

Estrutura de Dados

Relatório do Trabalho

Introdução

Desde pequenos somos apresentados ao mundo da matemática, que utilizamos no cotidiano para abstrair algumas ideias que a princípio seriam bastante complexas, desde contar a quantidade de objetos até a simulação de vigas, circuitos, condução de calor e diversas outras aplicações. Em resumo, o mundo vive a matemática.

Contudo, neste caminhar vemos que as contas ficam bastante complexas para se fazer à mão e se torna inviável efetuar todas as contas manualmente e então inventou-se a [La pascaline](#), a primeira máquina de calcular. Desde então, as calculadoras foram desenvolvidas para efetuar contas de maneira mais rápida e diminuir o esforço de quem as utiliza. As calculadoras(ou computadores, as quais são máquinas de computar, calcular) modernas utilizam circuitos e toda uma teoria de [Circuitos Digitais](#) com portas lógicas.

Para transformar expressões que utilizamos na matemática da melhor maneira para que o computador entenda, utilizamos uma outra notação e a transformação da expressão nesta notação é o objetivo deste trabalho. Neste trabalho abordaremos a transformação de uma expressão infixa para uma expressão pós-fixa, depois o seu cálculo, caso possível.

Utilização do programa

Execução

Para compilar o arquivo em C, utiliza-se o comando:

```
gcc -Wall -ansi final.c -o executavel
```

Para executar o arquivo para verificar vazamento de memória(pois no programa há alocação dinâmica de memória), executa-se através de:

```
valgrind --leak-check=full ./executavel
```

Para melhor visualização dos resultados, troque a função *limpa_tela* no arquivo *final.c*

Navegação entre menus

Quando se executa o programa, aparece uma tela como mostrada pela Figura :

Figura 1: Menu principal

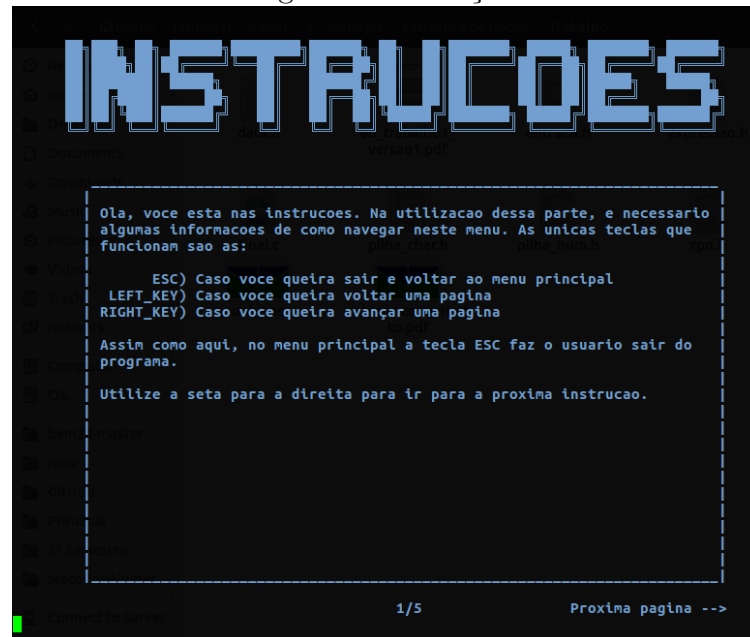


Para selecionar uma das opções, você pode digitar o número associado. Por exemplo, para entrar na calculadora de expressões, basta digitar a tecla 1. Em instruções também há explicações, pode segui-las caso queira.

Instruções

Uma vez que entrou no menu de Instruções, você pode navegar entre páginas com as setas direcionais para esquerda e direita. Com a utilização das teclas, você pode alterar entre as páginas das instruções. Um exemplo de imagem é mostrado pela Figura 2.

Figura 2: Instruções



Calculadora modo RPN

Uma vez que você entrou na calculadora modo RPN, é aparece um menu assim como indica a Figura 3.

Figura 3: Menu principal



Para sair deste modo, basta digitar “exit” e apertar *ENTER* que então retornará ao menu principal.

Essa é uma calculadora, como o nome já diz, no modo RPN. Para utiliza-la, comece digitando dois numeros. Primeiro você digita o numero 3, aperta *ENTER* e depois digite 2 e novamente *ENTER*. Você perceberá que dois numeros foram adicionados à pilha. Ficando o número 3 na posição 2 e o número 2 na posição 1. Agora, digite o simbolo + e aperte

ENTER. Você verá que mostrará agora na posição 1 o número 5, que é resultado da soma de 2 com 3.

Agora, testamos outro exemplo: Com o 5 na pilha, agora digite 6 e aperte *ENTER*. Após isso, digite o símbolo $-$ e aperte *ENTER*. Você verá que apareceu o número -1 na posição 1. Um detalhe que devemos notar é que no caso da subtração e divisão é que a ordem importa enquanto a soma e multiplicação independem da ordem utilizada. Sempre as operações(+, $-$, $*$ e $/$) ocorrem com o elemento na posição 2 com a posição 1. Por exemplo, sendo A na posição 2 e B na posição 1, então digitando $-$ obtém-se $A - B$.

Se por exemplo só existir um termo e tentar fazer qualquer operação que necessitam de 2 operandos, o programa vai reclamar e não fazer a operação, mantendo o ultimo número intacto.

Outra coisa que vale ressaltar, se mais de 15 números forem digitados, embora eles não sejam todos mostrados na tela, o usuário ainda poderá acessá-los. Para acessá-los, é necessário utilizar todos os números já digitados.

Um detalhe adicional é que o programa repete o ultimo comando do usuário ao apertar *ENTER* sem nada na caixa. Por exemplo, se digitar o número 2 e apertar 2 vezes o *ENTER*, então aparecerá duas vezes o número 2 na pilha. Esse meio também serve para o caso de efetuar a mesma operação várias vezes. Por exemplo, se houver 3 números na pilha e então digitar $+$, *ENTER*, *ENTER*, então aparecerá na pilha a soma dos três números.

No caso de digitar uma entrada inválida(seja não número ou uma operação inválida), então é acrescentado o número 0 na pilha.

Calculadora de Expressões

Quando se entra na calculadora de expressões, aparece o menu como mostrado pela Figura 4.

Figura 4: Menu principal



Para sair deste modo, basta digitar “*exit*” e apertar *ENTER* que então retornará ao menu principal.

Neste modo, o usuário entra com uma expressão numérica, como por exemplo

$$3*(4-7*2/(4+3))-9*4$$

que deve resultar em -30 , e então calcula passo a passo mostrando ao usuário a leitura e transformação dos dados. Para fazer isso, é primeiro feito a leitura completa e é então colocado em duas pilhas: uma numérica, outra simbólica.

Toda vez que um número é reconhecido, o número é adicionado à pilha numérica e é acrescentado um n à pilha simbólica.

Caso seja encontrado um símbolo, ele é acrescentado na pilha simbólica e neste caso independe da ordem. Neste processo, caso seja encontrado uma letra ou caracter inválido, então o programa indica que a entrada é inválida. Caso a quantidade de (e) sejam diferentes, é indicado também a invalidade da entrada.

No caso de conseguir acrescentar todos os elementos às pilhas, então ele começa a efetuar as operações de transformar a notação infixa para pós-fixa utilizando duas pilhas auxiliares chamadas de *PILHA_NUM* e *PILHA_CHAR* em um escopo chamado *RPN*.

Alguns exemplos que dão errado quando se tenta calcular pois faltam parenteses:

$3*(4-7*2/(4+3))-9*4$

Ou este quando falta-se operandos:

$3*(4-7*2/(4+3))-9*$

Considerações adicionais

Para facilitar a correção, pode-se trocar a função `limpa_tela()` no arquivo `final.c` para no lugar de `system("clear")`, seja apenas imprimido uma quebra de linha e verificar as mensagens do debugador.

Para mudar a velocidade de análise que o programa faz, basta alterar o valor do `define delay`. Se colocar 0, ele imprimirá apenas a resposta final, sem os passos intermediários.