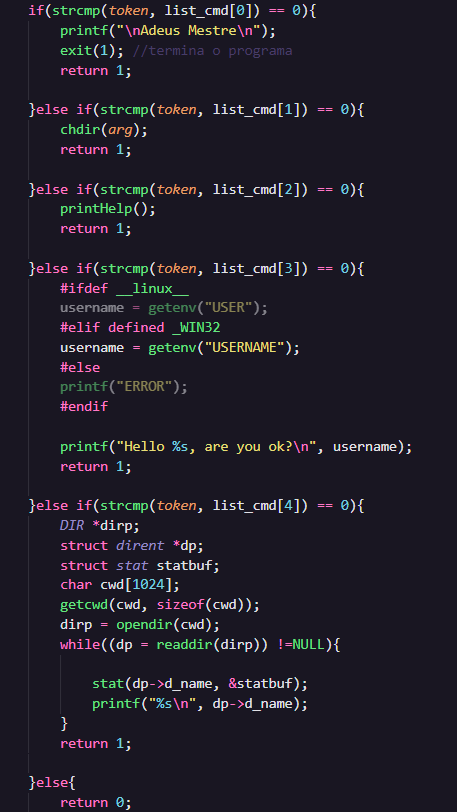
# 3. Comandos Builtins

O mini terminal shell possui alguns comandos builtins. Aguns deles são: fim, cd, help, hello, ls. Esses comandos são tratados pela função *exe\_command,* ela irá lidar com a execuções desses comandos de acordo com a opção usuário. Para isso primeiramente, criamos um ponteiro de char chamado *list\_comd* de tamanho 5, onde em cada posição estará um dos comandos citados.



**Figure 6. ExeCommand**

Essa função trata o gerenciamento de qual comando irá ser executado através das condições *if/else if.*



**Figure 6. Comandos Builtins**

# 4. Pipes

O pipe é uma das maneiras que o shell pode utilizar para comunicar processos, ou seja, ele faz um encadeamento de processos, e esse encadeamento de processos pode ser ativado quando o usuário usa o comando “|”.

No shellso, para trabalhar com pipes, ele possui duas funções: o *findPipe* que chamamos na função main e o *exePipe.*



**Figure 7. Main-findPipe**

Primeiro ele irá chamar a função  *findPipe* passando como argumento o *user\_input.* Logo quando ela entra na função, irá ser criado dois ponteiros de char chamados *pipes* que irá pegar a segunda parte do comando, ou seja, depois do “|”e o *command*  que irá pegar a primeira parte. Se houver pipes, ele irá entrar na condição, irá tratar os espaços e chamará a segunda função.



**Figure 8. FindPipe**

Quando ele chamar a segunda função, *exePipe,* onde ele irá executar os pipes utilizando a função fork, para fazer as execuções dos children e parent.



**Figure 9. ExePipe**

# 7. References

Leonardo Xavier. (2004) “Usando Pipe”, <https://www.vivaolinux.com.br/dica/Usando-o-pipe>, Dezembro.

Pedro Muxfeldt. (2017) “Linux - O Shell”, <https://br.ccm.net/contents/320-linux-o-shell>, Dezembro.

Geeks for Geeks. (2020) “Making your own Linux Shell in C”, <https://www.geeksforgeeks.org/making-linux-shell-c/?ref=lbp>, Dezembro.