APRESENTAÇÃO TRABALHO FINAL - MIPS

MATHEUS PECORARO – **Nº USP** 11917271
SUNGWON YOON – **Nº USP** 9822261

Problema

O problema escolhido pelo nosso grupo é reconhecer se uma string é ou não palíndromo. Palíndromos são palavras ou frases que podem ser lidas da esquerda para direita ou da direita para a esquerda mantendo o mesmo significado. Para frases os espaços não contam logo devem ser ignorados ao percorrer a string.

Solução

```
.text
.globl main
main:
	# Imprime $textoInicial na tela
	la $a0,$textoInicial
	li $v0,4
	syscall

	# Recebe uma string do usuário
	la $a0,$buffer
	li $a1,50
	li $v0,8
	syscall

	# Remove espaços da string
	jal removerEspacos

	# Calcula e guarda o tamanho da string em $a1
	la $t0,($a0)
	li $a1,0
	loopTamanho:
		lb $t2,($t0)
		addiu $t0,$t0,1
		addiu $a1,$a1,1
		bne $t2,'\n',loopTamanho
	subiu $a1,$a1,1		# Retira o byte de newline do contador
```

Inicialmente é utilizada a syscall 4 para imprimir o valor guardado em $a0, que é definido no final do código na seção de data, sendo um texto que pede ao usuário para digitar uma string, que é depois recebida através da syscall 8 em $a0 no espaço reservado por buffer na seção de data, depois disso a função “removerEspacos” é chamada removendo os caracteres de espaço da string. Após o retorno da função o código entra em um loop que passa por todos caracteres da string e a cada execução incrementa o valor de $a1 (que começa em 0) por 1, utilizando o registrador como um contador do tamanho da string, no final é subtraído 1 para descontar o byte newline.

```
ble $a1,2,menorQueDois      # Termina o programa caso a string tenha menos que 3 caracteres

subiu $a1,$a1,1             # Deixa o contador no formato para uso em arrays
jal checarPalindromo        # Chama a função que checa se a string é palíndromo
beq $a2,0,false             # Condição do resultado da função

# Caso a string for palíndromo, retorna um texto correspondente e termina o programa
true:
	la $a0,$textoPalindromo
	li $v0,4
	syscall
	j end
# Caso a string não for palíndromo, retorna um texto correspondente e termina o programa
false:
	la $a0,$textoNaoPalindromo
	li $v0,4
	syscall
	j end



# Caso a string tiver menos que 3 caracteres, retorna um texto correspondente e termina o programa
menorQueDois:
	la $a0,$textoErro
	li $v0,4
	syscall
# Termina o programa
end:
	li $v0,10
	syscall
```

Uma instrução de branch ocorre e caso o contador $a1 for menor que 3 o código pula para uma seção que usa a syscall 4 para imprimir um texto correspondente ao erro (palavras de 2 ou menos letras não podem ser palíndromo) e pula para “end” encerrando o programa com a syscall 10. Após o branch é chamada a função “checarPalindromo” que retorna 0 em $a2 caso a string não for palíndromo e 1 caso for, o código pula para uma seção que usa a syscall 4 para imprimir um texto que diz ao usuário se a string é ou não palíndromo (baseado no valor de $a2) e depois pula para “end”, encerrando o programa.

```
checarPalindromo:
	# Início da string
	la $t0,($a0)
	# Fim da string
	la $t1,($a0)
	addu $t1,$t1,$a1

	# Loop principal para checar se a string é palíndromo
	loopPalindromo:
		beq $t0,$t1,stringPalindromo
		bgt $t0,$t1,stringPalindromo

		lb $t2,($t0)
		lb $t3,($t1)

		bne $t2,$t3,stringNaoPalindromo

		addiu $t0,$t0,1
		subiu $t1,$t1,1

		j loopPalindromo

	# Caso a string for palíndromo, guarda 1 (True) em $a2 e retorna
	stringPalindromo:
		li $a2,1
		jr $ra

	# Caso a string não for palíndromo, guarda 0 (False) em $a2 e retorna
	stringNaoPalindromo:
		li $a2,0
		jr $ra
```

Função que checa se a string é palíndromo, inicia carregando o endereço da string em $t0 e $t1, porém incrementando o endereço de $t1 com $a1 (que guarda o tamanho da string), fazendo com que $t1 aponte para o final, depois entra em um loop que a cada execução incrementa o endereço de $t0 por 1 e o subtraí o de $t1 por 1, assim movendo $t0 para o próximo caractere e $t1 para o caractere anterior, no caso de $t0 e %t1 não representarem o mesmo caractere em qualquer execução o programa pula para “stringNaoPalindromo”, caso isso não ocorra até o momento em que $t0 é maior ou igual a $t1 isso significa que todos os caracteres até o meio da string são iguais e o código pula para “stringPalindromo”, caso não palíndromo guarda 0 em $a2 e retorna, caso contrário guarda 1 em $a2 e retorna.

```
removerEspacos:
	# Guarda o endereço da string em dois registradores, um para per
editar a string
	la $t0,($a0)
	la $t2,($t0)

	loopEspacos:
		# Percorre pelos espaços com $t2 até chegar no próximo ca
		condicaoEspaco:
			lb $t1,($t2)
			bne $t1,32,condicaoEspacoFim
			addiu $t2,$t2,1
			j condicaoEspaco

		condicaoEspacoFim:
		# Guarda o valor de $t2 em $t0 para todo caractere, assim
		lb $t1,($t2)
		sb $t1,($t0)

		addiu $t2,$t2,1
		addiu $t0,$t0,1

		lb $t1,($t2)
		bne $t1,0,loopEspacos

	# Adiciona o caractere null que pode faltar e retorna
	lb $t1,($t2)
	sb $t1,($t0)
	jr $ra
```

Função que remove espaços da string em $a0, inicia carregando o endereço da string em ambos $t0 e $t2, depois entra em um loop que a cada execução entra em outro loop que continua executando enquanto um caractere de espaço estiver sendo apontado por $t2, caso o primeiro caractere ao entrar não seja um espaço o código pula para fora desse loop sem alterar nada, caso contrário vai sendo incrementado 1 em $t2 a cada execução de “condicaoEspaco” assim pulando os caracteres de espaço, ao sair desse loop o caractere após os espaços que agora está sendo apontado por $t2 é guardado na posição de $t0, mesmo que $t2 não tenha pulado nenhum espaço o seu valor ainda sim é escrito em $t0, assim fazendo com que $t0 seja reescrito removendo todos os caracteres de espaço, em toda execução do loop principal é incrementado 1 em $t0 e $t2, quando o caractere apontado por $t2 for NULL o loop principal encerra a função e retorna.

Seção do código utilizada para alocar espaço de memória e declarar variáveis, declara os diversos textos impressos durante a execução do código e um espaço de 50 bytes em $buffer para receber a string do usuário.

```
.data
$textoInicial: .asciiz "Digite a string: "
$textoErro: .asciiz "Uma string deve ter no mínimo 3 caracteres para ser um palíndromo."
$textoPalindromo: .asciiz "A string inserida é palindromo"
$textoNaoPalindromo: .asciiz "A string inserida não é palindromo"
$buffer: .space 50
```