Estrutura Geral

Alunos : Estevão Moraes de Andrade , Gabriel Alves Fernandes

O trabalho pratico 1 refere-se a controle de processos e como proposto foi gerada uma “mini-shell” desenvolvida basicamente em dois arquivos, que são o main.c e funtions.c.
O main.c possui o código que conduz a shell e basicamente esse arquivo contem a leitura da linha de prompt, a tradução do texto em argumentos e a chamada de um processo que inicia a execução dos comandos. No arquivo funtions.c se encontram as funções principais, como a função executarComando, e as funções auxiliares.

Funções Principais:

executarComandos
Está função recebe os argumentos e a quantidade de pipes na linha de comando. Dentro dessa função são feitas as chamadas para as demais Situações, como os redirecionamentos.

verificaSinal
Está função recebe como argumentos os simbolos de redirecionamento e valida qual deve ser a função executada,se é a de leitura, a de escrita, ou são ambas.

executaUmProcesso
Função que é chamada em executarComandos em casos que a linha de comando não possui pipes.

fazLeitura, fazEscrita e **leDepoisFazEscrita**
Funções responsáveis pela execução dos processos de redirecionamentos, sendo um para cada tipo de simbolo (“<=”, “=>”, “<= arq =>”).

Funções Auxiliares

Funções criadas para modularização do codigo. Elas auxiliam as funções principais na execução de tarefas e seus nomes descrevem sua finalidade.
São elas:
- countWords:
- removeSimboloBackground
- splitVetor
- localizaRedirecionamentos
- encontraPipe
- quantidadePipe
- parseCommands

A sequencia de execução acontece da seguinte forma :

- O prompt é mostrado.
- O comando é lido e dividido em argumentos e pipes .
- É chamada a funçao “**executaComando()**”, que recebe os argumentos e a quantidades de pipes da linha de comando. Dentro dessa função são feitas as chamadas para as demais situações. Se não são encontrados pipes a função **executaUmProcesso()** é chamada.
. Se um redirecionamento é encontrado a função **verificaSinal()** é chamada recebendo como argumentos os simbolos de redirecionamento, validando qual das funções **fazLeitura()**, **fazEscrita()** ou **leDepoisFazEscrita()** deve ser executada.
- Ao final disso é executado o comando, se for para executar em background o código do main é preparado para mostrar o prompt e é automaticamente finalizado quando um outro codigo é chamado.

Considerações:
O sistema é bem simples e utiliza fork() para gerar um processo filho e por meio dele executar o comando . O dup2 é usado tanto para ligar as saídas e entradas do pipe, quanto para linkar as saídas de escrita do prompt com as entradas de escrita dos arquivos para fazer os redirecionamentos.
É importante comentar que todo processamento de redirecionamentos é usado como produtor / consumidor, utilizando o dup2 e ao mesmo tempo fazendo um while(read())ou(write()). Que recebe cada caracter da saída do pipe e escreve no processo que vai executar o comando.

Bug conhecido:
Bug na execução em background, em teoria os processos deveriam rodar por trás, ou seja, após a chamada do comando, o prompt apareceria novamente e o resultado só seria mostrado quando o processo finalizasse, porém, não está sendo garantido que se executado um segundo processo ocorrerá a interrupção daquele que já estava rodando em background. Porque o comando para pegar esse processo finaliza antes da chamada do proximo processo.