Laboratorio de Computación II

Asignación dinámica de memoria

Asignación dinámica de memoria

- Proceso que permite solicitar memoria adicional al sistema operativo en tiempo de ejecución.

- Nos permite utilizar la memoria exacta que necesitamos para trabajar y, una vez utilizada, debemos liberarla.

- Nos permite utilizar una mayor cantidad de memoria que de la manera convencional.

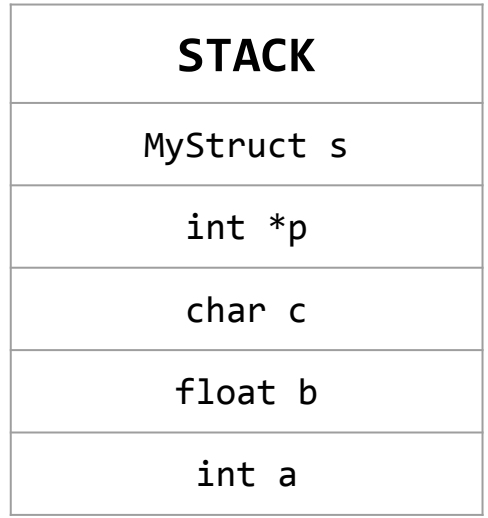
Asignación dinámica de memoria

- La memoria se puede clasificar, según su ubicación, en stack o heap.
- Esto significa que una variable puede figurar en la memoria stack o en la memoria heap dependiendo de cómo la declaremos.
- Hasta este momento siempre utilizamos la memoria stack.
- La memoria dinámica ubica la información de nuestro programa en la memoria heap (que es compartida por otros programas).

Memoria stack

- Cada variable que declaremos en una función (incluso main) se apila en la memoria stack.
- La memoria stack es limitada. De superar el límite genera una excepción (Desbordamiento de pila).

```
int main(){
  int a, float b, char c;
  int *p;
  struct MyStruct s;
}
```



Pedir memoria dinámica

Proceso que se realiza mediante la función malloc

```
void * malloc ( long int  bytes );
```

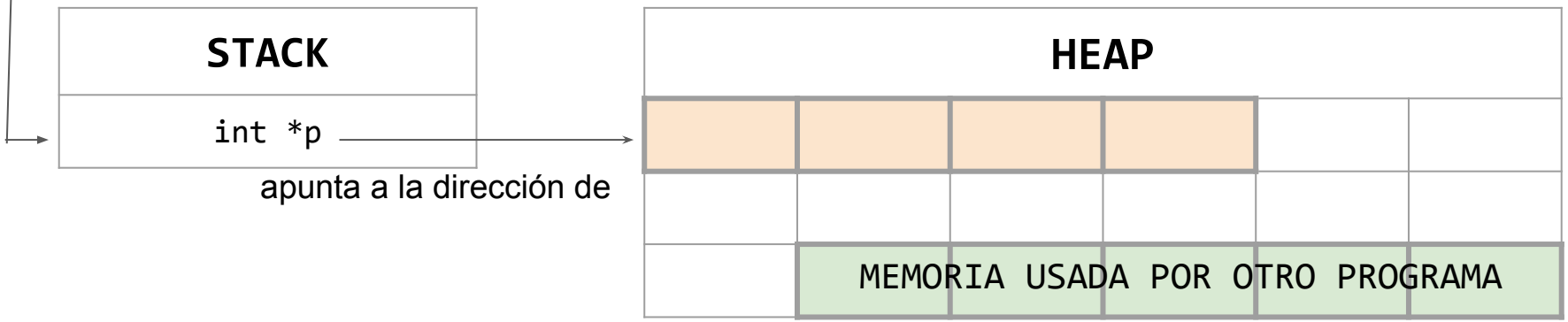
- Devuelve un puntero void, por lo que, hay que castear al tipo de dato necesitado o NULL si no hay memoria disponible.
- Es necesario un puntero a ese tipo de dato.
- Debe liberarse con free

```
int main(){
/* Memoria dinámica para un vector de 500
elementos */
  int *vec;
  vec = (int *) malloc (500 * sizeof (int));
  if (vec == NULL)
    exit(1); // No hay memoria

  // Resto del programa

  free (vec);
}
```

Memoria heap

Es utilizada cuando pedimos memoria dinámica.

```c
int main(){
  float *p;
  p = (float *) malloc (4 * sizeof (float));
  free(p);
}
```



Ejemplo en C/C++