

# Manejo de memoria dinámica

Stefano Romero

# Memoria dinámica

- La localización de memoria dinámica es necesaria para manejar el espacio disponible en memoria.
- Por ejemplo, en el tiempo de compilación, puede que no sepamos cuánta memoria necesitaremos para ejecutar el programa, sobre todo en arreglos con una gran cantidad de elementos.
- C, a diferencia de Python, no tiene un manejo automatizado de manejo de memoria, por lo que es de interés la utilización de memoria dinámica.
- Las funciones de interés se encuentran en la librería `<stdlib.h>`.

# Memoria dinámica - Funciones

- calloc:

```
void *calloc( N, size)
```

Asigna memoria para un arreglo de N elementos del tamaño size. Retorna un puntero en donde está asignada la memoria. La memoria se asigna con el valor de 0.

- malloc:

```
void *malloc( N, size)
```

Asigna memoria para un arreglo de N elementos (byte). Retorna un puntero en donde está asignada la memoria.

# Memoria dinámica - Funciones

- realloc()

```
void *realloc(void *ptr, size)
```

Cambia el tamaño del bloque de memoria. La nueva asignación no cambiará a los valores de memoria del tamaño anterior.

- free

```
void free (void *ptr)
```

Libera el espacio de memoria del puntero que ha generado previamente malloc, calloc o realloc.

# Memoria dinámica - Ejemplos

- Reservar en memoria un arreglo de N elementos enteros. Luego, limpiar la memoria asignada:

```
int* A = (int *)malloc(n*sizeof(int));
```

```
free(A)
```

Considerar que malloc siempre devuelve un puntero a void, por lo que siempre será necesario hacer un cast al tipo de datos que usemos para el arreglo.

# Memoria dinámica - Ejemplos

- Crear un arreglo de N elementos enteros con valor inicial 0. Luego, limpiar la memoria asignada:

```
int* A = (int *)calloc(n, sizeof(int));
```

```
free(A)
```

# Memoria dinámica

- ¿Qué sucede si en lugar de almacenar solo un arreglo, se quisiera almacenar una matriz?

Las matrices se suelen guardar en la memoria de forma lineal (sea por fila o columna), por tanto, también es posible extrapolar el uso de la memoria dinámica utilizando las funciones malloc o calloc.

# Memoria dinámica - Ejemplo

- Reservar un espacio de memoria para una matriz de int de r filas y c columnas.

## Alternativa 1:

```
int *arr = (int *)malloc(r*c*sizeof(int));
```

```
free(arr);
```

## Alternativa 2:

```
int** arr = (int**)malloc(r * sizeof(int*));
```

```
    for (int i = 0; i < r; i++)
```

```
        {arr[i] = (int*)malloc(c * sizeof(int));}
```

```
for (int i = 0; i < r; i++)
```

```
    {free(arr[i]);}
```

```
free(arr);
```



# Bibliografía

- Lecture 08, Dynamic Memory Allocation, Ananda Gunawardena