CHƯƠNG 8

TỔ CHỨC BẢNG DANH BIỂU

8.1. Giới thiệu

Có bốn phương pháp truy xuất trên bảng danh biểu:

1. Tìm kiếm tuyến tính (linear search)
2. Tìm kiếm nhị phân (binary search)
3. Tìm kiếm trên cây (tree search)
4. Mã hóa băm (hash coding)

8.2. Các tác vụ trên bảng danh biểu

Bảng 8.1. *Các tác vụ trên bảng danh biểu*

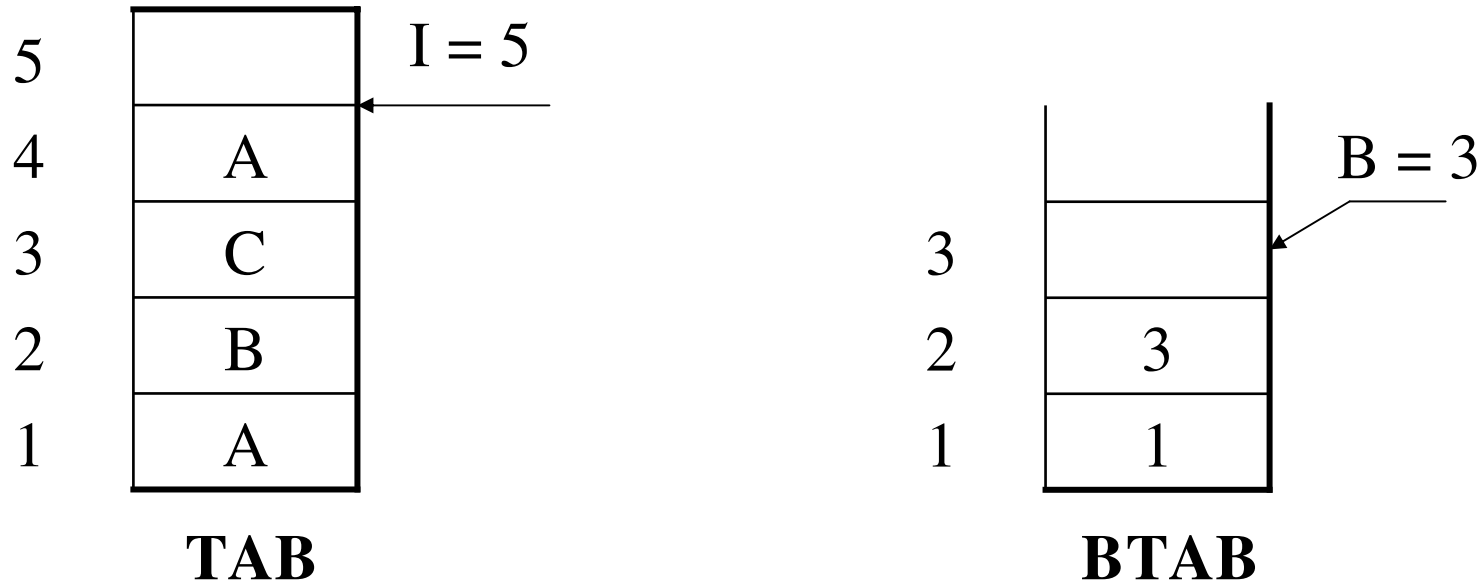
Tên chương trình con	Cách gọi	Hành vi thực thi
Enter	Enter (id)	Khi gặp một danh biểu mới được khai báo, thủ tục này sẽ kiểm tra xem danh biểu mới đó có trùng với tên nào trong cùng một tầm vực? Nếu không, thủ tục enter sẽ đưa danh biểu mới vào bảng danh biểu. Ngược lại enter sẽ thông báo lỗi về việc khai báo một danh biểu nhiều lần trong cùng một tầm vực.
loc (hàm)	n := loc (id)	Khi cần truy xuất một danh biểu, loc sẽ tìm trên bảng danh biểu từ phân tử mới nhất của tầm vực mới nhất đến phân tử cũ nhất của tầm vực cũ nhất để tìm vị trí của id và trả về thông qua tên loc của hàm.
Scopeentry	Scopeentry	Khi trình biên dịch đi vào một tầm vực mới, scopeentry sẽ đánh dấu trên Stack (bảng danh biểu) một tầm vực mới.

Scopeexit	Scopeexit	Khi trình biên dịch đi hết một tầm vực scopeenxit sẽ thải hồi những tên biến không còn có ý nghĩa và tái lập một tầm vực ngoài cùng gần nhất.

8.3. Bảng danh biểu tuyến tính (linear symbol table)

Thí dụ 8.1. Cho đoạn chương trình trong ngôn ngữ Algol.

```
begin real A, B;
    begin real C, A;
    . . . . . .
end;
end;
```



Hình 8.1. Bảng danh biểu tuyến tính của thí dụ 8.1

Các tác vụ trên bảng danh biểu tuyến tính được trình bày như sau:

Giải thuật:

const tab lim = ...;

btablim = ...;

```
type  tabinden = 1 .. tablim;
        item = record
                key: alfa; /* alfa là kiểu chuỗi các ký tự */
                end;
var btab: array [1 .. btablim] of integer;
                tab: array [1 .. Tablim] of item;
                b: 1 •• tablim;
                t: tabindex;
procedure enter (id: alfa)
                var sb: tabindex;
                        begin sb := btab [b –1];
```

Tìm kiếm trên bảng TAB từ vị trí sb đến vị trí t – 1, xem có phần tử nào mang key bằng id không? Nếu có, thủ tục error sẽ thông báo lỗi 1 là lỗi có hai danh biểu cùng tên trong cùng tầm vực. Ngược lại, **if** t = tablim **then** error (12)

```
                    else begin      tab [t] key := id;
                                    t := t + 1
                                end;
            end;
    function loc (id: alfa): tabindex;
            begin
```

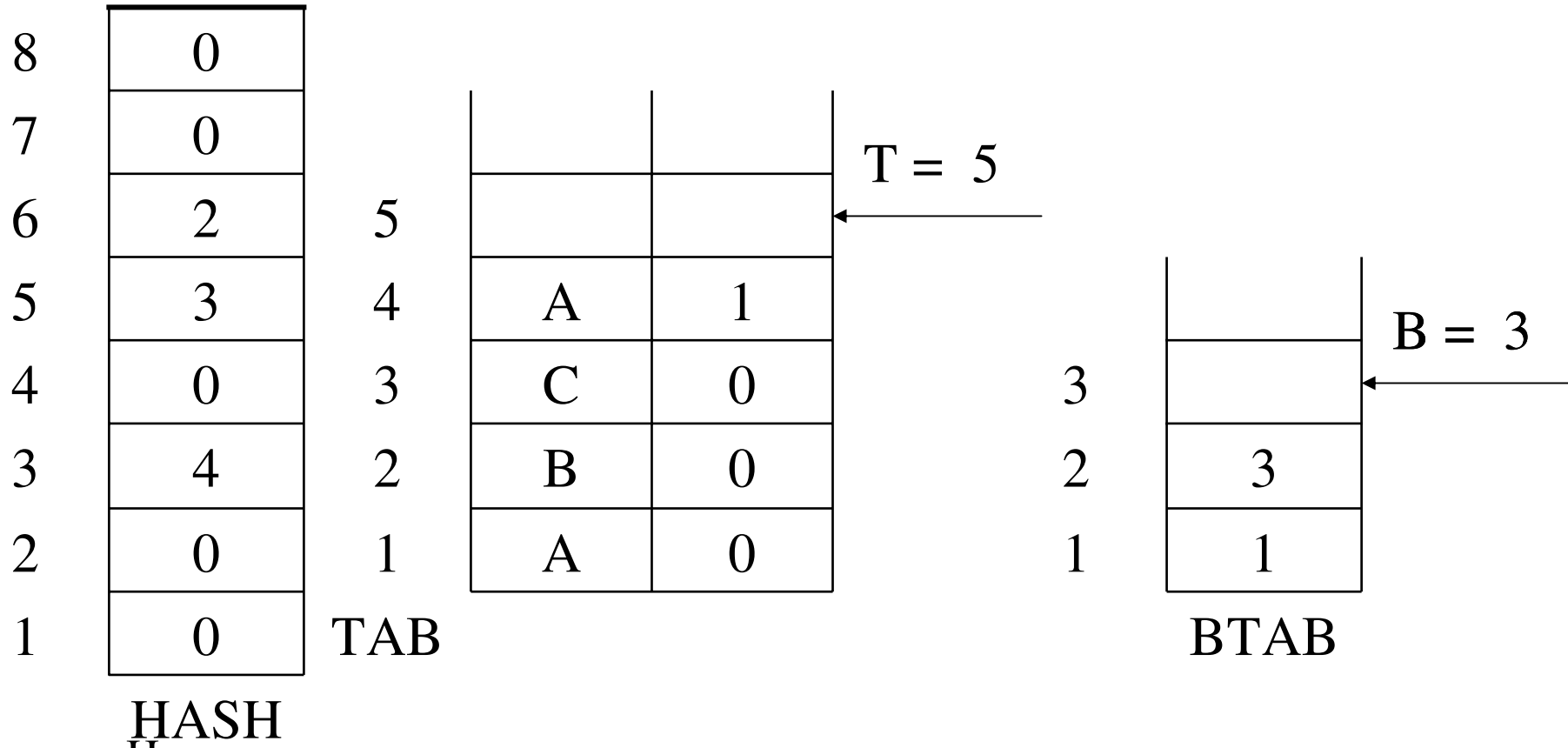
Tìm kiếm từ vị trí đầu TAB đến vị trí t –1, xem có phần tử nào có key là id? Nếu không có thì error sẽ thông báo lỗi 13. Ngược lại nếu tìm thấy danh biểu có khóa id tại vị trí index thì thực thi lệnh loc := index;

```
        end;
Procedure scopeentry;
        begin if b = btablim then error (14)
                else    begin btab [b] := t; b := b + 1
                        end;
        end;
Procedure Scope exit;
        begin b := b – 1; t := btab [b]
        end;
```

8.4. Bảng danh biểu băm (hash symbol table)

Một danh biểu $\xrightarrow{H}$ chỉ số k

Chúng ta lấy lại thí dụ 8.1 để minh họa việc xây dựng bảng danh biểu theo phương pháp băm ở (H.8.2). Giả sử A biến đổi H có k = 3, B có k = 6 và C có k = 5.



$A \xrightarrow{H} k = 3$

$A \xrightarrow{H} k = 6$

$A \xrightarrow{H} k = 5$

Hình 8.2. Bảng danh biểu băm

Các tác vụ làm việc trên bảng danh biểu băm được trình bày bằng các chương trình con sau:

```
const   hashsize = ...;
        tabsize = ...;
        btabsize = ...;
type    tabindex = 1 .. tabsize;
        hashindex = 1 .. hashsize;
        iterm = record
                key: id;
                ptr: tabindex;
                end;
var     tab: array [1 .. tablim] of item
        hash: array [hashindex] of tabindex;
        btab: array [i .. btabsize] of tabindex;
```

```
        t: tabindex; b: 1 .. Btabsize;
function H (id: alfa): hashindex;
        begin


        end;
procedure enter (id: alfa);
        var sb: tabindex; k: hashindex; ind: tabindex;
        begin
            k := H(id);
            ind := HASH [k];
            sb := btab [b – 1];
            if HASH [k] < > 0 then
            while ind > = sb do
                if id = tab [ind]. key then error (11) ...
```

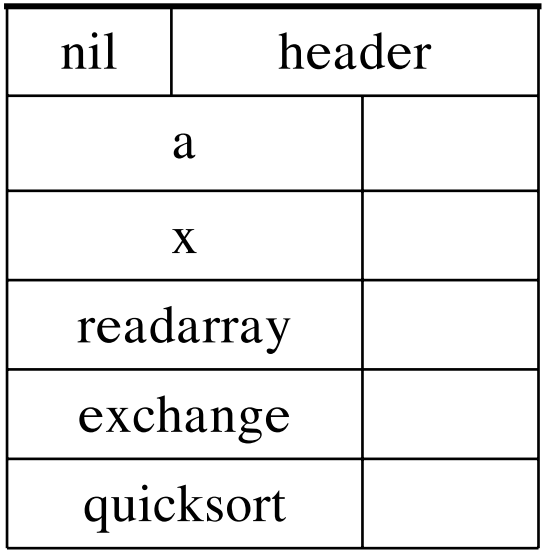
```
                    /* trùng tên danh biểu trong cùng tầm vực */
                else ind := tab [ind]. Ptr; {không trùng tên}
            if t = tabsize then error (12)
            else begin
                tab [t]. Key :=  id;
                tab [t]. Ptr := HASH [k];
                HASH [k] := t;
                t := t + 1;
            end;
        end;
function loc (id: alfa): tabindex;
        var q: boolean; ind: tabindex;
        begin ind := HASH [H(id)];
```

```
        q := false;
    while (ind < > 0) and (not (q)) do
        begin q := id = tab [ind]. Key;
            if not (q) then ind := tab [ind]. Ptr
        end;
    if q then loc := ind else error (13);
        /* chưa có danh biểu trong bảng danh biểu */
    end;
procedure Scopeentry;
    begin   if b = tabsize then error (14)
            else    begin btab [b] := t;
                        b := b + 1;
                    end;
```

```
procedure Scopeexit;
        var     ind: tabindex;
                k: hashindex;
        begin
                ind := t; t := btab [b – 1];
                b := b – 1;
        while ind > t do
            begin       ind := ind – 1;
                        k := H (tab [ind]. Key);
                        HASH [k] := tab [HASH [k]]. Ptr;
            end;
        end;
```

8.5. Hàm băm (hashing function)

8.6. Lưu giữ thông tin của tầm vực ý nghĩa

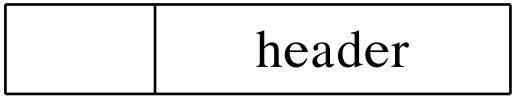


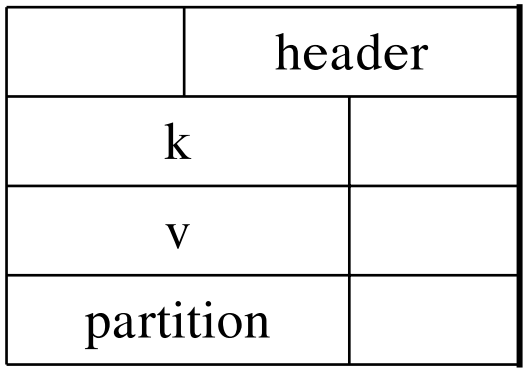
sort

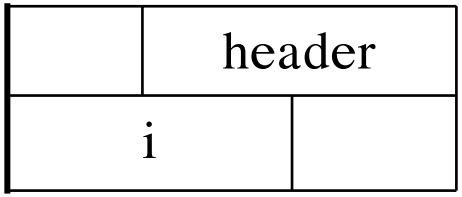
bảng readarray

bảng exchange

exchange

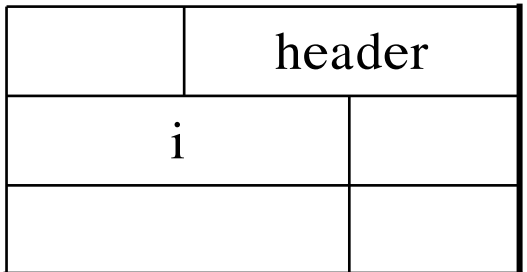






readarray

partition



Muốn thực hiện việc tạo bảng danh biểu cho chương trình con bị gọi, ta phải tạo các hàm như sau:

1. mktable (x)
2. enter (table, name, type, offset)
3. addwidth (table, width)
4. enterproc (table, name, newtable)