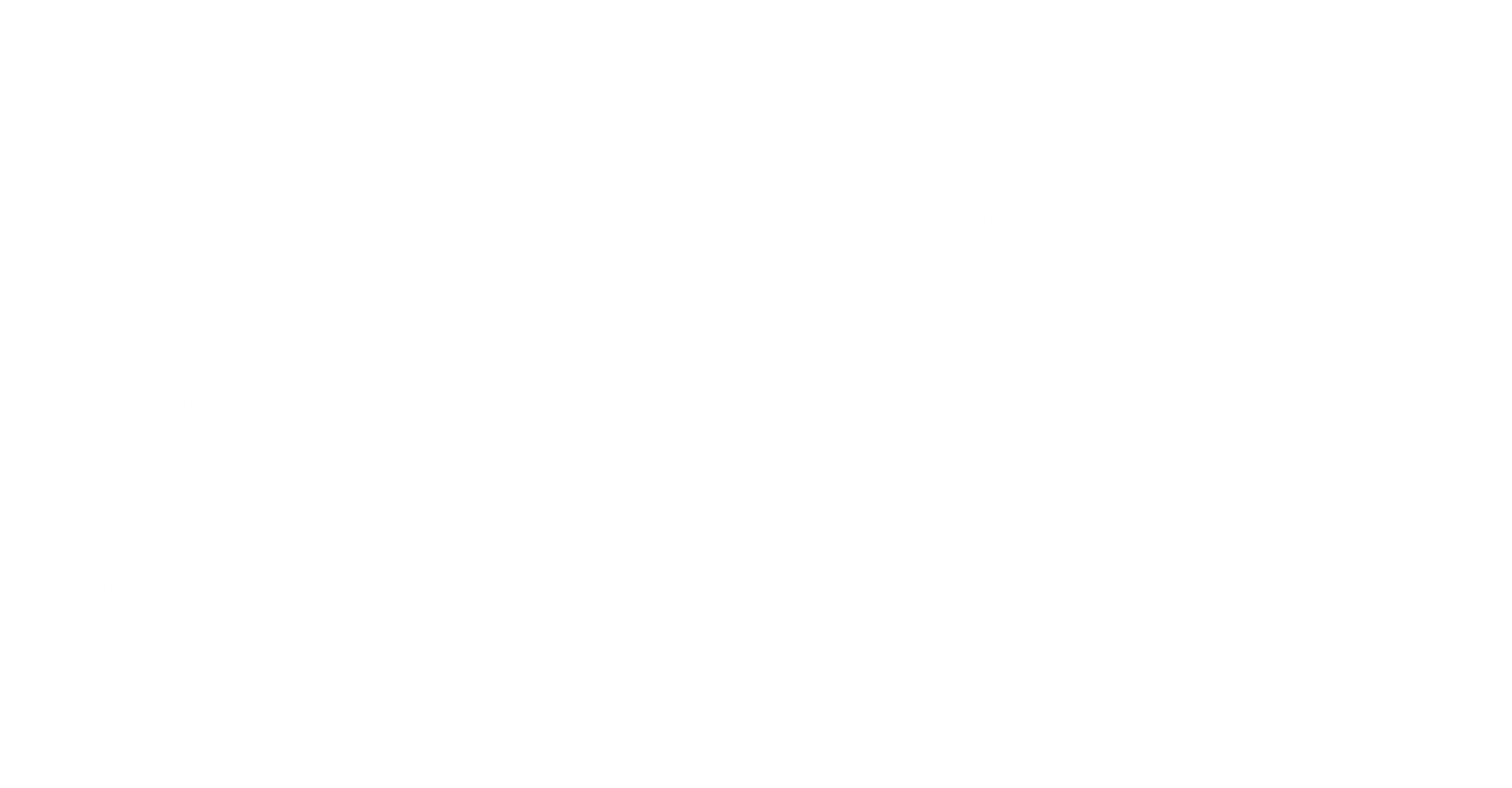
**Procedures**

**Procedures** are like functions that are used in higher level languages. However, we cannot pass **arguments** to procedures like we can with functions. It is essentially just a **separate piece** of code. We must **call** it and when it is finished, we must **return** to the previous point from where the procedure was called.



The CALL instruction causes:

1. The value of the **IP register** to be **incremented** to the address of the next instruction and **stored** in the stack.
2. The address of **first instruction** from the procedure to be set as the value of the **IP register**.
3. **Execution** to be restarted in the procedure.

The RET instruction causes:

1. The old value of the **IP register** to be restored from the stack.
2. **Execution** to be restarted at the point where the procedure was called.

Example:

ORG 0100H  
  
.DATA  
StrArray DB 'Hello World!!$' *; define string to display*.CODE  
MAIN PROC  
 MOV AX, @DATA  
 MOV DS,AX  
 LEA DX, StrArray *; DX = StrArray[0]* CALL USER *; call procedure* MOV AH, 4CH   
 MOV AL, 00H *; a code after procedure call and return* INT 21H *; exit to DOS*MAIN ENDP  
  
USER PROC *; declare a procedure named USER* MOV AH, 09H   
 INT 21H   
 RET *; return to MAIN procedure*USER ENDP *; end of procedure USER*END MAIN *; end of program*

ASSEMBLY