



# HÀM - MACRO

---

## ■ **Procedures**

name PROC

; here goes the code  
; of the procedure ...

RET

name ENDP

## ■ **CALL** proc



# HỢP NGỮ

---

- **Stack - LIFO** (Last In First Out) algorithm

**PUSH** - stores 16 bit value in the stack.

**POP** - gets 16 bit value from the stack.

- **SS:SP :top of the stack.**



# HỢP NGŨ

---

- **PUSH REG**
- PUSH SREG**
- PUSH memory**
- PUSH immediate ; 80186**

**REG: AX, BX, CX, DX, DI, SI, BP, SP.**

**SREG: DS, ES, SS, CS.**

**memory: [BX], [BX+SI+7], 16 bit variable, etc...**

**immediate: 5, -24, 3Fh, 10001101b, etc...**



# HỢP NGŨ

---

- **POP REG**  
**POP SREG**  
**POP memory**

**REG: AX, BX, CX, DX, DI, SI, BP, SP.**

**SREG: DS, ES, SS, (except CS).**

**memory: [BX], [BX+SI+7], 16 bit variable, ...**



# HỢP NGỮ

---

- **"PUSH *source*" :**
  - **Subtract 2 from SP register.**
  - **Write the value of *source* to the address SS:SP.**



# HỢP NGỮ

---

- **"POP *destination*" :**
  - **Write the value at the address SS:SP to *destination*.**
  - **Add 2 to SP register.**



# HỢP NGỮ

---

- **Các lệnh về chuỗi: (lệnh + tiền tố lặp)**

## Các lệnh:

- **SCASB: Scan string byte:** tìm kiếm ký tự trên chuỗi.
- **SCASW: Scan string word:** tìm kiếm 1 word trên chuỗi.
- **MOVSB: Move String Byte:** Sao chép chuỗi theo từng byte
- **MOVSW: Move String WORD:** Sao chép chuỗi theo từng word



# HỢP NGỮ

---

- **CMPSB: Compre String byte: so sánh chuỗi theo từng byte**
- **CMPSW: Compre String word: so sánh chuỗi theo từng word**





# HỢP NGỮ

---

- **LODSB: Load String Byte: 1 Lấy ký tự từ chuỗi**
- **LODSW: Load String Word: 1 Lấy word từ chuỗi**
- **STOSB: Store String Byte: Ghi ký tự vào chuỗi**
- **STOSW: Store String Word: Ghi 1word vào chuỗi**



# HỢP NGỮ

---

- **Tiền tố lặp: (giống lệnh lặp)**
  - **REP : repeat lặp lại**
  - **REPE/REPZ: repeat equal lặp lại nếu bằng**
  - **REPNE/REPNZ: repeat not equal lặp lại nếu khác**



# HỢP NGỮ

---

- **Các bước thực hiện lệnh lặp trên chuỗi:**
  - **Thiết lập địa chỉ nguồn: DS:SI**
  - **Thiết lập địa đích: ES:DI**
  - **Thiết lập cờ hướng: DF = 0 (tăng địa chỉ) CLD, DF = 1 (giảm địa chỉ) STD**
  - **Số lần lặp: CX**
  - **Lựa chọn tiên tố lặp.**

# HỢP NGỮ

---

- **Tìm chiều dài chuỗi: SCASB**
  - **Chuỗi kết thúc: 0**
  - **Ký tự cần tìm: AL**
  - **Thiết lập địa đích: ES:DI**
  - **Thiết lập cờ hướng: DF = 0 (tăng địa chỉ) CLD.**
  - **Số lần lặp: CX= 65535**
  - **Lựa chọn tiên tố lặp: REPNZ**

# HỢP NGỮ

---

- Sao chép chuỗi: **MOVSB**
  - Thiết lập chuỗi nguồn : **DS:SI**
  - Thiết lập chuỗi đích: **ES:DI**
  - Thiết lập cờ hướng: **DF = 0** (tăng địa chỉ)  
**CLD.**
  - Số lần lặp: **CX = chiều dài chuỗi + 1**
  - Lựa chọn tiền tố lặp: **REP**

# HỢP NGỮ

---

- Sao chép chuỗi, biến thành chữ thường: LODSB, STOSB
  - Thiếp lập chuỗi nguồn : DS:SI
  - Thiết lập chuỗi đích: ES:DI
  - Thiết lập cờ hướng: DF = 0 (tăng địa chỉ) CLD.
  - Số lần lặp: CX= chiều dài chuỗi +1
  - Lệnh lặp: LOOP

# HỢP NGỮ

---

- **So sánh 2 chuỗi: CMPSB**
    - **Thiết lập chuỗi nguồn : DS:SI**
    - **Thiết lập chuỗi đích: ES:DI**
    - **Thiết lập cờ hướng: DF = 0 (tăng địa chỉ) CLD.**
    - **Số lần lặp: CX = chiều dài chuỗi + 1**
    - **Tiền tố lặp: REPE**
- => CX = 0, 2 chuỗi giống nhau, ngược lại CX != 0, khác nhau.**



# HỢP NGỮ

---

- **Macros** (Lệnh đại )

TênMacro MACRO [Các tham số]

LOCAL Các nhãn cục bộ  
các lệnh

TênMacro ENDM





# HỢP NGỮ

---

## ■ **Macros**

name MACRO [parameters,...]

LOCAL LABEL

<instructions>

name ENDM



# HỢP NGŨ

---

```
MyMacro MACRO p1, p2, p3
```

```
    MOV AX, p1
```

```
    MOV BX, p2
```

```
    MOV CX, p3
```

```
MyMacro ENDM
```