```
# EXERCICIO 5 - IMPLEMENTAÇÃO STRCPY

                 .data
                 .align 0
str_welcome:           .asciiz ":: Implementacao da funcao strcpy ::\n>
char *strcpy(char *dest, const char *src) <\n\n"
str_input:       .asciiz " Entre com a string src (fonte) para a funcao:
"
str_result:      .asciiz "\n Resultado da copia: "

                 .align 2
source_sz:       .word 40
destiny_sz:      .word 100

                 .text

                 .globl main
main:
      # 1. Imprimindo as boas vindas :) ---------------------------------
      la    $a0,  str_welcome      # load str_welcome
      li    $v0,  4                # print str code
      syscall                      # system, do it!

      # 2. Alocando espaco para a string fonte e para a string destino --
      lw    $a0,  source_sz        # load source string size
      li    $v0,  9                # alloc memory code
      syscall                      # system, do it!
      move  $s0,  $v0              # $s0 = &str_src

      lw    $a0,  destiny_sz       # load destiny string size
      li    $v0,  9                # alloc memory code
      syscall                      # system, do it!
      move  $s1,  $v0              # $s1 = &str_dest

      # 3. Lendo a string src -------------------------------------------
      la    $a0,  str_input        # load str_input
      li    $v0,  4                # print str code
      syscall                      # system, do it!

      move  $a0,  $s0              # $a0 = &str_src
      lw    $a1,  source_sz        # $a1 = str source size
      li    $v0,  8                # read string code
      syscall

      # 4. Realizando a copia -------------------------------------------
      move  $a0,  $s0              # $a0 = &str_src
      move  $a1,  $s1              # $a1 = &str_dest
      jal   strcpy                 # do strcpy
      move  $s1,  $v0              # $s1 = result string

      # 5. Imprimindo na tela o resultado -------------------------------
      la    $a0,  str_result       # load str_result
      li    $v0,  4                # print str code
      syscall                      # system, do it!

      move  $a0,  $s1              # load result
      li    $v0,  4                # print str code
      syscall                      # system, do it!
      # 6. Encerrando o programa
      li    $v0,  10               # exit code
      syscall                      # system, do it!

#:::::::::::::::::: STRCPY :::::::::::::::::::
strcpy:
      # 1. Ajustando a pilha para a funcao
      add   $sp,  $sp,  -12        # stack build
      sw    $a0,  8($sp)           # save &str_src in the stack
      sw    $a1,  4($sp)           # save &str_dest in the stack
      sw    $ra,  0($sp)           # save return address in the stack

      # 2. Copiando a string src
      lw    $t7,  source_sz        # $t7 = source_sz
      add   $t7,  $t7,  $a0        # $t7 = &str_src + str_src_sz

cpy_loop:
      lb    $t0,  0($a0)           # $t0 = str_src[i]
      sb    $t0,  0($a1)           # str_dest[i] = $t0

      add   $a0,  $a0,  1          # $a0 += 1
      add   $a1,  $a1,  1          # $a1 += 1
      bne   $a0,  $t7,  cpy_loop   # if $a0 != source_sz goto cpy_loop

      lw    $v0,  4($sp)

      # 3. Reajustando a pilha para retornar para a chamada da funcao
      lw    $a0,  8($sp)           # load &str_src in the stack
      lw    $a1,  4($sp)           # load &str_dest in the stack
      lw    $ra,  0($sp)           # load return address in the stack
      add   $sp,  $sp,  -12        # stack free

      jr    $ra                    # return to call_func address
```