# Stack Frame

A stack frame is an extension of the stack data structure to organize function calls, parameters, and local variables.

Following is not tested so be cautious.

```
sub sp, 2           ; Make return space for the return value
push word 10        ; Arg2, Arguments are pushed in reverse order
push word 3         ; Arg1
call my_function    ; my_function(3,10)
add sp, 6           ; Cleanup arguments
jmp $

;======================= My Function =====================
; Adds two parameters together. Preserves all registers.
; Input:
;   [bp + 6] = Parameter 1
;   [bp + 8] = Parameter 2
; Output:
;   [bp + 10] = Return value
SECTION .text
my_function:
    push bp             ; Save the old base pointer
    mov bp, sp          ; Set bp to the current stack frame address

    sub sp, 4           ; Allocate space for any local variables. (Unused)

    push ax             ; It is best to preserve any registers that
                        ; are used throughout the function unless
                        ; the registers are defined to be changed

    mov ax, [bp + 6]    ; Move Parameter 1 into ax
    add ax, [bp + 8]    ; Add Parameter 2 to ax

    mov [bp + 10], ax   ; Set return values

    pop ax              ; Restore ax to its previous state

    leave               ; This removes the stack frame and restores
                        ; sp and bp
    ret
```