```
/**
 * concat - concatenates two strings
 * @param x0: pointer to the first string
 * @param x1: pointer to the second string
 * @return x0: pointer to the concatenated string (allocated using malloc)
 *
 * This function concatenates the second string at the end of the first string
 * and returns the combined string. The function allocates memory for the new
 * concatenated string using malloc,  which needs to be freed by the caller.
 *
 * Registers used: x0,  x1,  x2,  x3,  w3
 * Registers saved: x19-x30,  lr
 */

.text
.global concat
    concat:
        // allocate memory for new string
        stp lr, x0, [sp, #-16]!    // PUSH return address and str0
        str x1, [sp, #-16]!       // PUSH str1
        bl  length           // get length of str0 in x0
        mov x1, x0              // move str0.length to x1
        ldr x0, [sp], #16         // POP str1
        ldp x0, x1, [sp, #-16]!    // PUSH str1 and str0.length to stack
        bl  length           // get length of str1 in x0
        ldp x1, x2, [sp], #16      // POP str1 and str0.length from stack
        add x0, x0, x2            // add string lengths together
        add x0, x0, #1            // add 1 for null terminator
        str x1, [sp, #-16]!       // PUSH str1
        bl  malloc              // allocate memory for new string
        ldr x2, [sp], #16         // POP str1
        ldp lr, x1, [sp], #16      // POP return address and str0
        stp lr, x0, [sp, #-16]!    // PUSH new string for return
    // write str0 to new string
    str0_concat:
        ldrb W3, [x1], #1         // load character of str0 and inc ptr
        cmp  W3, #0              // if current char == null terminator
        b.eq str1_concat        // start writing str1,  we are done with str0
        // else,  if we are not at end of str0
        strb W3, [x0], #1         // store character of str0 to new string and i
nc ptr
        b    str0_concat        // continue loop
    // write str1 to new string
    str1_concat:
        ldrb W3, [x2], #1         // load character of str1 and inc ptr
        cmp  W3, #0              // if current char == null terminator
        b.eq end_concat         // end function,  we have added both
        // else,  if we are not at end of str1
        strb W3, [x0], #1         // store character of str1 to new string and i
nc ptr
        b    str1_concat        // continue loop
    end_concat:
        ldp lr, x0, [sp], #16      // POP new string address
        ret                     // return to calling function
```