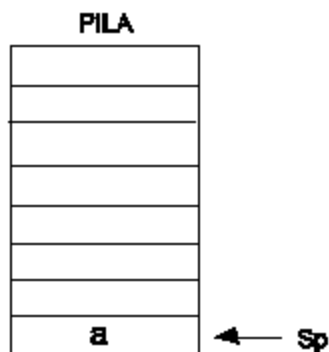


Representación en memoria de las pilas

Las pilas no son estructuras de datos fundamentales, es decir, no están definidas como tales en los lenguajes de programación. Las pilas pueden representarse mediante el uso de :

- Arreglos.
- Listas enlazadas.

Nosotros ahora usaremos los arreglos. Por lo tanto debemos definir el tamaño máximo de la pila, además de un apuntador al último elemento insertado en la pila el cual denominaremos SP. La representación gráfica de una pila es la siguiente:



Como utilizamos arreglos para implementar pilas, tenemos la limitante de espacio de memoria reservada. Una vez establecido un máximo de capacidad para la pila, ya no es posible insertar más elementos.

Una posible solución a este problema es el uso de espacios compartidos de memoria. Supongase que se necesitan dos pilas , cada una con un tamaño máximo de n elementos. En este caso se definirá un solo arreglo de $2*n$ elementos, en lugar que dos

arreglos de n elementos.

En este caso utilizaremos dos apuntadores: SP1 para apuntar al último elemento insertado en la pila 1 y SP2 para apuntar al último elemento insertado en la pila 2. Cada una de las pilas insertará sus elementos por los extremos opuestos, es decir, la pila 1 iniciará a partir de la localidad 1 del arreglo y la pila 2 iniciará en la localidad $2n$. De este modo si la pila 1 necesita más de n espacios (hay que recordar que a cada pila se le asignaron n localidades) y la pila 2 no tiene ocupados sus n lugares, entonces se podrán seguir insertando elementos en la pila 1 sin caer en un error de desbordamiento.

Especificaciones de una pila Las operaciones que sirven para definir una pila y poder manipular su contenido son las siguientes (no todas ellas se implementan al definir una pila):

Operaciones básicas de las pilas

Tipo de dato	Dato que se almacena en la pila.
Operaciones	
<i>Crear Pila</i>	Inicia.
<i>Insertar (push)</i>	Pone un dato en la pila.
<i>Quitar (pop)</i>	Retira (saca) un dato de la pila.
<i>Pila vacía</i>	Comprueba si la pila no tiene elementos.
<i>Pila llena</i>	Comprueba si la pila está llena de elementos.
<i>Limpiar pila</i>	Quita todos sus elementos y deja la pila vacía.
<i>Cima Pila</i>	Obtiene el elemento cima de la pila.
<i>Tamaño de la pila</i>	Número de elementos máximo que puede contener la pila.

1. Datos de la pila (TipoDato es cualquier tipo de dato primitivo o tipo clase).
2. CrearPila inicializa una pila. Es como crear una pila sin elementos, por tanto, vacía.
3. Verificar que la pila no está llena antes de insertar o poner ("push") un elemento en la pila; verificar que una pila no está vacía antes de quitar o sacar ("pop") un elemento de la pila. Si estas precondiciones no se cumplen, se debe visualizar un mensaje de error y el programa debe terminar.
4. Pila Vacía devuelve verdadero si la pila está vacía y falso en caso contrario.
5. Pila Llena devuelve verdadero si la pila está llena y falso en caso contrario. Estas dos últimas operaciones se utilizan para verificar las precondiciones de insertar y quitar.
6. Limpiar Pila vacía la pila, dejándola sin elementos y disponible para otras tareas.
7. Cima Pila devuelve el valor situado en la cima de la pila, pero no se decrementa su puntero, ya que la pila queda intacta.

Bibliografias

Alma Leticia Palacios Guerrero , Algoritmos y Estructuras de Datos, 2006