# Alunos: Caio Rangel e Rafael Donner

# Relatório

## Menu Principal

### Descrição

O menu principal do script é a interface inicial exibida ao usuário. Segue o padrão exigido no trabalho e, por conta do uso do comando tput, a alteração na data e hora não interfere no alinhamento da interface, garantindo que as informações sejam exibidas de forma clara e organizada.

### Comandos Usados

* printf: Exibe a interface do menu principal com informações formatadas.
* date: Coleta a data e hora atuais do sistema.
* tput: Controla a posição do cursor, organizando visualmente os campos de data e hora.
* read: Captura a entrada do usuário para selecionar uma opção de menu.
* clear: Limpa o terminal antes de exibir a interface do menu, proporcionando uma experiência visual mais limpa.

## Opção 1: Listagem de diretórios

### Descrição

O script consiste em listar arquivos e(ou) diretórios em um caminho dado pelo usuário. O usuário interage com o programa através de um menu que oferece opções para listar apenas arquivos, apenas diretórios, ambos ou sair do programa. O script verifica as preferências do usuário sobre o detalhamento da listagem (detalhado ou simples) e executa as listagens usando comandos de terminal.

### Comandos Usados

* find: Realiza buscas em diretórios específicos com filtros de tipo, como arquivos ou diretórios. A opção -maxdepth limita a profundidade da busca, enquanto -mindepth 1 exclui o diretório raiz da busca. É usado para detalhagem simples.
* ls -lh: Lista o conteúdo do diretório de forma detalhada (longa) e com tamanhos legíveis.
* grep: Filtra o conteúdo do comando ls para identificar apenas arquivos ('^-') ou apenas diretórios ('^d'). Este filtro é usado apenas quando a listagem detalhada é desejada e serve para separar arquivos de diretórios.
* read: Coleta as escolhas do usuário sobre o tipo e detalhamento da listagem.
* clear: Limpa a tela antes de exibir o conteúdo do menu ou dos resultados da busca.

### Observações

* **! -name '.\*'**: O filtro ! -name '.\*' impede que arquivos e diretórios ocultos sejam incluídos na listagem.
* **Pipe (|)**: O pipe é utilizado para direcionar a saída de um comando para a entrada de outro.
* **-z e -d**:
  + **-z**: A opção -z é usada no comando test ou [[ ... ]] para verificar se uma variável está vazia.
  + **-d**: O teste -d verifica se um caminho corresponde a um diretório.

## Opção 2: Copiar um arquivo para um subdiretório

### Descrição

A **Opção 2** do menu permite ao usuário copiar um arquivo do diretório atual para um subdiretório de sua escolha. O script exibe todos os arquivos do diretório atual e permite ao usuário selecionar um arquivo a ser copiado. Em seguida, o script apresenta os subdiretórios disponíveis. Após a escolha, o arquivo é copiado para o diretório de destino, com uma mensagem de confirmação da operação.

### Comandos Usados

* find: Usado para localizar arquivos e diretórios no diretório atual. A opção -maxdepth 1 é usada para listar apenas os arquivos no nível atual, sem entrar em subdiretórios.
* cp: Comando utilizado para copiar o arquivo selecionado para o diretório de destino.
* read: Captura as entradas do usuário.
* basename: Usado para extrair o nome do arquivo ou diretório a partir de seu caminho completo.

## Opção 3: Renomear um arquivo

### Descrição

A **Opção 3** permite ao usuário renomear um arquivo do diretório atual. O script permite ao usuário selecionar um arquivo para renomear. O usuário deve fornecer um novo nome para o arquivo. Se o novo nome for diferente do nome atual, o arquivo é renomeado.

### Comandos Usados

* mv: O comando mv é utilizado para mover ou renomear arquivos e diretórios. Neste caso, é usado para renomear o arquivo selecionado para o novo nome fornecido pelo usuário.
* read: Captura a entrada do usuário.
* basename: Usado para extrair o nome do arquivo, sem o caminho completo, para exibição na mensagem de confirmação.

## Opção 4: Deletar arquivos ou diretórios

### Descrição

Na opção 4, o script exibe uma lista numerada de todos os itens disponíveis no diretório atual, e o usuário escolhe um item. O programa verifica se é um arquivo ou um diretório e, caso seja um diretório, informa ao usuário que todos os arquivos e subdiretórios internos serão removidos juntos. Antes de deletar, o script solicita a confirmação do usuário.

### Comandos Usados

* find: Utilizado para listar arquivos e diretórios no diretório atual com profundidade máxima e mínima de 1, para que, respectivamente, apenas os itens do diretório atual sejam exibidos e o diretório . (atual) não seja exibido.
* basename: Exibe apenas o nome do arquivo ou diretório, sem o caminho completo, para facilitar a visualização e seleção pelo usuário.
* rm e rm -r: rm é usado para excluir arquivos, enquanto rm -r é usado para remover diretórios recursivamente, incluindo todo o conteúdo interno, caso o usuário tenha confirmado a exclusão.
* read: Captura as entradas do usuário.

### Observações

* -d e -f: Utilizados para verificar se o item selecionado é um diretório (-d) ou um arquivo (-f).

## Referências

* Documentação do comando tput: [tput usage documentation](https://man7.org/linux/man-pages/man1/tput.1.html)
* Documentação do comando find: [GNU find documentation](https://www.gnu.org/software/findutils/)
* Documentação do comando test e [[ ... ]]: [Bash test documentation](https://www.gnu.org/software/bash/manual/html_node/Bash-Builtins.html#index-test)
* Documentação do comando basename: [GNU basename documentation](https://www.gnu.org/software/coreutils/manual/html_node/basename-invocation.html)

## Script:

#!/bin/bash

trap '' SIGINT

trap '' SIGTSTP

while true; do

clear

printf "###############################################################\n"

printf "# IBMEC #\n"

printf "# Sistemas Operacionais Semestre 2 de 2024 #\n"

printf "# Código: IBM8940 Turma 8001 #\n"

printf "# Professor: Luiz Fernando T. de Farias #\n"

printf "#-------------------------------------------------------------#\n"

printf "# Equipe Desenvolvedora: #\n"

printf "# Aluno: Caio Rangel #\n"

printf "# Aluno: Rafael Donner #\n"

printf "#-------------------------------------------------------------#\n"

printf "# Rio de Janeiro, %s de %s de %s" "$(date +%d)" "$(date +%B)" "$(date +%Y)"

tput cup 11 62

printf "#\n"

printf "# Hora do Sistema: %d Horas e %d Minutos " "$(date +%H)" "$(date +%M)"

tput cup 12 62

printf "#\n"

printf "###############################################################\n\n"

printf "Menu de Escolhas:\n"

printf " 1) Listar conteúdo de um diretório.\n"

printf " 2) Copiar um arquivo do diretório atual para um subdiretório.\n"

printf " 3) Renomear um arquivo do diretório atual.\n"

printf " 4) Excluir arquivos ou diretórios do diretório atual.\n"

printf " 5) Finalizar o programa.\n\n"

printf "Selecione uma opção: "

read opcao

if ! [[ "$opcao" =~ ^[1-5]+$ ]]; then

printf "Erro: %s não é uma opção válida\n" "$opcao"

elif [ "$opcao" = "1" ]; then

clear

while true; do

printf "\nPara voltar ao menu principal digite 4.\nDigite o caminho do diretório a ser listado: \n"

read dir

if [ "$dir" = "4" ]; then

break

elif [ -d "$dir" ]; then

printf "\nDigite 1 para listar apenas arquivos, 2 para listar apenas diretórios, 3 para ambos, e 4 para voltar ao menu principal.\n"

read tipoListagem

if [ "$tipoListagem" = "4" ]; then

break

elif [ "$tipoListagem" != "3" ] && [ "$tipoListagem" != "2" ] && [ "$tipoListagem" != "1" ]; then

printf "Erro: a opção %s não existe.\n" "$tipoListagem"

continue

fi

printf "\nDeseja uma listagem detalhada? (s/n): "

read detalhagem

if [ "$detalhagem" != "s" ] && [ "$detalhagem" != "n" ]; then

printf "Erro: comando inválido digitado.\nDigite s ou n.\n"

continue

fi

if [ "$tipoListagem" = "1" ]; then

if [ "$detalhagem" = "s" ]; then

resultado=$(ls -lh "$dir" | grep '^-') # Lista apenas arquivos com detalhes

else

resultado=$(find "$dir" -maxdepth 1 -type f ! -name '.\*') # Lista arquivos sem subdiretórios

fi

elif [ "$tipoListagem" = "2" ]; then

if [ "$detalhagem" = "s" ]; then

resultado=$(ls -lh "$dir" | grep '^d') # Lista apenas diretórios com detalhes

else

resultado=$(find "$dir" -maxdepth 1 -mindepth 1 -type d ! -name '.\*') # Lista apenas diretórios

fi

elif [ "$tipoListagem" = "3" ]; then

if [ "$detalhagem" = "s" ]; then

resultado=$(ls -lh "$dir") # Lista todos com detalhes

else

resultado=$(find "$dir" -maxdepth 1 -mindepth 1 ! -name '.\*') # Lista todos sem detalhes

fi

fi

if [ -z "$resultado" ]; then

printf "Nada foi encontrado com os seus critérios de busca. \n"

else

printf "%s\n" "$resultado"

fi

elif [ "$dir" = "4" ]; then

break

else

printf "Erro: O diretório %s não existe, ou é outro tipo de arquivo.\n\n" "$dir"

fi

done

elif [ "$opcao" = "2" ]; then

clear

while true; do

# Exibir arquivos disponíveis

printf "\nSelecione um arquivo para copiar:\n"

files=($(find . -maxdepth 1 -type f))

if [ "${#files[@]}" -eq 0 ]; then

printf "Nenhum arquivo encontrado no diretório atual.\n"

else

for i in "${!files[@]}"; do

printf "%d) %s\n" $((i+1)) "$(basename "${files[$i]}")"

done

fi

# Solicitar o arquivo a ser copiado

printf "\nDigite o número do arquivo que deseja copiar ou 4a para voltar ao menu principal: "

read file\_num

if [ "$file\_num" = "4a" ]; then

break

elif [[ "$file\_num" =~ ^[0-9]+$ ]] && [ "$file\_num" -ge 1 ] && [ "$file\_num" -le "${#files[@]}" ]; then

source\_file="${files[$((file\_num-1))]}"

else

printf "Opção inválida.\n"

continue

fi

# Exibir diretórios de destino

printf "\nSelecione o diretório de destino (ou digite 4a para voltar ao menu principal):\n"

dirs=($(find . -maxdepth 1 -mindepth 1 -type d))

for i in "${!dirs[@]}"; do

printf "%d) %s\n" $((i+1)) "$(basename "${dirs[$i]}")"

done

printf "Digite o número do diretório de destino ou 4a para voltar: "

read target\_dir\_num

if [ "$target\_dir\_num" = "4a" ]; then

break

elif [[ "$target\_dir\_num" =~ ^[0-9]+$ ]] && [ "$target\_dir\_num" -ge 1 ] && [ "$target\_dir\_num" -le "${#dirs[@]}" ]; then

target\_dir="${dirs[$((target\_dir\_num-1))]}"

cp "$source\_file" "$target\_dir"

printf "Arquivo %s copiado para o diretório %s.\n" "$(basename "$source\_file")" "$(basename "$target\_dir")"

printf "Para voltar ao menu principal digite qualquer tecla: "

read saida

if [ "$saida" = "4" ]; then

break

else

break

fi

else

printf "Opção inválida.\n"

fi

done

elif [ "$opcao" = "3" ]; then

clear

while true; do

printf "\nEscolha um arquivo para renomear:\n"

files=($(find . -maxdepth 1 -type f))

if [ "${#files[@]}" -eq 0 ]; then

printf "Nenhum arquivo encontrado no diretório atual.\n"

else

for i in "${!files[@]}"; do

printf "%d) %s\n" $((i+1)) "$(basename "${files[$i]}")"

done

fi

printf "\nDigite o número do arquivo que deseja renomear ou 4a para voltar ao menu principal: "

read file\_num

if [ "$file\_num" = "4a" ]; then

break

elif [[ "$file\_num" =~ ^[0-9]+$ ]] && [ "$file\_num" -ge 1 ] && [ "$file\_num" -le "${#files[@]}" ]; then

old\_name="${files[$((file\_num-1))]}"

else

printf "Opção inválida.\n"

continue

fi

printf "Digite o novo nome para o arquivo ou 4 para voltar ao menu principal: "

read new\_name

if [ "$new\_name" = "4" ]; then

break

elif [ -n "$new\_name" ]; then

if [ "$new\_name" = "$(basename "$old\_name")" ]; then

printf "O novo nome é igual ao nome atual.\n"

else

mv "$old\_name" "$new\_name"

printf "Arquivo renomeado de %s para %s.\n" "$(basename "$old\_name")" "$new\_name"

fi

printf "Para voltar ao menu principal digite qualquer tecla.\n"

read sair

if [ "$sair" = "4" ]; then

break

else

break

fi

else

printf "Erro: Nome inválido.\n"

fi

done

elif [ "$opcao" = "4" ]; then

clear

while true; do

printf "\nEscolha um arquivo ou diretório para excluir:\n"

items=($(find . -maxdepth 1 -mindepth 1)) # Lista arquivos e diretórios no diretório atual

if [ "${#items[@]}" -eq 0 ]; then

printf "Nenhum item encontrado no diretório atual.\n"

break

else

for i in "${!items[@]}"; do

printf "%d) %s\n" $((i+1)) "$(basename "${items[$i]}")"

done

fi

printf "\nDigite o número do item que deseja excluir ou 4a para voltar ao menu principal: "

read item\_num

if [ "$item\_num" = "4a" ]; then

break

elif [[ "$item\_num" =~ ^[0-9]+$ ]] && [ "$item\_num" -ge 1 ] && [ "$item\_num" -le "${#items[@]}" ]; then

item="${items[$((item\_num-1))]}"

if [ -d "$item" ]; then

printf "Se estiver apagando um diretório, todos os seus arquivos e subdiretórios também serâo exluídos.\n"

fi

printf "Tem certeza de que deseja excluir %s? (s/n): " "$(basename "$item")"

read confirmacao

if [ "$confirmacao" = "s" ]; then

if [ -d "$item" ]; then

rm -r "$item"

printf "Diretório %s excluído com sucesso.\n" "$(basename "$item")"

else

rm "$item"

printf "Arquivo %s excluído com sucesso.\n" "$(basename "$item")"

fi

else

printf "Exclusão cancelada.\n"

fi

printf "Pressione qualquer tecla para voltar ao menu principal.\n"

read saindo

if [ "$saindo" = "4" ]; then

break

else

break

fi

else

printf "Opção inválida.\n"

fi

done elif [ "$opcao" = "5" ]; then

printf "Programa finalizado.\n"

trap - SIGINT

trap - SIGTSTP

break

fi

done