

Representación en memoria en las pilas

Las pilas no son estructuras fundamentales de datos; es decir no están definidas como tales en los lenguajes de programación. Para su representación requieren de otras EDs, como:

- Arreglos
- Listas

OC/SG utilizan arreglos. Es importante definir el tamaño del máximo de la pila, así como una variable auxiliar que se denomina TOPE. Esta variable se utiliza para indicar el último elemento que se insertó en la pila.

Las dos operaciones aplicables a todas las pilas son:

Una operación apilar, en la que un elemento de datos se coloca en el lugar apuntado por el puntero de pila, y la dirección en el puntero de pila se ajusta por el tamaño de los datos de partida.

Una operación desapilar: un elemento de datos en la ubicación actual apuntado por el puntero de pila es eliminado, y el puntero de pila se ajusta por el tamaño de los datos de partida.

Hay muchas variaciones en el principio básico de las operaciones de pila. Cada pila tiene un lugar fijo en la memoria en la que comienza. Como los datos se añadirán a la pila, el puntero de pila es desplazado para indicar el estado actual de la pila, que se expande lejos del origen (ya sea hacia arriba o hacia abajo, dependiendo de la aplicación concreta).

Por ejemplo, una pila puede comenzar en una posición de la memoria de mil, y ampliar por debajo de las direcciones, en cuyo caso, los nuevos datos se almacenan en lugares que van por debajo de 1000, y el puntero de pila se decrementa cada vez que un nuevo elemento se agrega. Cuando un elemento es eliminado de la pila, el puntero de pila se incrementa.

Los punteros de pila pueden apuntar al origen de una pila o de un número limitado de direcciones, ya sea por encima o por debajo del origen (dependiendo de la dirección en que crece la pila), sin embargo el puntero de pila no puede cruzar el origen de la pila. En otras palabras, si el origen de la pila está en la dirección 1000 y la pila crece hacia abajo (hacia las direcciones 999, 998, y así sucesivamente), el puntero de pila nunca debe ser incrementado más allá de 1000 (para 1001, 1002, etc.) Si una desapilar operación en la pila hace que el puntero de pila se deje atrás el origen de la pila, una pila se produce desbordamiento. Si una operación de apilar hace que el puntero de pila incremente o decremente más allá del máximo de la pila, en una pila se produce desbordamiento.

Operaciones básicas de las pilas

Operación CrearPila

La creación de la pila se realizará mediante el constructor por defecto. La tarea que deberá realizar será establecer un estado inicial en el que no existen elementos en la pila:

```
Pila::Pila ()
```

```
{  
    cima = -1;  
}
```

Operación PilaVacía

Esta operación permitirá determinar si la estructura tiene o no elementos almacenados. Aunque el array empleado tiene un número de elementos fijo (MAX), no todos ellos representan valores almacenados en la pila. Sólo están almacenados en la pila los valores comprendidos entre los índices 0 (fondo de la pila) y cima. Por lo tanto, la pila estará vacía cuando la cima indique una posición por debajo del fondo:

```
bool Pila::PilaVacía ()
```

```
{  
    return (cima == -1);  
}
```

Operación Apilar

La operación de inserción normalmente se conoce por su nombre inglés Push, o Apilar. La operación, aplicada sobre una pila y un valor, almacena el valor en la cima de la pila. Esta operación está restringida por el tipo de representación escogido. En este caso, la utilización de un array implica que se tiene un número máximo de posibles elementos en la pila, por lo tanto, es necesario comprobar, previamente a la inserción, que realmente hay espacio en la estructura para almacenar un nuevo elemento. Con esta consideración, el algoritmo de inserción sería:

```
bool Pila::Apilar (Valor x)
```

```
{  
    bool ok;  
    if (cima == MAX)  
        ok = false;  
    else  
    {  
        cima++;  
        datos[cima] = x;  
    }
```

```

        ok = true;
    }
    return (ok);
}

```

Operación Desapilar

La operación de borrado elimina de la estructura el elemento situado en la cima. Normalmente recibe el nombre de Pop en la bibliografía inglesa. Eliminar un elemento de la pila consiste fundamentalmente en desplazar (decrementar) la cima. Puesto que no es posible eliminar físicamente los elementos de un array, lo que se hace es dejar fuera del rango válido de elementos al primero de ellos.

```
bool Pila::Desapilar (void)
```

```

{
    bool ok;
    if (cima == -1)
        ok = false;
    else
    {
        cima--;
        ok = true;
    }
    return (ok);
}

```

Operación CimaPila

La operación de consulta permite conocer el elemento almacenado en la cima de la pila, teniendo en cuenta que si la pila está vacía no es posible conocer este valor.

```
bool Pila::CimaPila (Valor & x)
```

```

{
    bool ok;
    if (cima == -1)
        ok = false;
    else
    {
        x = datos[cima];
        ok = true;
    }
    return (ok);
}

```

Referencias:

Cairó, O., & Guardati, S. (2002). Estructuras de datos (3rd ed.). México: McGrawHill.

Joyanes Aguilar, L., & Zahonero Martínez, I. (1998). Estructura de datos. Algoritmos, abstracción y objetos. (Ed. rev.). Madrid, España: McGraw-Hill.