lec8-string-struct

April 11, 2018

1 Lecture 8, String, Structure

Zewei Chu 4/11/2018

1.1 String

- Need to #include<string.h> for string library functions
- strcpy function to copy a string

```
In [2]: #include <stdio.h>
        #include <string.h>

        int main(){
            char myString[20];
            strcpy(myString, "hello");
            printf("%s\n", myString);
            printf("%lu\n", strlen(myString));
            for (int i = 0; i < 5; ++i)
                putchar(myString[i]);
            return 0;
        }
```

```
hello
5
hello
```

write our own strcpy function: strcpy2

```
In [3]: #include <stdio.h>
        #include <string.h>

        void strcpy2(char dest_str[], char source_str[]){
            int i = 0;
            while (source_str[i] != '\0'){
                dest_str[i] = source_str[i];
                i ++;
            }
```

```
        dest_str[i] = '\0';
    }

    int main(){
        char myString[20];
        strcpy2(myString, "hello");
        printf("%s\n", myString);
        printf("%lu\n", strlen(myString));
        for (int i = 0; i < 5; ++i)
            putchar(myString[i]);
        return 0;
    }
```

```
hello
5
hello
```

- strlen to count string length
- exercise: write our own strlen function - strlen2

```
In [5]: #include<stdio.h>
        #include<string.h>

        int strlen2(char str[]){
            int length = 0;
            while (str[length] != '\0')
                length ++;
            return length;
        }

        int main(){
            printf("%lu\n", strlen("hello"));
            printf("%d\n", strlen2("hello"));
            return 0;
        }
```

```
5
5
```

1.1.1 command line string arguments

- argc: number of arguments
- argv: arguments - an array of strings

```
In [6]: #include<stdio.h>

        int main(int argc, char* argv[]){
            printf("%d\n", argc);
```

```
            for (int i = 0; i < argc; ++i){
                printf("%s ", argv[i]);
            }
            printf("\n");
        }
```

```
1
/var/folders/87/k3tmbndn0b77_wxs0bbd_n4h0000gq/T/tmpc7x6jni9.out
```

1.1.2 An exercise of pointers

In [45]:
```
#include<stdio.h>

int main(){
    char* s = "string";
    char* c[] = {"ENTER", "NEW", "POINT", "FIRST"};
    printf("%s\n", c[3]);
    char **cp[]={c+3,c+2,c+1,c};
    char*** cpp = cp;

    for (int i = 0; i < 4; ++i) printf("%s\n", c[i]);
    for (int i = 0; i < 4; ++i) printf("%s\n", *cp[i]);
    printf("\n");

    // challenge
    printf("%s\n",c[1]+1);
    printf("%s%c%c\n",c[3]+3,c[2][2],*((*c)+4));
    printf("\n");

    printf("\n");
    printf("%s\n", *(*(cpp+1)));
    printf("%s\n", *(*(cpp+1) + 1));
    printf("%s\n", **(++cpp));
    printf("%s\n", (*(-- (*(++cpp))))+3);
    printf("%s\n", *(cpp [-2])+3);
    printf("%s\n", cpp[-1][- 1]+1);

}
```

```
FIRST
ENTER
NEW
POINT
FIRST
FIRST
POINT
NEW
```

```
ENTER

EW
STIR


POINT
FIRST
POINT
ER
ST
EW
```

1.2 Struct

A structure is a collecton of one or more variables, possibly of different types, grouped together under a single name for convenient handling.

Now lets define a structure for a student. We want the following information for a student: - char name[] - unsigned int studentID - double scores[]

In [60]:
```c
#include<stdio.h>
#include<string.h>
struct student{
    char name[50];
    unsigned int studentID;
    double scores[13];
};

int main(){
    struct student a, b, c;
    struct student arr[10];
    strcpy(a.name, "aaaaaa");
    a.studentID = 1;
    a.scores[0] = 90;
    a.scores[1] = 80;
    arr[0].studentID = 12;
    arr[1] = a;
    printf("%p %p\n", &(arr[1].name), &(a.name));
    printf("%s %s\n", arr[1].name, a.name);
    return 0;
}
```

```
0x7ffee88f2210 0x7ffee88f28f8
aaaaaa aaaaaa
```

In [51]:
```c
#include<stdio.h>
struct point{
```

```
    int x;
    int y;
};

struct point makepoint(int x, int y){
    struct point temp;
    temp.x = x;
    temp.y = y;
    return temp;
}

int main(){
    struct point p = makepoint(1,2);
    printf("%d %d\n", p.x, p.y);
    struct point p2 = p;
    printf("%d %d\n", p2.x, p2.y);

}
```

```
1 2
1 2
```