26.- `ft_strtrim.-`

Not directly based on any BSD man page, but similar to `strsep(3)` and `strtok(3)` with different behavior. Associated library: “libft.h”. (**ft_strtrim: Trim Leading and Trailing Characters.)**

Synopsis:

```
char *ft_strtrim(char const *s1, char const *set);
```

Purpose:

- Creates a new string by removing all leading and trailing characters from a given string (`s1`) that are present in a set of characters (`set`).
- Allocates memory for the trimmed string and returns a pointer to it.

Parameters:

- `s1`: The string to be trimmed.
- `set`: The set of characters to be removed from the beginning and end of `s1`.

Return Value:

- Returns a pointer to the newly allocated trimmed string, or `NULL` if memory allocation fails or invalid parameters are provided.

Description:

1. **Handles invalid inputs:** Checks if `s1` or `set` is `NULL` and returns `NULL` if so.
2. **Trims leading characters:** Iterates through `s1` from the beginning, removing characters that are present in `set`, until a character not in `set` is found.
3. **Trims trailing characters:** Finds the length of the trimmed string (`len`) and iterates from the end, removing characters in `set` until a character not in `set` is found.
4. **Creates trimmed substring:** Uses `ft_substr` to extract the trimmed portion of `s1` based on the adjusted start and length.
5. **Returns trimmed string:** Returns the pointer to the newly allocated trimmed string.

Code:

```
#include "libft.h"

char *ft_strtrim(char const *s1, char const *set)
{
    size_t len;

    if (!s1 || !set)
        return (NULL);
    while (*s1 && ft_strchr(set, *s1))
        s1++;
    len = ft_strlen(s1);
    while (ft_strrchr(set, s1[len]) && len)
        len--;
    return (ft_substr(s1, 0, len + 1));
}
```

Code Explanation:

- The code first checks for invalid inputs and returns `NULL` if necessary.
- It then uses `ft_strchr` to iterate through `s1` and remove leading characters in `set`.
- It calculates the length of the trimmed string using `ft_strlen`.
- It uses `ft_strrchr` to iterate from the end and remove trailing characters in `set`.
- It calls `ft_substr` to create the trimmed substring based on the adjusted start and length.
- It returns the pointer to the newly created trimmed string.

Comments for the `main` Function:

```
int main(void)
{
    char *s1 = "   Hola, mundo   ";
    char *set = " ";
    char *ret;

    ret = ft_strtrim(s1, set);  // Trim leading and trailing spaces
    if (ret == NULL) {
        printf("Error allocating memory for new string\n");
        return (1);
    }
    printf("New string: %s\n", ret);  // Output: "Hola,mundo"
    free(ret);  // Free allocated memory
    return (0);
}
```

Key Points:

- `ft_strtrim` allocates memory for the trimmed string,