

Ejemplo sobre punteros en C

Arquitectura del Computador - LCC - FCEIA-UNR

Agosto 2021

En este ejemplo se pretende repasar el tema de punteros en C. Si bien es un ejemplo muy elemental de punteros en C, nos va a servir para practicar con GDB.

Dado el siguiente código en lenguaje C `punteros.c`:

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int a=145; //Declaración de variable entera
6     int *puntero; //Declaración de variable puntero de tipo entero
7     puntero = &a; //Asignación de la dirección de memoria de a
8     printf("El valor de a es: %d\n", a);
9     printf("El valor de *puntero es: %d\n", *puntero);
10    printf("La dirección de memoria de *puntero es: %p\n", puntero);
11    return 0;
12 }
```

Si compilamos y ejecutamos el código, la salida es la siguiente:

```
$ ./a.out
El valor de a es: 145
El valor de *puntero es: 145
La dirección de memoria de *puntero es: 0x7ffc33bd4954
```

Ahora compilemos utilizando la opción `-g` para poder *debuggear* utilizando GDB:

```
$ gcc -g punteros.c
```

y ejecutemos con GDB:

```
$ gdb ./a.out
```

Una vez que estamos dentro de la sesión de *debugging*, ponemos un *breakpoint* en `main` y ejecutamos el comando `run`. Luego vamos ejecutando línea a línea hasta ejecutar los tres `printf`:

```
(gdb) br main
Breakpoint 1 at 0x113d: file printt.c, line 4.
(gdb) r
Starting program: /home/dferoldi/2021/agosto/a.out

Breakpoint 1, main () at printt.c:4
4         int a=145; //Declaración de variable entera
(gdb) n
6         puntero=&a; //Asignación de la dirección de memoria de a
```

```

(gdb) n
7         printf("El valor de a es: %d\n", a);
(gdb) n
El valor de a es: 145
8         printf("El valor de *puntero es:  %d\n", *puntero);
(gdb) n
El valor de *puntero es:  145
9         printf("La dirección de memoria de *puntero es %p\n", puntero);
(gdb) n
La dirección de memoria de *puntero es 0x7fffffff104
10        return 0;

```

Ahora verifiquemos lo anterior utilizando comandos de GDB. En primer lugar, podemos usar el comando `print` para ver el contenido de la variable `a`, luego podemos imprimir a puntero dicha variable y finalmente examinar el contenido de la memoria utilizando el comando `examine`¹:

```

(gdb) p a
$1 = 145
(gdb) p &a
$2 = (int *) 0x7fffffff1d4
(gdb) x/d &a
0x7fffffff1d4: 145

```

Efectivamente, vemos que en la dirección `0x7fffffff1d4`, que es la dirección almacenada en la variable tipo puntero `a` entero `puntero`, se encuentra almacenado el valor 145.

¹Utilizamos la opción `d` para ver el resultado en formato decimal.