25.- `ft_strjoin.-`

Based on BSD Man Pages: Similar to `strjoin(3)`

Synopsis:

```
char *ft_strjoin(const char *s1, const char *s2);
```

Purpose:libft

- Concatenates (joins) two given strings (`s1` and `s2`) into a single new string.
- Allocates memory for the new string and returns a pointer to it.
- Automatically adds a null terminator (`\0`) to the combined string.

Parameters:

- `s1`: The first string to join.
- `s2`: The second string to join.

Return Value:

> Returns a pointer to the newly allocated and joined string, or `NULL` if memory allocation fails.

Description:

1. **Calculates string lengths:** Uses `ft_strlen` to determine the lengths of both `s1` and `s2`.
2. **Allocates memory:** Calls `malloc` to create enough memory to hold the combined string length plus the null terminator.
3. **Checks for allocation failure:** Returns `NULL` if memory allocation fails.
4. **Copies strings:**
    - Calls `strcpy` to copy `s1` to the beginning of the new string.
    - Calls `strcat` to append `s2` to the end of the string, ensuring the added null terminator.
5. **Returns joined string:** Returns the pointer to the new string.

Code:

```
#include "libft.h"

char *ft_strjoin(const char *s1, const char *s2)
{
    char *new_str;
    size_t len1;
    size_t len2;

    len1 = ft_strlen(s1);
    len2 = ft_strlen(s2);
    new_str = malloc(len1 + len2 + 1);
    if (new_str == NULL) {
        return (NULL); // Allocation failed
    }
    strcpy(new_str, s1);
    strcat(new_str, s2);
    return (new_str);
}
```

Explanation:

1. **Combined length:** The total length needed for the new string is calculated as `len1 + len2 + 1`. The extra `+ 1` is for the null terminator.
2. **Memory allocation:** `malloc` is used to allocate the calculated memory block. If allocation fails, `NULL` is returned.
3. **String copying:**
    - `strcpy`: Efficiently copies `s1` to the beginning of the new string.
    - `strcat`: Appends `s2` to the end of the new string, making sure to add the null terminator.
4. **Return value:** The pointer to the newly joined string is returned.

Key Points:

- Dynamically allocates memory for the combined string using `malloc`.
- Efficiently joins strings using `strcpy` and `strcat`, ensuring null termination.
- Handles memory allocation failures gracefully by returning `NULL`.
- Requires manual mlibftemory deallocation using `free` to avoid memory leaks.

Comments for the `main` Function:

```
int main(void)
{
    char *s1 = "Hello";
    char *s2 = ", world";
    char *new_str;

    new_str = ft_strjoin(s1, s2);
    if (new_str == NULL) {
        printf("Error allocating memory for new string\n");
        return (1);
    }
    printf("Joined string: %s\n", new_str);
    free(new_str); // Remember to free memory
    return (0);
}
```