<Adv C & App/>

Advanced C Programming And It's Application

String II: concatenate, comparison & search

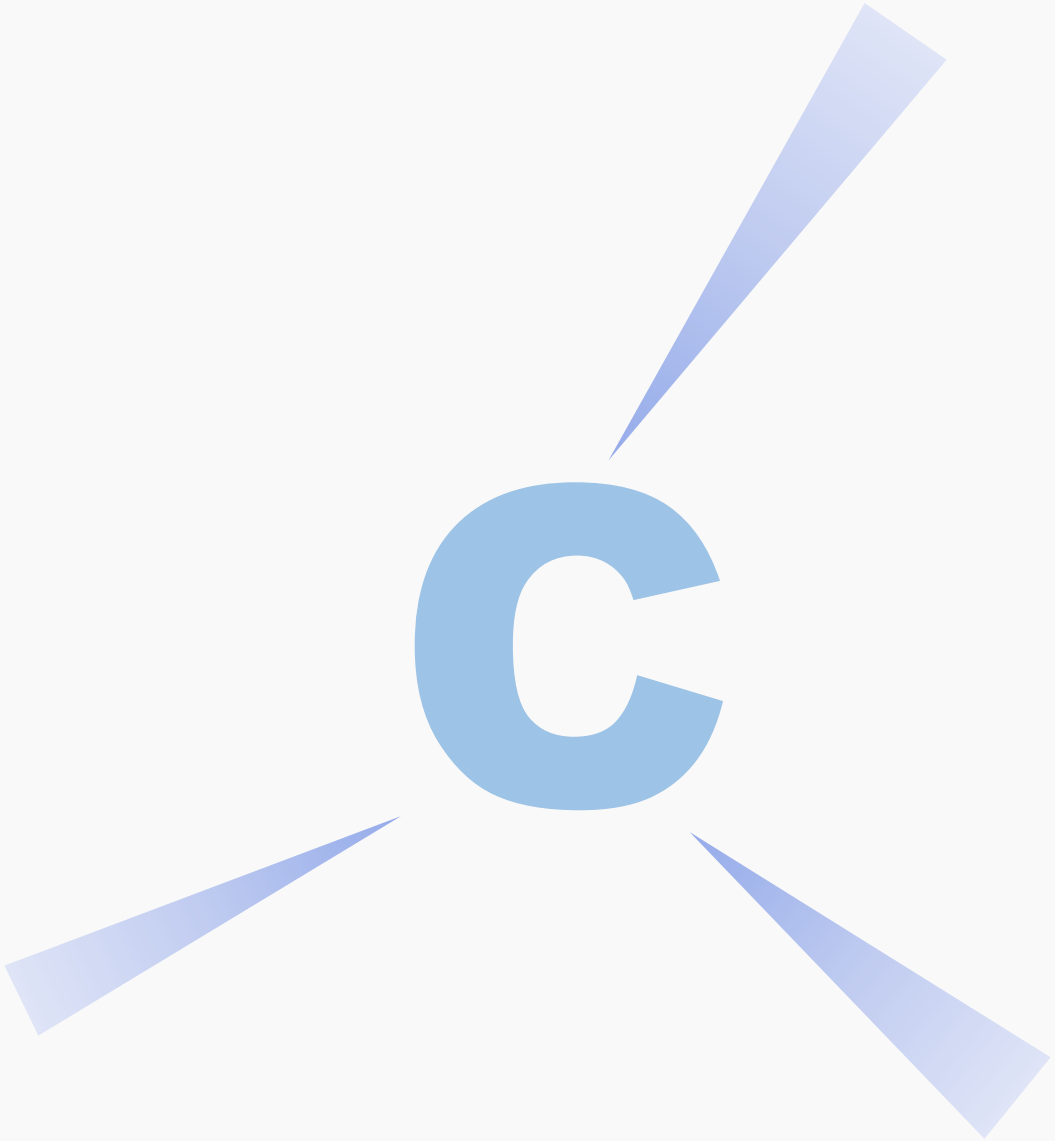
Assistant Prof. Chan, Chun-Hsiang

Department of Artificial Intelligence, Tamkang University

Dec. 15, 2021

</ Adv C & App >

大綱



memset

上週的許多實驗告訴我們一件很重要的事情，你必須要精確地知道你每個變數到底會存多長的字串。

然而我們很難一開始就知道這件事，但有時候我們希望利用一些方法讓資料看起來有去個資的感覺，那該怎麼辦呢?

這裡我們介紹一個有趣的函數 – 叫做memset。

可以複製特定長度的某個字元，並指向一個字元陣列。

*可自行延伸閱讀memcpy()

memset

```
/*Ex 11-1: String Basic */
/* String - memset*/
char s[50];
printf("Ex 11-1: String - memset\n");
strcpy(s, "Do you want to build a snow man?");
printf("string before: %s\n", s);

memset(s, '*', 10);
printf("string after: %s\n", s);
```

```
Ex 11-1: String - memset
string before: Do you want to build a snow man?
string after: **********t to build a snow man?
```

strcat & strncat

Cat 是concatenate的縮寫，因為這個字太長了，所以在Matlab中就是直接簡寫為cat；python中還是用全名。Whatever，我們今天要介紹的cat是在C的string函數庫的函數 – strcat()，可以用來合併不同字串。

既然有了strcat()，就也會有另一個函數叫做strncat()。

他多了一個引數，可以控制需要被合併的字元。

strcat

```
/*Ex 11-2: String Basic */
/* String - Cat*/
char c1[50];
char c2[50];

printf("Ex 11-2: String - Cat\n");
strcpy(c1, "Hello");
strcpy(c2, " world!");
printf("c1 string: %s \nc2 string: %s\n", c1,  c2);
strcat(c1, c2);
printf("Answer: %s\n", c1);
printf("%d %d\n", sizeof(c1), strlen(c1));
```

```
Ex 11-2: String - Cat
c1 string: Hello
c2 string:  world!
Answer: Hello world!
c1 size: 50; c1 length: 12
```

strncat

```
/*Ex 11-3: String Basic */
/* String - Cat2*/
char c1[50];
char c2[50];

printf("Ex 11-3: String – Cat2\n");
strcpy(c1, "Hello");
strcpy(c2, " world!");
printf("c1 string: %s \nc2 string: %s\n", c1,  c2);
strncat(c1, c2, 6);
printf("Answer: %s\n", c1);
printf("%d %d\n", sizeof(c1), strlen(c1));
```

```
Ex 11-3: String - Cat2
c1 string: Hello
c2 string:  world!
Answer: Hello world
c1 size: 50; c1 length: 11
```

Concatenate string with memset

Lab 11-1:

宣告三個字元陣列，分別儲存三個字串:

(1) I have an apple

(2) , I have a pen

(3) , uh! Apple pen!

利用memset的方式創造一個可以剛好裝得下(1-3)字串的空間。

再將三個字串與前面的memset的空字串cat在一起。

最後在printf印出來。結果與程式碼一起截圖到word檔中，程式碼另外壓縮上傳。

***Hint: memset(字串變數, '\0', 長度);**

String Comparison

字串比對是一個很重要的工具。舉凡我們平常在電腦中做檔案檢索等，都是要利用到字串比對的功能。再這裡我們提到的字串比對功能相對簡單，就是要一模一樣的才會被抓出來。

利用Ex 11-4, Ex 11-5, Ex 11-6等三個練習，來找出strcmp的return值，意義為何?

String Comparison

```
/*Ex 11-4: String Basic */
/* String - strcmp*/
char s1[4] = "Wow";
char s2[4] = "Wow";
int res = strcmp(s1, s2);
printf("Ex 11-4: String - strcmp\n");
printf("Result: %d \n", res);
if (res==0){
        printf("Two strings are equal!\n");
}else{
        printf("Two strings are different!\n");
}
```

```
Ex 11-4: String - strcmp
Result: 0
Two strings are equal!
```

```
/*Ex 11-5: String Basic */
/* String – strcmp2*/
char s1[4] = "wow";
char s2[4] = "Wow";
int res = strcmp(s1, s2);
printf("Ex 11-5: String – strcmp2\n");
printf("Result: %d \n", res);
if (res==0){
        printf("Two strings are equal!\n");
}else{
        printf("Two strings are different!\n");
}
```

```
Ex 11-5: String - strcmp2
Result: 1
Two strings are different!
```

String Comparison

```
/*Ex 11-6: String Basic */
/* String - strcmp3*/
char s1[4] = "Wow";
char s2[4] = "wow";
int res = strcmp(s1, s2);
printf("Ex 11-6: String - strcmp3\n");
printf("Result: %d \n", res);
if (res==0){
        printf("Two strings are equal!\n");
}else{
        printf("Two strings are different!\n");
}
```

Lab 11-2:

利用這三個範例嘗試說明res回傳為正/負數的時候，其代表的意義為何？ 請利用word檔說明。

***Hint: 注意字串每個字元的ASCII大小。**

```
Ex 11-6: String - strcmp3
Result: -1
Two strings are different!
```

puts and fgets

putchar	印出指定的字元。
getchar	取得第一個字元。
puts	印出指定的變數、字元或字串。
fgets	取得指定的字串。

Lab 11-3:

puts與printf的差異在哪? 請寫在word檔中。

puts and fgets

```
/*Ex 11-7: String Basic */
/* String - fgets, puts*/
char s4fg[50];

printf("Ex 11-7: String - fgets, puts\n");
printf("Plz enter a word: \n");
fgets(s4fg, sizeof(s4fg), stdin); //fgets has auto newline functionality

puts(s4fg);
```

String Search

strstr	比較兩個字串，尋找第二個字串**完整**出現在第一個字串的位置。
strspn	比較兩個字串，從第二個字串第一個字元開始相同字元總字元數
strcspn	比較兩個字串，找出兩字串有共同字元的第一個位置 ≠ 第二個字串完整的出現在第一個字串中。
strpbrk	比較兩個字串，找出第二字串中出現在第一個字串中的第一個字元。

strstr

```
#define LENGTH 80
/*Ex 11-8: String Basic */
/* String - strstr*/
char str1[LENGTH];
char str2[LENGTH];
printf("Ex 11-8: String - strstr\n");
printf("input string: ");
fgets(str1, LENGTH, stdin);
printf("string for searching: ");
fgets(str2, LENGTH, stdin);
// delete the last character (newline char)
str2[strlen(str2) - 1] = '\0';
char* loc = strstr(str1, str2);
if (loc == NULL) {
        printf("The second string cannot be
        found in the first one!\n");
} else {
        printf("The second string completely
        appears in the location %lu of the
        first one!\n", loc - str1);
}
```

strstr

Lab 11-4:

試試看用Ex 11-8的程式碼測試:

1st string: 123456 //第一個字串統一使用這個

2nd string: 123, 456, 654, 000 //第二個字串分別利用這四組做測試

請說明輸出結果的差異，請說明在word檔中。

strspn

```
#define LENGTH 80
```

```
/*Ex 11-9: String Basic */
/* String - strspn*/
char str1[LENGTH];
char str2[LENGTH];
printf("Ex 11-9: String - strspn\n");
printf("Plz enter a string: ");
fgets(str1, LENGTH, stdin);
printf("Plz enter a string for searching: ");
fgets(str2, LENGTH, stdin);
```

```
// delete the last character (newline char)
str1[strlen(str1) - 1] = '\0';
str2[strlen(str2) - 1] = '\0';
size_t loc = strspn(str1, str2);
if (loc == strlen(str1)) {
        printf("The second string completely
        appears in the first one!\n");
} else {
        printf("The different characters start
        from %lu\n", loc);
}
```

strspn

Lab 11-5:

試試看用Ex 11-9的程式碼測試:

1st string: 123456789 //第一個字串統一使用這個

2nd string: 321, 456, 123456789, 12123 //第二個字串分別利用這四組做測試

請說明輸出結果的差異，請說明在word檔中。

strcspn

```
#define LENGTH 80
```

```
/*Ex 11-10: String Basic */
/* String - strcspn*/
char str1[LENGTH];
char str2[LENGTH];
printf("Ex 11-10: String - strcspn\n");
printf("Plz enter a string: ");
fgets(str1, LENGTH, stdin);
printf("Plz enter a string for searching: ");
fgets(str2, LENGTH, stdin);
```

```
// delete the last character (newline char)
str1[strlen(str1) - 1] = '\0';
str2[strlen(str2) - 1] = '\0';
size_t loc = strcspn(str1, str2);
if (loc == strlen(str1)) {
        printf("For all characters in the
        second string cannot find in the first
        one! (loc value: %lu)\n", loc);
} else {
        printf("At least one character in the
        second string is found in the location
        %lu of the first string.\n", loc);
}
```

strcspn

Lab 11-6:

試試看用Ex 11-10的程式碼測試:

1st string: 123456789 //第一個字串統一使用這個

2nd string: 123, 004, 123456789, 000 //第二個字串分別利用這四組做測試

請說明輸出結果的差異，請說明在word檔中。

strpbrk

```
#define LENGTH 80
```

```
/*Ex 11-11: String Basic */
/* String - strpbrk*/
char str1[LENGTH];
char str2[LENGTH];
printf("Ex 11-11: String - strpbrk\n");
printf("Plz enter a string: ");
fgets(str1, LENGTH, stdin);
printf("Plz enter a string for searching: ");
fgets(str2, LENGTH, stdin);
char *ret;
```

```
// delete the last character (newline char)
printf("%s %d--", str1, strlen(str1));
str1[strlen(str1) - 1] = '\0';
str2[strlen(str2) - 1] = '\0';
printf("%s %d--", str1, strlen(str1));
ret = strpbrk(str1, str2);
printf("%s %d--", str1, strlen(str1));
if(ret) {
        printf("First matching character:
%c\n", *ret);
} else {
        printf("Character not found\n");
}
```

strpbrk

Lab 11-7:

試試看用Ex 11-11的程式碼測試:

1st string: abcde //第一個字串統一使用這個

2nd string: cba, 000, 123gcd45

//第二個字串分別利用這三組做測試

請說明輸出結果的差異，請說明在word檔中。

作業一

請整理出strstr, strspn, strcspn, 與 strpbrk之間的差異。
分別設計四個程式與測試資料，以舉例說明其差異為何?

Functions	Input arguments	Output arguments	Situations		
			完全相同	部分不同	完全不同
strstr	…	…	…	…	…
strspn	…	…	…	…	…
			…	…	…
strcspn	…	…	…	…	…
			…	…	…
strpbrk	…	…	…	…	…
			…	…	…

References

https://openhome.cc/Gossip/CGossip/index.html
https://edisonx.pixnet.net/blog/post/35305668
https://www.learn-c.org/
http://tw.gitbook.net/cprogramming/
https://blog.techbridge.cc/2020/05/03/simple-c-language-introduction-tutorial/
https://openhome.cc/Gossip/CGossip/StringLengthCopyCat.html
https://www.huaweicloud.com/articles/12640716.html
https://skylinelimit.blogspot.com/2018/02/c-2.html
https://www.learn-c.org/en/Strings