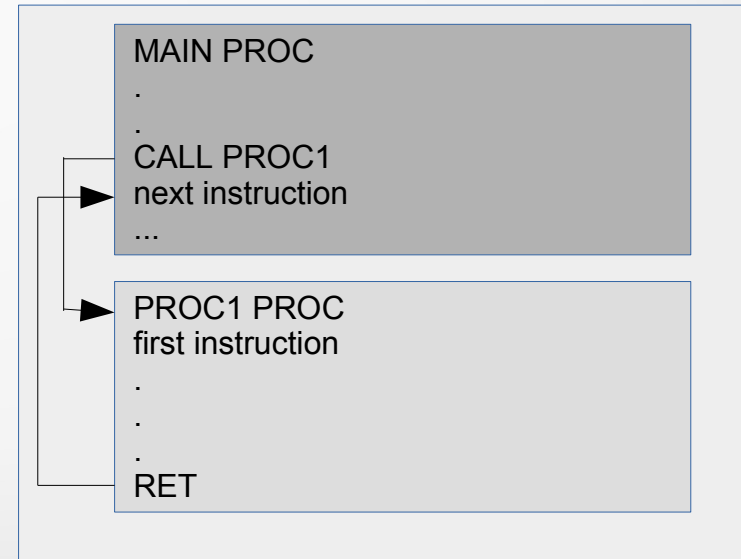# Procedure

Covering topics

- Introduction
- Procedure Terminology
- CALL and RET
- An example of procedure

Introduction

- Decomposing original problem into sub problems.

- High level languages uses procedure to employ solution to these sub problems.

- Assembly program can be structured as a collection of procedure.

Continue....

- Main procedure is the entry point for every program.

- To carry out a task, main calls another procedure.

- When one procedure calls another, control transfer to the called procedure; called procedure usually returns control to the caller at the next instruction after the call statement.

```
MAIN PROC
.
.
CALL PROC1
next instruction
...


PROC1 PROC
first instruction
.
.
.
RET
```

Terminology of procedure

- Procedure Declaration: Syntax for procedure declaration is done as follow:

  name   PROC   type

  ;body of the procedure

        RET

  name   ENDP

- Name is user defined name for procedure.

- Type is optional operand can be NEAR(stmt that calls procedure are in same segment) or FAR(stmt calling procedure are in different segments).

- RET causing control to transfer back to calling procedure.

CALL and RET

- CALL: to invoke a procedure the CALL instruction is used.
    - Direct call
        - CALL name
    - Indirect call
        - CALL address_expression ; address_expression specifying memory location or register containing address of procedure.
- Executing CALL causes following to happen:
    - Return address to the calling program is saved on the stack. It is often offset of the next instruction after the CALL. CS:IP
    - IP gets the offset address of the first instruction of the procedure.
- To return from a procedure RET instruction is used.
    - RET pop_value; pop_value is optional
    - RET causes the stack to be popped into IP.

An example of procedure

```
.MODEL  SMALL
.STACK  100h
.DATA
    MSG  DB "Enter a sequence of character press enter to end reading", 0DH,0AH,  "$"
.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX

LEA AX,MSG

CALL STROUT
CALL CHARIN
CALL CHAROUT


MAIN ENDP
END MAIN
```

```
CHARIN PROC
MOV AH,1
INT 21h
RET
DHARIN ENDP

CHAROUT PROC
MOV AH,2
MOV DL,AL
INT 21h
RET
DHAROUT ENDP

STROUT PROC
MOV AH,9
LEA DX,AX
RET
STROUT ENDP
```

References

- Assembly Language Programming and Organization of the IBM PC (Ytha Yu, Charles Marut)