```
/**
 * indxofstr – Returns the index of string str in a particular String.
 *
 * This function searches for the first occurrence of the specified string str i
n the target
 * string and returns the index of the first occurrence of the string str in the
 target string.
 * The search starts from the beginning of the target string. If the string str
is not found
 * in the target string, the function returns –1.
 *
 * @param x0: Address of the target string to search.
 * @param x1: String to find in the target string.
 *
 * Registers used: x0, x1, x2, x3, x4
 * Registers saved: x0, x1
 */
.text
.global indxofstr
    indxofstr:
        str x0, [sp, #-16]!     // Save the address of the target string on the s
tack
        ldrb w3, [x1], #1       // Load the first character of the string to sear
ch and move the pointer

    loop:
        ldrb w2, [x0], #1       // Load the next character of the target string a
nd move the pointer
        cmp w2, #0              // Check if the end of the target string has been
 reached
        b.eq cherror           // If so, the character to search was not found i
n the target string, so return –1
        cmp w2, w3             // Compare the current character of the target st
ring with the first character of the string to search
        b.eq chfound           // If they match, start searching for the rest of
 the string to search

        b loop                 // If they don't match, keep searching for the ch
aracter to search in the target string

    chfound:
        stp x0, x1, [sp, #-16]! // Save the current target string and string to
search pointers on the stack

    sloop:
        ldrb w2, [x0], #1       // Load the next character of the target string a
nd move the pointer
        ldrb w4, [x1], #1       // Load the next character of the string to searc
h and move the pointer
        cmp w4, #0              // Check if the end of the string to search has b
een reached
        b.eq strfound          // If so, the entire string to search was found i
n the target string, so return the index
        cmp w2, #0              // Check if the end of the target string has been
 reached
        b.eq strerror          // If so, the string to search was not found in t
he target string, so return –1
        cmp w2, w4             // Compare the current character of the target st
ring with the current character of the string to search
        b.eq sloop             // If they match, continue searching for the rest
 of the string to search
```

```
        ldp x0, x1, [sp], #16  // If they don't match, pop the saved target stri
ng and string to search pointers off the stack
        b loop                 // and continue searching for the character to se
arch in the target string

    strfound:
        // Pop the stack frame to restore the saved registers
        ldp x0, x1, [sp], #16
        // Calculate the index of the found string in the target string
        ldr x1, [sp], #16
        sub x0, x0, x1
        sub x0, x0, #1
        // Return the index
        ret

    cherror:
        // No string match found, so pop the stack frame to restore the saved re
gisters
        add sp, sp, #16
        // Return –1
        mov x0, #-1
        ret

    strerror:
        // No string match found, so pop the stack frame to restore the saved re
gisters
        add sp, sp, #16
        // Return –1
        mov x0, #-1
        ret
```