

Relatorio da calculadora RPN

Bernardo Marques
nº 40535

Bruno Ascensão
nº 45460

31 de Maio de 2020



1 Introdução

Este trabalho foi realizado no âmbito da unidade curricular de Arquitetura de Sistemas e Computadores I com o intuito de criar uma calculadora em notação polaca inversa (RPN), implementado em assembly MIPS.

A RPN é uma notação onde os números são introduzidos primeiro seguidos dos operadores. Por exemplo em vez de ter $1 + 1$, tem-se $1\ 1\ +$.

Esta calculadora deve aceitar um input que pode ser de números ou de operadores em string, mettendo os números numa pilha e usando os operadores para aplicar nos números mais no topo da pilha, substituindo os mesmos pelo resultado.

2 Desenvolvimento

No desenvolvimento da calculadora em assembly MIPS tentamos seguir-nos pela mesma estrutura implementada em C.

Não conseguimos implementar a calculadora a 100 percento, daí ela não funcionar. O maior problema está na implementação da pilha onde seriam guardados os números inseridos pelo utilizador. A outra parte do programa que não funciona é na comparação dos operadores inseridos pelo utilizador com as operações existentes em string, cujo programa não os identifica.

O programa acenta-se num loop infinito até o utilizador inserir a operação 'off'. Neste loop o programa começa por fazer print dos números existentes na pilha, e após acabar o print pede o input. Como o input inserido vem no formato de string tem que se separar a cada espaço e converter para inteiro caso seja um número e inserir na pilha, ou no caso de ser um operador executar a respetiva operação. Após acabar de ler todo o input e executar todas as operações o programa volta a pedir input ao utilizador.

- **Nome: main**

Sinopse: função main onde o programa é iniciado;

- **Nome: mainLoop**

Sinopse: onde vai ocorrer o loop principal do programa;

- **Nome: strRead**

Sinopse: onde vai iniciar a leitura da string introduzida pelo utilizador;

- **Nome: loopString**

Sinopse: loop onde vai ocorrer a leitura da string que vai separar por espaços e introduzir a parte separada para avaliação

- **Nome: endLoop**

Sinopse: onde vai terminar o loop caso encontre um espaço ou fim do input;

- **Nome: input**

Sinopse: função que aceita o input do utilizador, com os devidos prompts;

- **Nome: printStack**

Sinopse: onde inicia a leitura da pilha da calculadora para demonstrar ao utilizador

- **Nome: printStackLoop**

Sinopse: loop principal para leitura da pilha

- **Nome: emptyStack**

Sinopse: no caso de a pilha estar vazia faz print a indica-lo

- **Nome: push**

Sinopse: introduz no topo da pilha um número aceitado como argumento, avaliando primeiro se a pilha está cheia ou se pode introduzir o número;

- **Nome: pop**

Sinopse: remove um numero do topo da pilha e retorna-o, avaliando primeiro se esta está vazia ou se pode remover o numero;

- **Nome: checker**

Sinopse: função que avalia a string formada em loopString, que representa um numero ou operador lido do input, o qual vai verificar se deve meter na pilha se for numero ou se deve chamar uma função para executar a operação;

- **Nome: resumo**

Sinopse: caso não seja um numero resumir checker aqui para identificar a operação ou se input invalido;

- **Nome: strcmp**

Sinopse: função para comparação de strings

- **Nome: success**

Sinopse: extensão de strcmp, devolve 1 se as strings forem iguais

- **Nome: different**

Sinopse: extensão de strcmp, devolve 0 se as strings forem diferentes

- **Nome: isDigit**

Sinopse: vai avaliar se a string é um numero e vai transformar essa string em integer;

3 Conclusao

No desenvolvimento desta segunda parte do projeto pudemos por em pratica a materia lecionada na cadeira de ASC1 sobre a arquitetura MIPS.

Não foi possível fazer o programa por completo, pois não foi implementado uma das operações (clear) e alguns erros que fazem com que o programa não faça o que se pretende, como a implementação da pilha não estar completa e a identificação das operações não funcionar.

Concluindo conseguimos perceber um pouco mais como a memoria de um computador funciona e aprofundamos os nossos conhecimentos de arquitetura de computadores apesar de algumas dificuldades.