



# LẬP TRÌNH HỢP NGỮ: NGĂN XẾP & THỦ TỤC

---

TS. Trần Ngô Như Khánh

# Ngăn xếp

- Vùng nhớ lưu trữ tạm thời một số dữ liệu dùng cho chương trình hoặc địa chỉ của các chương trình con.



**Stack of books**

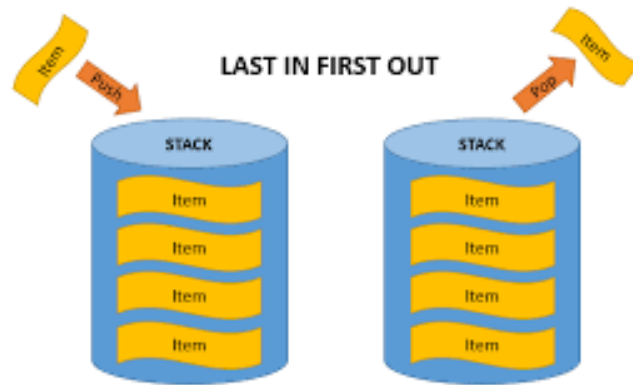


**Stack of Coins**



# Ngăn xếp

- Truy cập theo cơ chế LIFO (Last In First Out)
- Mỗi phần tử trong ngăn xếp là từ (word)
- Địa chỉ ngăn xếp được xác định bởi cặp giá trị thanh ghi SS:SP
  - SS chứa địa chỉ đoạn
  - SP chứa địa chỉ ô



# Hoạt động của ngăn xếp

- Với khai báo

**.STACK 100h**

- $SP = 100h$
- Khi chưa sử dụng, ngăn xếp rỗng, vị trí xác định bởi SP là đỉnh ngăn xếp
- Khi đưa dữ liệu vào ngăn xếp, SP sẽ giảm bớt 2, cập nhật định mới ngăn xếp
- Khi lấy dữ liệu ra khỏi ngăn xếp, SP sẽ tăng lên 2

# Hoạt động của ngăn xếp

Ngăn xếp khi chưa sử dụng

		000h
		...
		0FCh
		0FEh
		100h

SS:SP →

Đưa giá trị 1234h vào

SS:SP →

		000h
		...
		0FCh
12	34	0FEh
		100h

# Hoạt động của ngăn xếp

Đưa tiếp giá trị 5678h vào

SS:SP →			000h
			...
	56	78	0FCh
	12	34	0FEh
			100h

Lấy giá trị ở đỉnh ra

SS:SP →			000h
			...
			0FCh
	12	34	0FEh
			100h

# Hoạt động của ngăn xếp

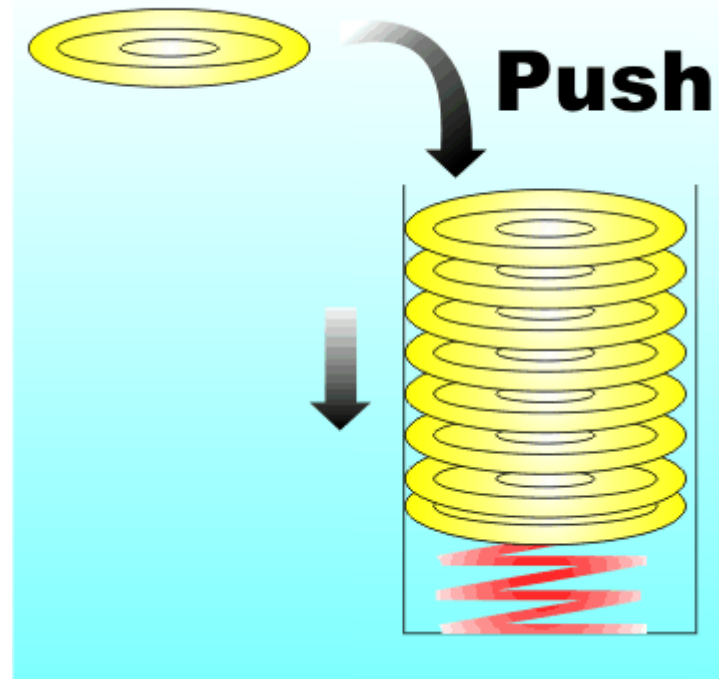
- Đưa trị vào ngăn xếp

## **PUSH Nguồn**

- Ví dụ

PUSH AX

PUSH 1000h



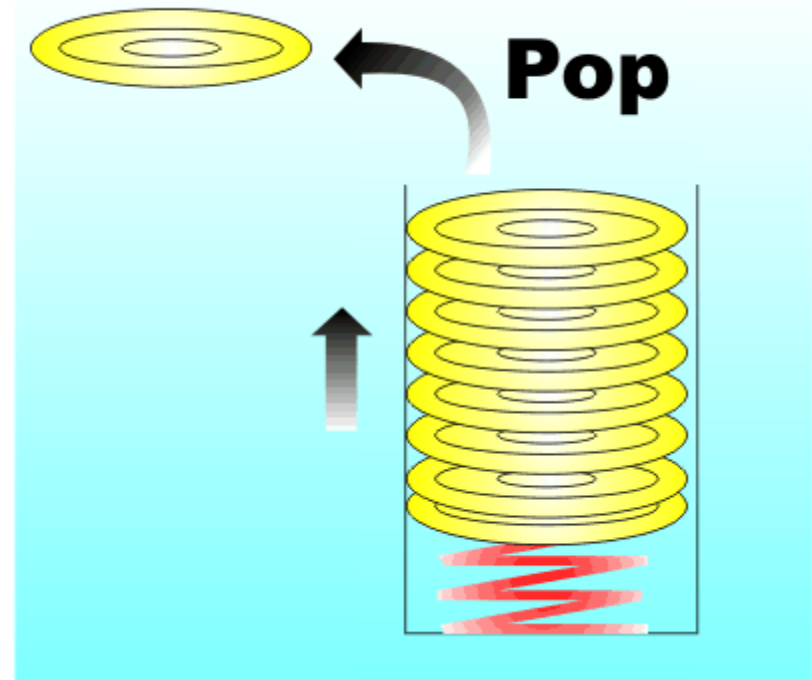
# Hoạt động của ngăn xếp

- Lấy trị ra khỏi ngăn xếp

**POP Đích**

- Ví dụ

POP AX





# Bài tập ví dụ

- Nhập một chuỗi ký tự, hiển thị theo thứ tự ngược lại
- Giải thuật:

```
CX=0 (Đếm số ký tự đã nhập)
Repeat
    Nhập một ký tự
    If Ký tự <> Enter
        Cất ký tự vào ngăn xếp
        Tăng CX
    EndIf
Until Ký tự = Enter
For CX
    Lấy ký tự ra khỏi ngăn xếp
    Hiển thị ký tự
End for
```

# Câu hỏi thảo luận

- Với AX=1234h, BX=5678h, CX=9ABCh và SP=1000h. Cho biết nội dung AX, BX, CX và SP sau khi thực hiện các lệnh sau:

PUSH AX

PUSH BX

XCHG AX, CX

POP CX

PUSH AX

POP BX

# Thủ tục

- Thủ tục là một đoạn chương trình được viết riêng, giúp chương trình dễ đọc và linh hoạt
- Khai báo thủ tục:

```
Tên_Thủ_Tục PROC  
                ; Các lệnh trong thủ tục  
                RET  
Tên_Thủ_Tục ENDP
```

# Gọi và kết thúc thủ tục

- Gọi thủ tục:

CALL Tên\_Thủ\_Tục

CALL Địa chỉ

- Khi thực hiện lệnh gọi thủ:

- Địa chỉ (ô) của lệnh kế lệnh CALL được cất vào ngăn xếp
- Địa chỉ (ô) thủ tục được đưa vào thanh ghi IP
- Lệnh được thi hành kế tiếp sau lệnh CALL chính là lệnh đầu tiên của thủ tục

- Khi thực hiện lệnh RET để quy về trình gọi thủ tục:

- Địa chỉ trong ngăn xếp được lấy ra và đặt vào IP
- Lệnh được thi hành tiếp theo sau lệnh RET là lệnh kế lệnh CALL trong trình gọi

# Ví dụ

```
.MODEL SMALL
.STACK 100h
.CODE
    MOV AL, 1
    MOV BL, 2
    CALL m2
    CALL m2
    CALL m2
    MOV AH, 4Ch
    INT 21h
;-----
m2 PROC
    MUL BL
    RET
m2 ENDP
;-----
END
```

# Lệnh luận lý

- NOT Đích

- Đổi toán hạng đích thành số bù. Không cờ nào bị ảnh hưởng
- Ví dụ:

```
MOV AL, 3Eh ;AL=00111110= 3Eh
```

```
NOT AL      ;AL=11000001= C1h
```

- AND Đích, Nguồn

- Thực hiện phép AND giữa các toán hạng, kết quả cuối cùng chưa trong Đích
- Ví dụ:

```
AND AL, 01111111B ;xóa bit dấu của AL
```

```
MOV AL, '5'        ;đổi mã ASCII thành số
```

```
AND AL, 0Fh        ;tương ứng
```

# Lệnh luận lý

- **OR Đích, Nguồn**

- Thực hiện phép OR
- Ví dụ:

```
OR AL, 10000001b ; bật 1 MSB và LSB của AL
```

```
MOV AL, 5 ; đổi số thành mã ASCII
```

```
OR AL, 30h ; tương ứng
```

- **XOR Đích, Nguồn**

- Thực hiện phép XOR
- Ví dụ:

```
MOV AL, 01010101b ; AL=01010101b
```

```
XOR AL, 11110000b ; AL=10100101b
```

```
XOR AL, AL ; AL=00000000b
```

# Lệnh luận lý

- TEST Đích, Nguồn

- Giống như lệnh AND nhưng không thay đổi giá trị các toán hạng, chỉ cập nhật kết quả các cờ
- Thường dùng để kiểm tra bit
- Ví dụ:

TEST DX, 2000h

JZ BitIs0;

BitIs1:                   ;Bit 13 là 1

BitIs0:                   ;Bit 13 là 0



# Lệnh thao tác Bit

- Dịch trái: SHL Đích, Số đếm
- Dịch phải: SHR Đích, Số đếm
- Quay trái: ROL Đích, Số đếm
- Quay phải: ROR Đích, Số đếm

# Bài tập

## 1) Nhập số nhị phân vào BX

1. Xóa BX
2. Nhập một ký tự
3. Nếu ký tự là Enter nhảy đến 8
4. Đổi thành số tương ứng (Ký tự là '0'/'1')
5. Dịch trái BX
6. Đưa trị đã đổi vào bit LSB của BX
7. Về 2
8. Kết thúc

## 2) Xuất số nhị phân trong BX

Lặp 16 lần:

Quay trái BX để lấy bit cực trái (Vào CF)

Nếu CF=1, Xuất '1'

Nếu CF=0, Xuất '0'

# Bài tập

## 1) Nhập số thập lục phân và chứa vào BX

1. Xóa BX
2. Nhập một ký tự
3. Nếu ký tự là Enter về 8
4. Đổi thành số Hex tương ứng
5. Dịch trái BX 4 bit
6. Đưa trị đã đổi và 4 bit thấp của BX
7. Về 2
8. Kết thúc

## 2) Xuất số thập lục phân trong BX

Lặp 4 lần;

Chép BH vào DL

Dịch phải DL 4 bit

Nếu  $DL < 10$ , đổi thành ký số 0..9 tương ứng

Nếu không, đổi thành chữ A..F tương ứng

Xuất ký tự trong DL

Quay trái BX 4 bit