

Lab 11

Manipulação de *strings*

Objetivo: Neste trabalho de laboratório pretende-se treinar a manipulação de *strings* e o uso das convenções de chamada de funções (*gcc*) em Assembly do MIPS.

1. Manipulação de *strings*

Escreva uma função em Assembly do MIPS, chamada `modify_string()`, para modificar um carácter específico de uma string terminada em 0, devolvendo o carácter substituído. O protótipo equivalente da função em C seria:

```
char modify_string(char *str, int idx, char newChar);
```

onde:

- `str` representa o ponteiro para a string original
- `idx` representa o índice da posição onde deve ser alterado o carácter
- `newChar` representa o novo carácter
- a função deve devolver o carácter que foi substituído

No caso de o índice `idx` ultrapassar o final da string, a função deve deixar a string inalterada e devolver o valor 0.

Pode usar como esqueleto do programa o ficheiro `lab11_1.s`. Neste esqueleto está definida a estrutura do programa e das funções a implementar, as funções usuais para entrada e saída usando a **`syscall`**, bem como a definição de um segmento de dados com uma string `str1` terminada em 0.

Teste a função usando a string `str1`, não esquecendo de imprimir na consola a string antes e depois de modificada e o carácter substituído e devolvido pela função `modify_string()`. Escolha qual o índice `idx` e carácter novo `newChar`.

O programa deve respeitar a convenção de chamada de funções usada pelo `gcc`, bem como as convenções de registos e passagem de parâmetros estudadas.

Teste o programa no emulador MARS.

2. Interface com o utilizador

Adapte a solução do ponto anterior para escrever um programa em Assembly do MIPS, para pedir ao utilizador a string e gravá-la na memória (`str`). O programa deve ainda pedir ao utilizador a posição da string que pretende alterar (`idx`) e o novo carácter (`newChar`).

Para garantir a correta utilização da nova string introduzida pelo utilizador, use a string `str2` que está previamente declarada no segmento de dados e para a qual estão reservados 50 bytes.

O programa deve depois chamar a função `modify_string()` e, por fim, apresentar na consola as strings antes e depois de modificadas, bem como o carácter substituído.

Teste o programa no emulador MARS.