char    \*remove\_string(char \*tmp)

{

    size\_t      len;

    char        \*new\_tmp;

    size\_t      n;

    len = ft\_strchr(tmp, '\n', ft\_strlen(tmp));

    n = ft\_strlen(tmp);

    if (len == n)

    {

        free (tmp);

        tmp = NULL;

        return (tmp);

    }

    else

    {

        new\_tmp = (char \*)malloc(sizeof (char) \* (n - len + 1));

        if (!new\_tmp)

            return (ft\_free(tmp));

        ft\_strlcpy(new\_tmp, tmp, n - len + 1, len);

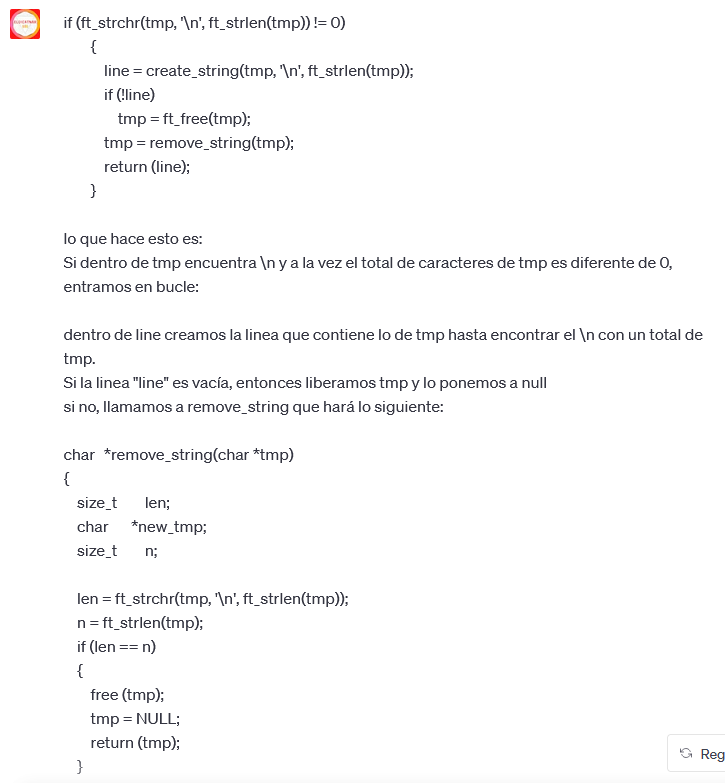
        free (tmp);

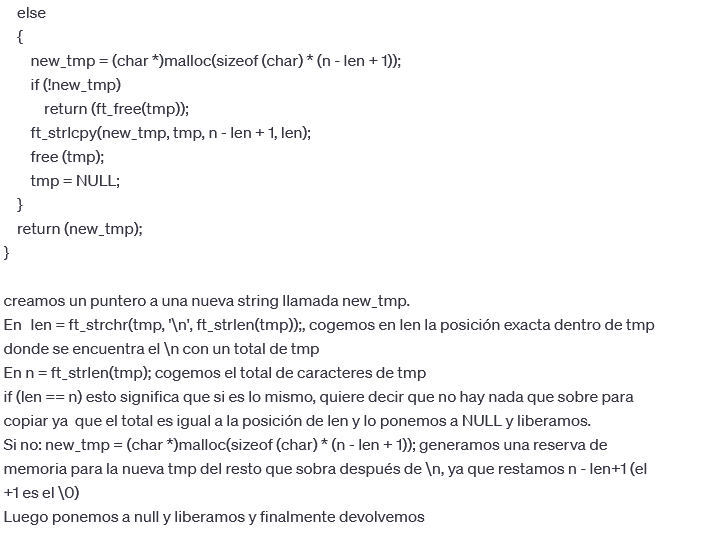
        tmp = NULL;

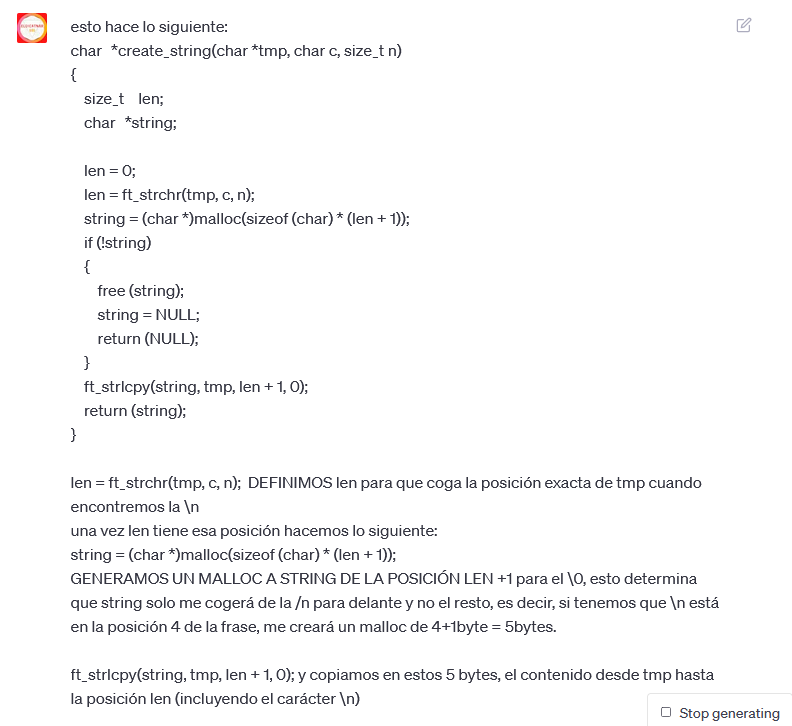
    }

    return (new\_tmp);

}



****

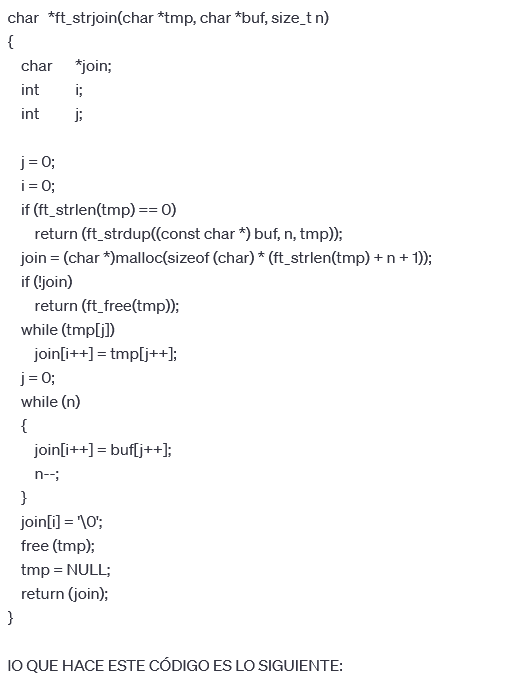
****

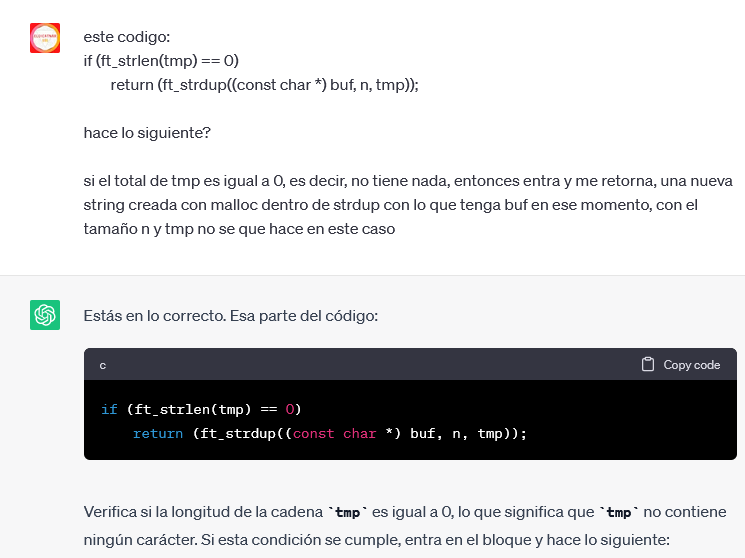
es decir el create string lo que hace es crear una string nueva con malloc de un total de bytes hasta encontrar el \n y lo pasa de tmp a string, "solo ese trocito" hasta llegar al \n

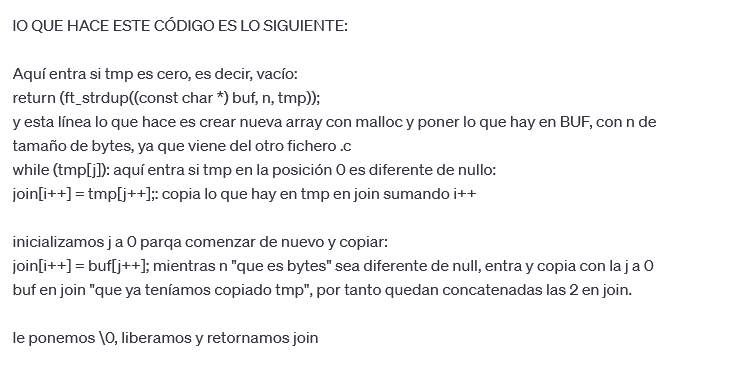
CREATE STRING COPIA SIMPLEMENTE DEL INICIO HASTA \N EN STRING.

**ejemplo: hola\n, pues string cogería hola\n**

STRJOIN



****

****

**1. ft\_strlen - Cuenta la longitud de una cadena de caracteres:**

Esta función toma una cadena de caracteres (s) como entrada y cuenta cuántos caracteres hay en esa cadena hasta llegar al final. Básicamente, está contando la cantidad de letras en la palabra.

**2. ft\_strlcpy - Copia parte de una cadena en otra cadena:**

Esta función copia una parte de una cadena de caracteres (src) en otra cadena (dst). Se copian dstsize - 1 caracteres de src en dst, y luego se agrega un carácter nulo \0 al final para asegurarse de que dst sea una cadena válida.

**3. ft\_strchr - Encuentra la posición de un carácter en una cadena:**

Esta función busca un carácter específico (c) en una cadena de caracteres (s). Comienza a buscar desde el principio de la cadena y detiene su búsqueda cuando encuentra el carácter o cuando alcanza una cantidad máxima de caracteres (n). Si encuentra el carácter, devuelve la posición en la que lo encontró.

**4. ft\_strdup - Duplica una parte de una cadena:**

Esta función crea una copia de una parte específica (n caracteres) de una cadena de caracteres (s1). Utiliza malloc para asignar memoria para la nueva cadena (tmp) y luego utiliza ft\_strlcpy para copiar la parte de la cadena original en la nueva cadena. Devuelve la nueva cadena.

**5. ft\_strjoin - Une dos cadenas de caracteres:**

Esta función toma dos cadenas de caracteres (tmp y buf) y las une en una nueva cadena (join). Si tmp es una cadena vacía, simplemente copia buf en join. Si no, asigna memoria para join, copia el contenido de tmp en join, luego agrega el contenido de buf al final de join. Finalmente, libera la memoria de tmp (para que no haya fugas de memoria) y devuelve la nueva cadena join.

La función ft\_strjoin tiene como objetivo combinar dos cadenas de caracteres en una sola cadena nueva. Una cadena es tmp, que contiene datos previos que se han leído de algún lugar, y la otra cadena es buf, que contiene datos recién leídos. La función crea una nueva cadena, join, que contendrá la combinación de los contenidos de tmp y buf.

**Pasos generales de lo que hace:**

1. Verifica si la cadena tmp está vacía. Si lo está, significa que no hay datos previos y la función simplemente crea una copia de la cadena buf y la devuelve como resultado.
2. Si tmp no está vacía, la función asigna memoria para la cadena join, que será lo suficientemente grande para contener los contenidos de tmp y buf.
3. Copia los caracteres de tmp a join para mantener los datos anteriores.
4. Luego, copia los caracteres de buf a join para agregar los nuevos datos al final.
5. Asegura que join sea una cadena válida agregando un carácter nulo al final (esto es importante para que sepa dónde termina la cadena).
6. Libera la memoria ocupada por tmp, ya que ya no es necesaria.
7. Devuelve la cadena join, que contiene la combinación de los datos de tmp y buf.
8. **Función ft\_free**:
   * Libera la memoria ocupada por un puntero (tmp) utilizando la función free.
   * Luego, establece el puntero tmp en NULL para evitar problemas de acceso a memoria liberada.
9. **Función remove\_string**:
   * Esta función elimina parte del contenido de una cadena tmp hasta la primera aparición de un carácter de nueva línea ('\n').
   * Si no encuentra un salto de línea, libera la memoria ocupada por tmp y lo establece en NULL.
   * Si encuentra un salto de línea, crea una nueva cadena (new\_tmp) que excluye el contenido antes del salto de línea y la copia.
   * Libera la memoria de tmp, lo establece en NULL y devuelve la nueva cadena new\_tmp.
10. **Función create\_string**:
    * Crea una nueva cadena (string) a partir del contenido en tmp hasta la primera aparición de un carácter específico (c).
    * La longitud de la subcadena se calcula mediante la función ft\_strchr que busca la posición de c en tmp.
    * Reserva memoria para string, copia la subcadena en string y agrega un carácter nulo al final.
    * Si no puede reservar memoria, libera la memoria de string (si se reservó) y devuelve NULL.
11. **Función create\_nullstring**:
    * Crea una nueva cadena (line) a partir del contenido en tmp (si no está vacío).
    * Reserva memoria para line, copia el contenido de tmp y agrega un carácter nulo al final.
    * Luego, libera la memoria ocupada por tmp y lo establece en NULL.
    * Si no puede reservar memoria para line, libera la memoria ocupada por line (si se reservó), la memoria ocupada por tmp y establece ambos punteros en NULL.
12. **Función get\_next\_line**:
    * Lee y procesa líneas de un archivo identificado por el descriptor fd.
    * Lee bloques de datos del archivo en un búfer (buf) utilizando read.
    * Acumula los datos leídos en la variable estática tmp.
    * Busca un carácter de nueva línea ('\n') en tmp. Si lo encuentra, crea una línea utilizando create\_string, la devuelve y elimina la parte procesada de tmp utilizando remove\_string.
    * Si no se encuentra un salto de línea y se llega al final del archivo, crea una línea utilizando create\_nullstring y libera la memoria de tmp.
    * La función se ejecuta en un bucle infinito para procesar las líneas hasta que no hay más datos para leer.
13. **Función main**:
    * En la función principal, se abre el archivo "1char.txt" en modo de lectura (O\_RDONLY).
    * Utiliza un bucle while para leer líneas del archivo utilizando get\_next\_line.
    * Imprime cada línea y libera la memoria de cada línea después de usarla.
    * Después de salir del bucle, intenta leer una vez más utilizando get\_next\_line para verificar si se detecta correctamente el final del archivo.
14. **Función get\_next\_line**:
    * La función get\_next\_line está diseñada para leer líneas de un archivo de manera incremental, en lugar de cargar todo el contenido en memoria de una vez.
    * Utiliza un búfer buf para leer datos desde el archivo especificado por su descriptor (fd).
    * Mantiene un puntero estático tmp que acumula el contenido leído en lecturas anteriores.
    * La función busca caracteres de nueva línea ('\n') en el contenido acumulado en tmp. Cuando encuentra uno, crea una nueva línea a partir del contenido antes del salto de línea y devuelve esa línea.
    * Si llega al final del archivo y no hay más líneas en el contenido acumulado, devuelve NULL.
15. **Función create\_string**:
    * La función create\_string se encarga de crear una nueva subcadena de caracteres a partir del contenido en tmp.
    * Esta subcadena va desde el inicio de tmp hasta la primera aparición del carácter c dentro de los primeros n caracteres.
    * Reserva memoria para la subcadena y utiliza ft\_strlcpy para copiar los caracteres relevantes en la nueva cadena.
16. **Función create\_nullstring**:
    * La función create\_nullstring se utiliza para crear una nueva cadena de caracteres a partir del contenido almacenado en tmp.
    * Si tmp no está vacío, se reserva memoria para la nueva cadena y se copia el contenido de tmp.
    * Luego, la memoria ocupada por tmp se libera, y el puntero tmp se establece en NULL.
    * Esto permite extraer y retener el contenido acumulado en tmp en una nueva cadena mientras se libera la memoria original.
17. **Función remove\_string**:
    * La función remove\_string se utiliza para crear una nueva cadena que excluye el contenido hasta la primera aparición de '\n' en tmp.
    * Se reserva memoria para la nueva cadena y se copia el contenido de tmp después de la posición del primer salto de línea.
    * La memoria de tmp se libera, y el puntero se establece en NULL.
    * Esto ayuda a eliminar la parte procesada del contenido acumulado en tmp.
18. **Función main**:
    * En la función principal, se abre el archivo "1char.txt" en modo de lectura (O\_RDONLY).
    * La función get\_next\_line se utiliza en un bucle para leer líneas del archivo.
    * Cada línea se imprime y luego se libera la memoria ocupada por esa línea.