Problemas Con Memoria Dinámica

Lo que siempre nos pasa y no nos damos Cuenta

Punteros

Int	Un valor entero
Int *	Un puntero a (un lugar de memoria que contiene) un entero
Int **	Un puntero a un lugar de memoria que contiene un puntero a un entero
Void *	Un puntero hacia un dato de tipo desconocido

Funciones

malloc() = asigna *size* bytes y devuelve un puntero a la memoria asignada. La memoria no es borrada.

free() =libera el espacio de memoria apuntado por *ptr*, que debe haber sido devuelto por una llamada previa a **malloc()**, **calloc()** o **realloc()**. En caso contrario, o si **free(*ptr*)** ya se ha llamado antes, se produce un comportamiento indefinido. Si *ptr* es **NULL**, no se realiza ninguna operación.

Errores Comunes

Olvidarse de Reservar Memoria

```
char* src;

char* dst;

strcopy(src,dst)
```

Olvidarse de Reservar Memoria

```
char* src=”hello”;

char* dst=(malloc(length(src)+1) );

strcopy(src,dst)
```

Reservar Menos Memoria

```
char* src=”hello”;

char* dst=(malloc( length(src) ) );

strcopy(src,dst)
```

Olvidarse de Inicializar la Memoria Reservada

malloc() NO INICIALIZA LA MEMORIA QUE RESERVA!!!!!

Olvidarse de Liberar Memoria Previamente Reservada

- Eso se denomina “**memory leak**”
- No liberamos memoria con free()

Olvidarse de Liberar Memoria Previamente Reservada

```
void func(){
    char *dp = malloc(A_CONST);
    /* ... */
    free(dp);       /* dp now becomes a dangling pointer */
    dp = NULL;      /* dp is no longer dangling */
    /* ... */
}
```

- Eso se denomina “**dangling pointer**”

Liberar Memoria Previamente Reservada más de una vez

- Eso se denomina “**double free**”

  free(buffer)

  ..

  free(buffer)