Str_copy Procedure

The Str_copy procedure copies a null-terminated string from a source location to a target location.

It takes two arguments: pointers to the source and target strings, respectively.

The Str_copy procedure works by first calling the Str_length procedure to get the length of the source string.

The Str_copy procedure then copies the source string to the target string byte by byte using the REP MOVSB instruction.

The following is the MASM code for the Str_copy procedure:

```
;-----------------------------------------------------
; Str_copy Procedure
; Copies a string from the source to the target.
; Requires: the target string must contain enough space
; to hold a copy of the source string.
;-----------------------------------------------------
Str_copy PROC USES eax ecx esi edi,
    source:PTR BYTE,   ; source string
    target:PTR BYTE    ; target string

    INVOKE Str_length, source   ; Calculate the length of the source string and store it in EAX.
    mov ecx, eax                ; Copy the length into ECX for REP count.
    inc ecx                     ; Add 1 for the null byte at the end of the string.

    mov esi, source             ; Initialize esi with the source pointer.
    mov edi, target             ; Initialize edi with the target pointer.

    cld                         ; Set the direction flag to forward.

    rep movsb                   ; Use REP to copy the string byte by byte from source to target.

    ret                         ; Return when the string is copied.

Str_copy ENDP
```

This code defines the Str_copy procedure, which copies a null-terminated string from the source to the target.

It uses the Str_length procedure to calculate the length of the source string, sets up pointers to both source and target strings, and then uses the rep movsb instruction to copy the string byte by byte.

The procedure returns once the entire string is copied to the target.

To use the Str_copy procedure, you would pass the addresses of the source and target strings to the procedure as arguments.

For example, to copy the string "Hello, world!" from one location to another, you would use the following code:

```
350 mov eax, OFFSET "Hello, world!"
351 mov ebx, OFFSET target_string
352 call Str_copy
353
354 ; The target_string variable will now contain a copy of the string "Hello, world!"
```