

Mar 14, 23 6:59

length.s

Page 1/2

```

/**
 * length - counts the number of characters in a string
 * @param x0: pointer to the string to count
 * @param x1: maximum number of characters to count
 * @return x0: number of characters counted, including null terminator
 *
 * This function counts the number of characters in a string pointed to by x0, u
p
 * to a maximum of x1 characters. The count includes the null terminator. If the
 * string is longer than x1 characters, the function stops counting at x1
 * characters.
 *
 * Registers used: x0, x1, x2, x3, w3
 * Registers saved: x19-x30, lr
 */

.text
.global length
length:
    // Save registers x19 to x30
    //stp x29, x30, [sp, -16]!
    // Setup the stack frame for strlen
    //mov x29, sp

    // Save registers x19 to x29, and the link register lr on the stack
    stp x29, x30, [sp, -16]!
    stp x19, x20, [sp, -16]!
    stp x21, x22, [sp, -16]!
    stp x23, x24, [sp, -16]!
    stp x25, x26, [sp, -16]!
    stp x27, x28, [sp, -16]!
    stp x29, lr, [sp, -16]!

    // Set up the stack frame for this function
    mov x29, sp

    // Move the maximum number of characters to count into x1
    mov x1, x2

    // Initialize the character count to zero
    mov x2, #0
loop:
    // Load a byte from the memory location pointed to by x0 with an offset
of x2 and store it in w3
    ldrb w3, [x0, x2]
    // Check if the byte we just loaded is zero (the null terminator)
    cbz w3, done
    // If the byte is not zero, increment the character count
    add x2, x2, #1
    // Check if we have counted the maximum number of characters
    cmp x2, x1
    // If we have, we are done counting characters, so jump to the end of th
e function
    b.eq done
    // Otherwise, continue counting characters
    b loop

done:
    // Restore the saved registers x19 to x30 from the stack
    //ldp x29, x30, [sp], 16
    // Restore the saved registers x19 to x29, and the link register lr from
the stack

```

Mar 14, 23 6:59

length.s

Page 2/2

```

ldp x29, lr, [sp], 16
ldp x27, x28, [sp], 16
ldp x25, x26, [sp], 16
ldp x23, x24, [sp], 16
ldp x21, x22, [sp], 16
ldp x19, x20, [sp], 16
ldp x29, x30, [sp], 16

// Move the character count into x0 (the return value)
mov x0, x2
ret

```