

33.- ft_putendl_fd.-

BSD Man Page References:

- `write(2)`: Used for writing data to a file descriptor.
- `ft_strlen(3)` (custom, not standard): For calculating string length.

Synopsis:

```
void ft_putendl_fd(char *s, int fd);
```

Purpose:

This function writes a null-terminated string (`s`) followed by a newline character (`\n`) to a specified file descriptor (`fd`).

Sinopsis:

```
void ft_putendl_fd(char *s, int fd);
```

Parameters:

- `char *s`: Pointer to the string to be written.
- `int fd`: File descriptor to write to.

Code Explanation:

```
void ft_putendl_fd(char *s, int fd)
{
    if (!s) // Check for null input
    {
        return;
    }

    write(fd, s, ft_strlen(s)); // Write the string
    write(fd, "\n", 1); // Write the newline character
}
```

- **Check for null input:** Returns if the pointer `s` is null.
- **Write the string:** Uses `write` to write the characters of `s` to `fd` with the number of bytes determined by `ft_strlen(s)`.
- **Write the newline:** Uses `write` to write the newline character (`\n`) to `fd` with 1 byte.

Key Points:

- This function provides a convenient way to print a string including a newline to different output streams specified by file descriptors.
- It relies on the `write` system call for efficient data writing.

Example Usage (Main Function):

```
int main(void)
{
    ft_putendl_fd("Adios Mundo Cruel", 1); // Write to standard output (FD 1)
}
```

```
    return (0);  
}
```

- This will print "Adios Mundo Cruel" followed by a newline to the standard output console.

Additional Notes:

- Remember to include the `libft.h` header file before using this function.
- Ensure proper error handling when using `write`.
- Consider using `strlen(s)` directly instead of `ft_strlen(s)` if performance is critical.

Here's a summary of the key differences between `ft_putstr_fd` and `ft_putendl_fd`:

Function	Purpose	Output
<code>ft_putstr_fd</code>	Writes a null-terminated string to a file descriptor.	String only - does not add a newline.
<code>ft_putendl_fd</code>	Writes a null-terminated string followed by a newline to a file descriptor.	String + newline character (" <code>\n</code> ").

In a nutshell:

- `ft_putstr_fd` prints the string **without** adding a newline.
- `ft_putendl_fd` prints the string **and** adds a newline, essentially starting a new line after the string.

Example usage:

```
ft_putstr_fd("Hello", 1); // Prints "Hello" to standard output (FD 1).  
ft_putendl_fd("World", 2); // Prints "World\n" to standard error (FD 2).
```

Additional notes:

- Both functions use `write` for efficient data writing to file descriptors.
- They offer flexibility for printing to different output streams beyond just standard output.
- When choosing between them, consider whether you want to include a newline character after the string.