

Ejercicio: 00 ft_putchar

Directorio de entrega: ex00/

Archivos a entregar: ft_putchar.c

Funciones autorizadas: write •

Escribe una función que muestre el carácter usado como parámetro. •

El prototipo de la función deberá ser el siguiente: **void ft_putchar(char c);**

Para mostrar el carácter, tendrás que utilizar la función write de la siguiente forma:

write(1, &c, 1);

```
#include <unistd.h>

void ft_putchar(char c) {
    write(1, &c, 1);
}

int main() {
    char character = 'A'; // Puedes cambiar este valor por el carácter que desees
    mostrar
    ft_putchar(character);
    return 0;
}
```

En este ejemplo, la función `ft_putchar` toma un carácter como parámetro y utiliza la función `write` para escribir ese carácter en la salida estándar (en este caso, la consola). La función `main` simplemente llama a `ft_putchar` con un carácter específico ('A' en este caso) para probar la función. Puedes cambiar el valor de `character` por cualquier otro carácter que desees mostrar.

Con gusto te explico cada línea del código:

Inclusión de bibliotecas:

c

Copy code

```
#include <unistd.h>
```

- Esta línea incluye la biblioteca `unistd.h`, que proporciona la declaración de la función `write`, entre otras cosas. Esta función es necesaria para escribir en la salida estándar.

Definición de la función `ft_putchar`:

`c`

Copy code

```
void ft_putchar(char c)
```

- Esta línea define una función llamada `ft_putchar` que toma un parámetro de tipo `char` llamado `c`. Esta función no devuelve ningún valor (`void`).

Llamada a la función `write`:

`c`

Copy code

```
write(1, &c, 1);
```

- Esta línea utiliza la función `write` para escribir en la salida estándar.
- *El primer argumento (1) es el descriptor de archivo que representa la salida estándar (la consola).*
- El segundo argumento (`&c`) es la dirección de memoria del carácter `c`.
- El tercer argumento (1) es el número de bytes que se van a escribir, que en este caso es solo un carácter.

Función `main`:

```
int main() {
```

- Esta línea define la función principal del programa, `main`, que no toma ningún argumento y devuelve un valor entero (`int`).

Declaración de la variable `character`:

```
char character = 'A';
```

- Esta línea declara una variable `character` de tipo `char` y le asigna el valor `'A'`. Puedes cambiar este valor por cualquier otro carácter que desees mostrar.

Llamada a `ft_putchar` dentro de `main`:

- Esta línea llama a la función `ft_putchar` pasando la variable `character` como argumento. Esto imprimirá el valor de `character` en la salida estándar.

Finalización de la función `main`:

c

Copy code

```
return 0;
```

- Esta línea indica que la función `main` ha finalizado correctamente y devuelve el valor `0`, indicando que el programa se ejecutó sin errores.