|  |  |  |
| --- | --- | --- |
| Assignment Code | : | C.S.P0023 |
| Assignment Name | : | Rewrite some basic functions in C |
| Student Name | : | Le Thi Thanh Nhan |
| Time/Date | : | 1h30,21/11/2019 |

Approach

Strcat: copy all character from source to destination and retun destination.

Strcmp: count length of each string than return length string 1 minus string 2.

Strcpy: copy all character from string 2 to string 1 and put “\0” at the end.

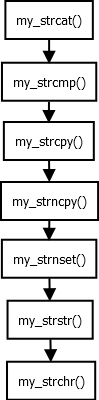
Strncpy: copy all character, index from 0 to num, from string 2 to string 1 and put “\0” at the end.

Strnset: change all character in string, index from 0 to num, to character ‘c’.

Strstr: find substring in string, if found, return all character in string from the substring characters, else return null.

Strchr: find the first character ‘c’ in string, if found, return all character in string from the character, else return null.

Flowchart



Source code

#include <stdio.h>

#include <string.h>

#include <conio.h>

char \*my\_strcat(char \*destination, const char \*source)

{

    char \*ptr = destination + strlen(destination);

    while (\*source != '\0')

        \*ptr++ = \*source++;

    \*ptr = '\0';

    return destination;

}

int my\_strcmp(const char \*X, const char \*Y)

{

    while (\*X)

    {

        if (\*X != \*Y)

            break;

        X++;

        Y++;

    }

    return \*(const unsigned char \*)X - \*(const unsigned char \*)Y;

}

void my\_strcpy(char d[], char s[])

{

    int c = 0;

    while (s[c] != '\0')

    {

        d[c] = s[c];

        c++;

    }

    d[c] = '\0';

}

char \*my\_strncpy(char \*s1, const char \*s2, size\_t num)

{

    if (s1 == NULL)

        return NULL;

    char \*ptr = s1;

    while (\*s2 && num--)

    {

        \*s1 = \*s2;

        s1++;

        s2++;

    }

    \*s1 = '\0';

    return ptr;

}

char \*my\_strstr(char \*str, char \*substr)

{

    while (\*str)

    {

        char \*Begin = str;

        char \*pattern = substr;

        while (\*str && \*pattern && \*str == \*pattern)

        {

            str++;

            pattern++;

        }

        if (!\*pattern)

            return Begin;

        str = Begin + 1;

    }

    return NULL;

}

void \*my\_strnset(char \*str, int c, size\_t num)

{

    int i;

    char \*ptr = str;

    for (i = 0; i < num; i++)

        ptr[i] = c;

    return ptr;

}

char \*my\_strchr(register const char \*s, int c)

{

    do

    {

        if (\*s == c)

        {

            return (char \*)s;

        }

    } while (\*s++);

    return (0);

}

int main()

{

    int i, j;

    char s1[10] = "This is ";

    char s2[20] = "my university";

    printf("s1: %s\ns2: %s\n", s1, s2);

    printf("mystrcat(s1,s2)\n");

    my\_strcat(s1, s2);

    printf("After concatenation: %s", s1);

    printf("\n\*\*\*\*\*\*\n");

    printf("s1: %s\ns2: %s\n", s1, s2);

    printf("mystrcmp(s1,s2)\n");

    int ret = my\_strcmp(s1, s2);

    if (ret > 0)

        printf("s1 is greater than s2");

    else if (ret < 0)

        printf("s2 is less than s1");

    else

        printf("s1 is equal to s2");

    printf("\n\*\*\*\*\*\*\n");

    printf("mystrcpy(s1,s2)\n");

    my\_strcpy(s1, s2);

    printf("s1: %s\ns2: %s\n", s1, s2);

    printf("\n\*\*\*\*\*\*\n");

    printf("mystrncpy(s1,\"I love you\",7)\n");

    my\_strncpy(s1, "I love you", 7);

    printf("s1: %s\n", s1);

    printf("\n\*\*\*\*\*\*\n");

    printf("mystrnset(s1,'\*',3)\n");

    my\_strnset(s1, '\*', 3);

    printf("s1: %s\n", s1);

    printf("\n\*\*\*\*\*\*\n");

    char str1[30] = "My name is Nhan";

    char str2[10] = "Nhan";

    printf("str1: %s\nstr2: %s\n", str1, str2);

    printf("mystrstr(str1,str2)\n");

    printf("substring: %s\n", my\_strstr(str1, str2));

    printf("\n\*\*\*\*\*\*\n");

    printf("mystrchr(str1,'i')\n");

    printf("string after 'i': %s\n", my\_strchr(str1, 'i'));

    getch();

    return 0;

}

Result

