

ft_split.c

Este código es una implementación de la función **ft_split** en C, que tiene como objetivo dividir una cadena de texto en palabras según un delimitador especificado. La función devuelve un arreglo de punteros a cadenas, donde cada cadena es una palabra de la entrada original.

En esta implementación, la función **ft_split** está compuesta por varias funciones auxiliares, cada una con una tarea específica. La primera es **ft_strndup**, que crea una copia de una cadena de texto con una longitud especificada. La función **num_words** cuenta el número de palabras en una cadena de texto según un delimitador especificado. La función **get_next_word** devuelve la siguiente palabra en una cadena de texto según un delimitador especificado.

La función **ft_split** comienza por contar el número de palabras en la cadena de texto utilizando la función **num_words**. Luego, reserva memoria para el arreglo de punteros a cadenas utilizando la función **malloc**. Si la reserva de memoria falla, la función devuelve **NULL**.

La función **ft_split** itera sobre la cadena de texto y utiliza la función **get_next_word** para obtener cada palabra en la cadena. La función **get_next_word** devuelve una copia de la palabra utilizando la función **ft_strndup**. La función **ft_split** almacena cada palabra en el arreglo de punteros a cadenas y continúa hasta que se procesa toda la cadena de texto.

Finalmente, la función **ft_split** agrega un puntero nulo al final del arreglo de punteros a cadenas para indicar el final de la lista. La función devuelve el arreglo de punteros a cadenas.

La función **ft_split** es útil cuando se necesita dividir una cadena de texto en palabras y procesar cada palabra de forma individual. Por ejemplo, se puede utilizar para tokenizar una cadena de texto y procesar cada token de forma individual. La función es segura y puede manejar cadenas de texto vacías y cadenas de texto con un solo delimitador.

En resumen, la función **ft_split** divide una cadena de texto en palabras según un delimitador especificado y devuelve un arreglo de punteros a cadenas, donde cada cadena es una palabra de la entrada original. La función está compuesta por varias funciones auxiliares y utiliza la reserva de memoria dinámica para almacenar las palabras. La función es útil cuando se necesita procesar cada palabra de forma individual en una cadena de texto.

La función static (interna): **ft_strndup**

La función **ft_strndup** se encarga de duplicar una parte de una cadena original, desde el inicio hasta un número determinado de caracteres. Su prototipo es la siguiente:

```
char *ft_strndup(const char *s, size_t n);
```

Donde **s** es la cadena original y **n** es el número de caracteres que se desean duplicar.

El funcionamiento de **ft_strndup** se puede resumir en los siguientes pasos:

1. Se reserva memoria para la nueva cadena utilizando **malloc**.
2. Se copian los caracteres de la cadena original a la nueva cadena utilizando **memcpy**.
3. Se agrega un carácter nulo (**\0**) al final de la nueva cadena para indicar su fin.
4. Se devuelve la nueva cadena duplicada.

Relación con **ft_split**

La función **ft_split** se utiliza para dividir una cadena en subcadenas según un carácter delimitador.

```
char **ft_split(char const *s, char c);
```

Donde **s** es la cadena original y **c** es el carácter delimitador.

La relación entre **ft_strndup** y **ft_split** se establece en el proceso de división de la cadena original en subcadenas. Cuando **ft_split** encuentra un carácter delimitador, utiliza **ft_strndup** para duplicar la parte de la cadena original que se encuentra entre el inicio de la cadena y el carácter delimitador.

Por ejemplo, si se llama a **ft_split** con la cadena "hello,world,foo" y el carácter delimitador ",", la función utilizará **ft_strndup** para duplicar las siguientes partes de la cadena:

- "hello" (desde el inicio hasta el primer carácter delimitador)
- "world" (desde el primer carácter delimitador hasta el segundo)
- "foo" (desde el segundo carácter delimitador hasta el final de la cadena)

En resumen, **ft_strndup** es una función fundamental dentro de **ft_split** que se encarga de duplicar partes de la cadena original.

Su relación con otras funciones como **ft_strlen**, **ft_strchr** y **ft_malloc** permite a **ft_split** dividir la cadena original en subcadenas según un carácter delimitador de manera eficiente y segura.

La función static (interna): **num_words**

La función **num_words** es una función estática que se encarga de contar el número de palabras en una cadena de caracteres (**const char *s**) separadas por un carácter específico (**char c**). Esta función es parte de un conjunto de funciones que trabajan juntas para implementar la función **ft_split**, que divide una cadena de caracteres en un array de cadenas de caracteres separadas por un carácter específico.

Funcionamiento de num_words

La función **num_words** itera sobre la cadena de caracteres **s** y cuenta el número de palabras separadas por el carácter **c**. La función utiliza un bucle **while** que itera sobre la cadena **s** hasta que se alcanza el final de la cadena. Dentro del bucle, se verifica si el carácter actual es diferente del carácter separador **c**. Si es así, se incrementa el contador **num** y se itera sobre la cadena hasta que se encuentra el próximo carácter separador **c**. Si se alcanza el final de la cadena, se sale del bucle.

Relación con otras partes del código

La función **num_words** es utilizada por la función **ft_split** para determinar el número de palabras en la cadena de caracteres que se va a dividir. La función **ft_split** utiliza el resultado de **num_words** para asignar memoria para el array de cadenas de caracteres que se va a devolver.

La función **get_next_word** también está relacionada con **num_words**, ya que se utiliza para obtener la próxima palabra en la cadena de caracteres. La función **get_next_word** utiliza la función **ft_strdup** para crear una copia de la palabra actual y devuelve una cadena de caracteres que representa la palabra.

En resumen, la función **num_words** es una parte fundamental del conjunto de funciones que implementan la función **ft_split**. Su función es contar el número de palabras en una cadena de caracteres separadas por un carácter específico, lo que permite a la función **ft_split** asignar memoria adecuada para el array de cadenas de caracteres que se va a devolver.

La función static (interna): num_words

La función **get_next_word** es una función estática que se encarga de extraer la próxima palabra de una cadena de caracteres **s** separada por un carácter **c**. Esta función es parte del código que implementa la función **ft_split**, que divide una cadena de caracteres en un array de palabras.

Funcionamiento de **get_next_word**

La función **get_next_word** toma dos parámetros: **s**, que es un puntero a una cadena de caracteres, y **c**, que es el carácter separador. La función devuelve un puntero a una cadena de caracteres que representa la próxima palabra en la cadena **s**.

El funcionamiento de la función se puede resumir en los siguientes pasos:

5. Se busca el inicio de la próxima palabra en la cadena **s** saltando los caracteres separadores **c**. Esto se logra mediante un bucle que avanza el puntero **start** hasta encontrar un carácter que no sea **c**.
6. Se busca el final de la palabra actual en la cadena **s** saltando los caracteres que no son separadores **c**. Esto se logra mediante un bucle que avanza el puntero **end** hasta encontrar un carácter que sea **c** o el final de la cadena.
7. Si el inicio y el final de la palabra son iguales, significa que no hay una palabra en la cadena, por lo que se devuelve **NULL**.
8. Se actualiza el puntero **s** para que apunte al final de la palabra actual.
9. Se devuelve una copia de la palabra actual utilizando la función **ft_strndup**, que crea una copia de una cadena de caracteres con una longitud específica.

Relación con otras partes del código

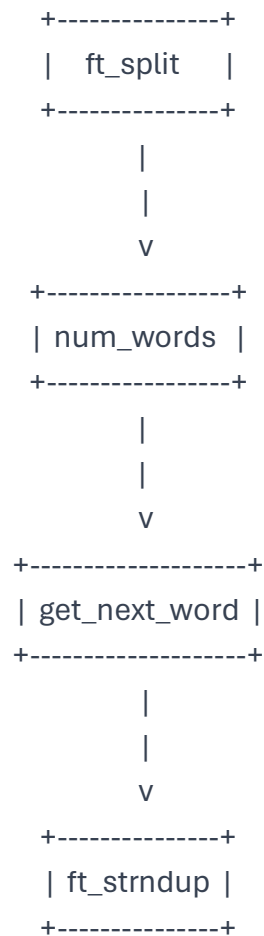
La función **get_next_word** es utilizada por la función **ft_split** para dividir una cadena de caracteres en un array de palabras. La función **ft_split** utiliza la función **num_words** para contar el número de palabras en la cadena y luego utiliza la función **get_next_word** para extraer cada palabra y almacenarla en un array.

La función **ft_strndup** es utilizada por la función **get_next_word** para crear una copia de la palabra actual. Esta función es similar a la función **strndup** de la biblioteca estándar de C, pero con una implementación propia.

En resumen, la función **get_next_word** es una parte clave del código que implementa la función **ft_split**, y se encarga de extraer las palabras individuales de una cadena de caracteres separada por un carácter específico.

Diagrama de flujo

A continuación, se muestra un diagrama de flujo que ilustra la relación entre las funciones:



En este diagrama, se puede ver que la función **ft_split** llama a la función **num_words** para contar el número de palabras en la cadena de caracteres. Luego, **ft_split** utiliza el resultado de **num_words** para asignar memoria para el array de cadenas de caracteres. La función **get_next_word** se utiliza para obtener la próxima palabra en la cadena de caracteres y devuelve una cadena de caracteres que representa la palabra. Finalmente, la función **ft_strndup** se utiliza para crear una copia de la palabra actual.