

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Lucas Araújo de Oliveira 12221BCC025
Gustavo Marques Oliveira 12221BCC021

MEMORIAL DESCRITIVO DA ATIVIDADE PRÁTICA EM MIPS ASSEMBLY -PARTE 2

UBERLÂNDIA
2024

Problema 1)

A resolução desse problema tem como objetivo, contar o número de caracteres presentes na string definida na seção (.data). O programa define a função main como o ponto de entrada principal usando .globl main. Dentro da função, há a inicialização de dois registradores. O registrador \$t0 é inicializado com zero, atuando como um contador de caracteres, e \$t1 é carregado com o endereço inicial da string definida "OLA".

O programa então entra em um loop que carrega um byte da string em \$t2, verifica se esse byte é o terminador nulo, incrementa o contador de caracteres em \$t0 e avança para o próximo caractere na string. Esse loop continua até que o caractere nulo seja encontrado.

Após a conclusão do loop, o registrador \$t0 contém o número total de caracteres na string "OLA". Esse valor é movido para \$a0, preparando-se para a impressão, e a syscall é utilizada para imprimir o valor como um inteiro.

Em seguida, o código imprime um caractere de nova linha no console usando a syscall. Finalmente, o programa encerra sua execução usando a syscall para terminar o programa.

Problema 2)

O principal objetivo da solução desse problema é modificar uma string fornecida como entrada, convertendo todos os caracteres maiúsculos para minúsculos. Para a resolução, o algoritmo modifica uma string inserida para converter todos os caracteres maiúsculos e minúsculos. Ele percorre a string caractere por caractere e, se um caractere estiver no intervalo de letras maiúsculas, ele o converte para minúsculas adicionando 32 ao valor ASCII do caractere. Após a conversão, a string modificada é impressa no console junto com uma nova linha. A seção (.data) define as variáveis necessárias para o programa, incluindo a string inicial "ArqUitetura", seu comprimento (11), strings para representar "a" e "A", e uma string de nova linha.

Na seção (.text), a execução começa na main. Inicialmente, os registradores são inicializados com os endereços das variáveis. Um loop é então usado para percorrer cada caractere da string. Dentro deste loop, cada caractere é verificado para determinar se é uma letra maiúscula. Se for, o código converte esse caractere para minúsculo e atualiza a string. O loop continua até encontrar o caractere nulo, indicando o final da string.

Problema 3)

Esse problema tem como objetivo apresentar um algoritmo para calcular uma aproximação do valor do seno de um ângulo que é apresentado em graus usando a série de Taylor. Para essa resolução, o programa começa solicitando ao usuário para inserir um ângulo em graus. Em seguida, realiza a conversão desse ângulo para radianos, utilizando a lógica da fórmula que converte esse valor multiplicando o ângulo pela minha constante pi e dividindo esse valor por 180.

Com o ângulo convertido para radianos, o programa entra em um loop que calcula os termos da série de Taylor para o seno. Dentro do loop, o programa calcula cada termo da série de Taylor iterativamente, atualizando o resultado e o denominador para cada termo calculado. O cálculo do termo é feito até que o valor absoluto do termo calculado seja menor que uma condição de parada, que é definida como (0.00001). Após o término do cálculo, o programa imprime o resultado final no console como a aproximação do valor do seno do ângulo fornecido em graus.

Problema 4)

Para esse problema, o código tem como objetivo calcular o cosseno de um ângulo fornecido pelo usuário, usando a série de Taylor para o cosseno. Na implementação foram definidas constantes com um valor fixo para o problema, como o valor de pi, constantes para o cálculo do ângulo em radianos e uma constante de precisão para determinar quando parar o cálculo da série.

O programa exibe uma mensagem para o usuário inserir o valor do ângulo em radianos. Depois de ler esse valor fornecido, o programa chama uma função (conversao) para converter o ângulo fornecido de graus para radianos.

Após a conversão, o programa chama uma função (cosex) para calcular o cosseno do ângulo fornecido usando a série de Taylor. O cálculo do cosseno é feito através de um loop que utiliza a série de Taylor para calcular até que a precisão especificada seja atingida.

Dentro do loop, várias funções auxiliares são chamadas para realizar as operações de potências e atualizar variáveis, que são utilizadas no cálculo da série de Taylor. A função (pot) calcula a potência de um número, enquanto as funções (atualiza) e (termo) são usadas para atualizar índices e calcular termos específicos da série, respectivamente.

Por fim, o resultado calculado é impresso na tela, e o programa é encerrado.

Problema 5)

Para esse problema o objetivo é código apresentar um programa que ordena um array de inteiros usando o algoritmo QuickSort. No seu funcionamento temos que, array (arr) é alocado com espaço para 80 bytes, e a string "ORDENADOS " é definida.

No início da função main, o programa chama a syscall 5 para ler um inteiro do usuário e armazená-lo no registrador \$v0. Em seguida, dois registradores \$s0 e \$s1 são inicializados com zero e o valor lido, respectivamente. O registrador \$s2 é carregado com o endereço inicial do array (arr).

O loop L é então executado para preencher o array(arr) com os inteiros fornecidos pelo usuário. Dentro desse loop, o programa lê um novo inteiro, calcula o endereço onde esse inteiro deve ser armazenado no array, e armazena o inteiro no array usando sw.

Após o preenchimento do array, o programa chama a função qsort para ordenar o array. Uma vez que o array é ordenado, o programa imprime os elementos ordenados e termina a execução.

A função qsortH utiliza a função (parter) para determinar um ponto de pivô e rearranjar os elementos do array em torno desse pivô. Após o rearranjo são feitas chamadas recursivas para ordenar as partições da esquerda e direita do array.

A função parter implementa a lógica de particionamento do QuickSort. Ela escolhe um elemento pivô, rearranja os elementos do array de forma que todos os elementos menores que o pivô estejam à esquerda dele e todos os elementos maiores estejam à direita.