

Direcciones de literales cadena

Guillermo Hernández

2023-12-10

Versión en línea: <https://www.dih5.es/array-decay-es.html>.

Considérese el siguiente código:

```
#include <stdio.h>

int main() {
    char *s="hola";
    printf("%p\n",&s);
    printf("%p\n",s);
    printf("%p\n",&"hola");

    return 0;
}
```

¿Cuál de las siguientes es cierta?

- El programa imprime la misma dirección de memoria tres veces.
- El programa imprime tres direcciones de memoria, pero no son las tres iguales.
- El programa da un error de ejecución porque se está guardando en el puntero un valor sin haber reservado espacio.
- El programa no compila porque &"hola" es ilegal, ya que se aplica el operador dirección de a un literal.

La respuesta correcta es la segunda. En la primera línea del main creamos un puntero a char y guardamos en él la dirección de memoria donde hay un literal cadena "hola", todo correcto. También es legal la operación &"hola", que recuperará la dirección de memoria del programa donde está guardado el literal. Paradójicamente, un literal cadena es un **lvalue**, a pesar de que no pueden ir al lado izquierdo de una asignación. Respecto a las direcciones, está claro que &s y s son diferentes, porque la primera es la dirección del puntero y la segunda su contenido, la dirección de la cadena "hola". Con esto hemos resuelto el test.

Queda una cuestión que no nos pedía distinguir las opciones del test: ¿coincidirán s y &"hola"? La respuesta más correcta es que **depende**. La descripción

del estándar de C no dice sobre los arrays en que se almacenan los literales cadena que «It is unspecified whether these arrays are distinct provided their elements have the appropriate values» (C11 6.4.5/7). Es decir, que queda en manos del compilador. Lo más cabal sería que, sobre todo en este escenario tan sencillo, lo reconozca y haga que estas dos direcciones sí coincidan. En todas las pruebas que he hecho con gcc y clang ha sido así.

Una observación interesante es que la direcciones de memoria que observamos no serán cercanas. El literal &"hola" se almacena en el segmento de datos, mientras que el puntero s estará en la pila (*stack*). Si en su lugar hubiéramos usado una variable global, esta también se hubiera ubicado en el segmento de datos.