

# **C *u trúc d* *li u* và *thu t* toán**

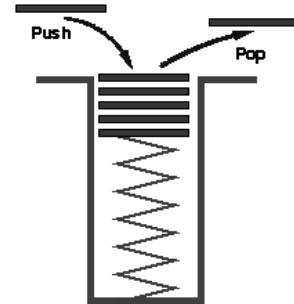
## **Stacks**



## Stacks

- Stacks là mảng đặc biệt của danh sách (mảng có chỉ mục) và có thể truy cập theo cách LIFO
- Hai phép toán
  - int push( Stack s, void \*item );  
- Bổ sung một phần tử vào cuối của stack
  - void \*pop( Stack s );  
- Loại bỏ một phần tử cuối cùng của stack
- Tính năng của stack máy tính
- Các phép toán khác

```
int IsEmpty( Stack s );
/* Return TRUE if empty */
void *Top( Stack s );
/* Return the item at the top,
without deleting it */
```



## **Stacks – Thẻ c**

- **Mảng (Arrays)**
  - Cung cấp một stack và có cách gọi ngắn
  - Hỗn hợp và mở rộng không áp dụng cho chỉ số
  - Giữ nguyên cáchабим tài nguyên  
bùc
    - Bộ nhớ trong máy tính cần
    - Kích thước máy xem, etc
- **Phép toán push, pop**
  - Các biến của AddToC..., DeleteFromC...
- **Danh sách liên kết có thể**
- **Stack:**
  - *Vì có bốn là một danh sách liên kết vàинг nghĩa là bị!*

## **Stacks – m t s v n liên quan**

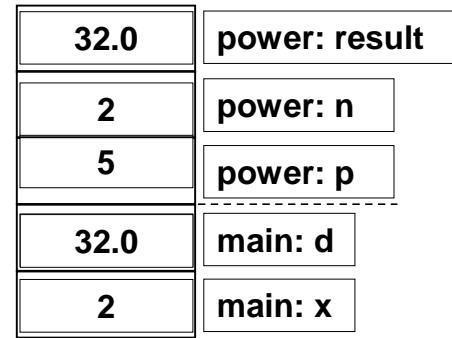
- **Stacks trong ch ng trình tin h c**
  - Là khóa call / return trong functions & procedures
  - Khuôn d ng Stack cho phép các l i g i qui
  - Call: push stack frame
  - Return: pop stack frame
- **Khuôn d ng Stack**
  - Các tham s c a Function
  - Return a ch
  - Các bi n c c b (Local variables)

## **Ví d : Ph ng th c trao id li u**

- C dùng 1 stack l u tr các bi n c c b và các chuy n các tham s cho hàm v i m i l n g i hàm th c hi n
- ①Hàm g i (O) c t các tham s vào stack.**
  - ②G i th c hi n hàm c g i (F).**
  - ③F nh n l y các tham s t stack**
  - ④F t o các bi n c c b ng v i các tham s trên stack**
  - ⑤Khi k t thúc, F c p nh t giá tr các tham s (ref) và tr i u khi n cho O**
  - ⑥O nh n l y các giá tr m i c a tham s c ng nh giá tr tr v**

## *Phong cách trao đổi liên*

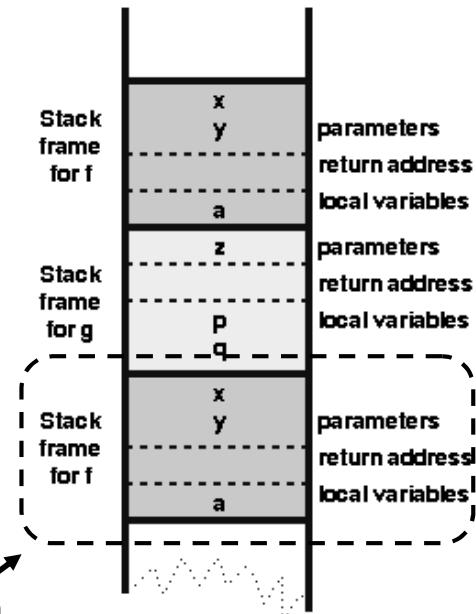
```
#include <stdio.h>
double power(int, int);
int main(void)
{
    int x = 2;
    double d;
    d = power(x, 5);
    printf("%lf\n", d);
    return 0;
}
double power(int n, int p)
{
    double result = n;
    while(--p > 0)
        result *= n;
    return result;
}
```



## ***Stack Frames - Functions in HLL***

- **Program**

```
function f( int x, int y) {  
    int a;  
    if ( term_cond ) return ...;  
    a = ....;  
    return g( a );  
}  
  
function g( int z ) {  
    int p, q;  
    p = .... ; q = .... ;  
    return f(p,q);  
}
```



**Context  
for execution of *f***

## *qui*

- Là công nghệ thời ưu áich
  - nh nghĩa các hàm toán học
  - nh nghĩa cấu trúc dữ liệu
  - Các cấu trúc qui có sẵn lý t nhiên b i các hàm qui!

## *qui*

- Là công nghệ thuật u ách
  - nh nghĩa các hàm toán học
  - nh nghĩa cấu trúc dữ liệu
  - Các cấu trúc qui có sẵn lý t nhiên b i các hàm qui!
- Các hàm nh nghĩa qui
  - factorial
  - Fibonacci
  - GCD bài thu t toán Euclid
  - Biến i Fourier
  - Trò chơi
    - Tháp Hanoi (Towers of Hanoi)
    - Cờ (Chess)

## *qui – Ví d*

- Dãy s (Fibonacci)

*Mã gi  
(Pseudo-code)*

```
fib( n ) = if ( n = 0 ) then 1
            else if ( n = 1 ) then 1
            else fib(n-1) + fib(n-2)
```

C

```
int fib( n ) {
    if ( n < 2 ) return 1;
    else return fib(n-1) + fib(n-2);
}
```

*Gi i pháp n gi n, d hi u  
(Simple, elegant solution!)*

## *qui – Ví d*

- Dãy s Fibonacci

```
C int fib( n ) {  
    if ( n <= 1 ) return n;  
    return fib(n-1) + fib(n-2);  
}
```

Nh ng, trong Fibonacci, th i gian ch y t i!!!!

Tuy nhiên, nh i u hàm qui khác,  
VD tìm ki m nh phân, r t n gi n, rõ ràng và hi u qu !