

Họ và tên: Nguyễn Trọng Tuệ

Mã số sinh viên: 20194710

Mã lớp: 130935

BÁO CÁO MINI PROJECT GIỮA KỲ

1. Nội dung bài toán

- Với 2 chuỗi ký tự đưa ra, hãy tìm số ký tự chung giữa chúng
- Ví dụ: Với chuỗi $s1 = \text{"aabcc"}$, chuỗi $s2 = \text{"adcaa"}$, kết quả đầu ra sẽ là $\text{commonCharacterCount}(s1, s2) = 3$. Hai chuỗi ký tự trên có 3 ký tự chung – 2 ký tự ‘a’ và 1 ký tự ‘c’.

2. Ý tưởng giải quyết bài toán

- Với từng phần tử ký tự nằm trong mảng chứa chuỗi ký tự $s1$, thực hiện so sánh lần lượt với mỗi ký tự trong mảng chứa chuỗi ký tự $s2$. Khởi tạo một biến đếm số ký tự trùng nhau với giá trị khởi tạo là 0.
 - Nếu ký tự đang xét của $s1$ không trùng với ký tự đang duyệt của $s2$, thực hiện duyệt tiếp $s2$ cho đến khi gặp ký tự trùng hoặc đến khi kết thúc $s2$. Nếu ký tự của $s1$ không trùng với bất cứ ký tự nào trong $s2$ thì chuyển sang duyệt ký tự tiếp theo của $s1$ với chuỗi $s2$.
 - Nếu ký tự đang xét của $s1$ không trùng với ký tự đang duyệt của $s2$, tại địa chỉ của ký tự đó, gán giá trị ‘ ‘ (ký tự khoảng trắng tương đương 32 trong bảng mã ASCII và không tính là ký tự có thể trùng) vào địa chỉ đó. Sau đó, biến đếm số ký tự trùng nhau tăng thêm 1 đơn vị và chuyển sang duyệt ký tự tiếp theo của $s1$ với chuỗi $s2$.
- Khi duyệt hết chuỗi $s1$, ta sẽ thu được giá trị đầu ra là số ký tự trùng nhau qua biến đếm được khởi tạo từ đầu chương trình và được tăng thêm một mỗi khi có ký tự trùng nhau giữa 2 dãy.

3. Mô phỏng giải quyết bài toán bằng ngôn ngữ C

```
CommonCharacterCount.c U X
CommonCharacterCount.c > ...
1  #include<stdio.h>
2  #include<string.h>
3
4  int commonCharacterCount(char x[], char y[]) {
5      int count = 0;
6      for (int i = 0; i < strlen(x); i++) {
7          for (int j = 0; j < strlen(y); j++) {
8              if (x[i] == y[j] && x[i] != ' ' && y[j] != ' ') {
9                  count++;
10                 y[j] = ' ';
11                 break;
12             }
13         }
14     }
15     return count;
16 }
17
18 int main() {
19     char s1[] = "aabcc";
20     char s2[] = "adcaa";
21     printf("Chuoi s1 la: %s\n", s1);
22     printf("Chuoi s2 la: %s\n", s2);
23     printf("commonCharacterCount(s1, s2) = %d\n", commonCharacterCount(s1, s2));
24 }
```

- Kết quả khi chạy chương trình:

```
PROBLEMS OUTPUT TERMINAL COMMENTS DEBUG CONSOLE
PS D:\Cybernetic\IT3322] - CTD> gcc .\CommonCharacterCount.c -o .\CommonCharacterCount
PS D:\Cybernetic\IT3322] - CTD> .\CommonCharacterCount.exe
Chuoi s1 la: aabcc
Chuoi s2 la: adcaa
commonCharacterCount(s1, s2) = 3
PS D:\Cybernetic\IT3322] - CTD> █
```

- Dữ liệu đầu vào:
 - Mảng chuỗi ký tự s1[] = “aabcc”.
 - Mảng chuỗi ký tự s2[] = “adcaa”.
- Hàm xử lý yêu cầu bài toán: int commonCharacterCount(char x[], y[]);
 - s1 và s2 được đưa vào trong hàm để xử lý thông qua 2 biến kiểu chuỗi ký tự với x := s1, y := s2. Trong phạm vi nội bộ hàm, ta sẽ sử dụng x và y thay cho s1 và s2.
 - Tại đây, ta khởi tạo một biến đếm int count = 0, tương đương với số ký tự trùng nhau giữa 2 chuỗi được truyền vào hàm tại lúc khởi tạo biến đếm này là 0.
 - Thực hiện ý tưởng duyệt chuỗi đã được nêu ra ở phần ý tưởng giải quyết bài toán ở trên với 2 vòng lặp for lồng nhau, vòng lặp ngoài với

biến *i* được khởi tạo để duyệt chuỗi *x*, vòng lặp trong với biến *j* được khởi tạo để duyệt chuỗi *y*. Lưu ý ở đây, ta không xét ký tự khoảng trắng ' ' (32 = 0x20), nên có thể sử dụng để thay thế các ký tự đã được duyệt là trùng nhau giữa 2 chuỗi để gán vào địa chỉ của ký tự trùng *y[j]* trong chuỗi *y*.

- Nếu tìm thấy ký tự trùng giữa 2 chuỗi, tăng biến đếm thêm 1 và thực hiện dừng vòng lặp trong bằng câu lệnh `break`, tiếp tục thực hiện từ vòng lặp ngoài để duyệt ký tự tiếp theo của *x* cho tới khi kết thúc chuỗi *x* (`strlen` được sử dụng để tính độ dài của chuỗi, thuận tiện cho việc duyệt chuỗi ký tự).
- Cuối cùng, giá trị được trả về của hàm chính là biến đếm `count` tại câu lệnh `return count`; sau khi đã kết thúc việc duyệt chuỗi. Hàm lúc này chứa giá trị trả về chính là số ký tự trùng nhau giữa 2 chuỗi đã được truyền vào *s1* và *s2*.
- Thực hiện các thao tác còn lại để hiển thị kết quả ra màn hình.
- Lưu ý: nếu thực hiện hàm với 2 string buffers thì chuỗi sẽ kết thúc tại ký tự '\n' xuống dòng (tại thời điểm ấn phím Enter) thay vì ký tự '\0', cần lưu ý điều này

```
CommonCharacterCount.c U  CommonCharacterCount_Buffer.c U X
CommonCharacterCount_Buffer.c > main()
1  #include<stdio.h>
2  #include<string.h>
3
4  int commonCharacterCount(char x[], char y[]) {
5      int count = 0;
6      for (int i = 0; i < strlen(x); i++) {
7          for (int j = 0; j < strlen(y); j++) {
8              if (x[i] == y[j] && x[i] != ' ' && y[j] != ' ') {
9                  count++;
10                 y[j] = ' ';
11                 break;
12             }
13         }
14     }
15     return count;
16 }
17
18 int main() {
19     char s1[100];
20     char s2[100];
21     printf("Nhap chuoi s1: \n");
22     fgets(s1, 100, stdin);
23     printf("Nhap chuoi s2: \n");
24     fgets(s2, 100, stdin);
25     printf("Chuoi s1 la: %s\n", s1);
26     printf("Chuoi s2 la: %s\n", s2);
27     printf("commonCharacterCount(s1, s2) = %d\n", commonCharacterCount(s1, s2));
28     printf("%d\t%d\n", strlen(s1), strlen(s2));
29 }
```

```

PS D:\Cybernetic\IT3322] - CTD> gcc .\CommonCharacterCount_Buffer.c -o .\CommonCharacterCount_Buffer
PS D:\Cybernetic\IT3322] - CTD> .\CommonCharacterCount_Buffer.exe
Nhap chuoì s1:
aabcc
Nhap chuoì s2:
adcaa
Chuoì s1 la: aabcc

Chuoì s2 la: adcaa

commonCharacterCount(s1, s2) = 4
6      6

```

4. Thực hiện giải quyết bài toán bằng ngôn ngữ Assembly

- a) Code thực thi (nhập 2 string buffers s1 và s2 từ bàn phím)
- Lúc này, chuỗi kết thúc tại ký tự '\n' = 10 = 0x0a

```

1  #Mid-term Mini Project, Task 14
2  #Nguyen Trong Tue - 20194710
3  .data
4      s1:      .space    100
5      s2:      .space    100
6      mes_1:   .asciiz   "Enter the first string s1: "
7      mes_2:   .asciiz   "Enter the second string s2: "
8      mes_out: .asciiz   "commonCharacterCount(s1, s2) = "
9      mes_err: .asciiz   "Invalid string input!"
10 .text
11 main:
12 str1_input:
13     li    $v0, 54
14     la    $a0, mes_1
15     la    $a1, s1
16     la    $a2, 100
17     syscall
18     beq   $a1, $0, str1_copy      #check status value if != 0 (OK)
19     li    $v0, 59
20     la    $a0, mes_err
21     syscall
22     j     exit
23 str1_copy:
24     la    $s0, s1
25 str2_input:

```

```

26  li    $v0, 54
27  la    $a0, mes_2
28  la    $a1, s2
29  la    $a2, 100
30  syscall
31  beq    $a1, $0, str2_copy      #check status value if != 0 (OK)
32  li    $v0, 59
33  la    $a0, mes_err
34  syscall
35  j      exit
36 str2_copy:
37  la    $s1, s2
38 function_calling:
39  j      commonCharacterCount
40 commonCharacterCount:
41  addi   $t0, $0, 0              #$t0 = count = init 0
42  addi   $t1, $0, 0              #$t1 = i = init -1
43  addi   $t2, $0, 0              #$t2 = j = init -1
44 L1:
45  add    $t3, $s0, $t1           #$t3 = $s0 + $t1 = x[0] + i = x[i]
46  lb     $t4, 0($t3)             #$t4 = value at $t3 = x[i]
47  beq    $t4, 0x20, L2_end       #check whether x[i] == space or not, if equals => skip to the next character of x
48  beq    $t4, 0x0a, end          #check whether x[i] == '\n' or not, if equals => end of x <=> end of function
49  j      L2
50 L2:

51  add    $t5, $s1, $t2           #$t5 = $s1 + $t2 = y[0] + j = x[j]
52  lb     $t6, 0($t5)             #$t6 = value at $t4 = y[j]
53  beq    $t6, 0x0a, L2_end       #check whether y[j] = '\n' or not => end of L2
54  bne    $t4, $t6, L2_next       #if x[i] != y[j] => move to L2_next to continue the inner loop, else
55 common_character_found:
56  addi   $t0, $t0, 1             #count += 1 <=> common character found
57  addi   $t1, $t1, 1             #i += 1 to traverse the next character of x
58  li     $t7, 32                 #$t7 = ' ' = 0x20 = 32
59  sb     $t7, 0($t5)             #y[i] = $t7 = 0x20 = 32 = ' ' = space character
60  addi   $t2, $0, 0              #j = 0 to traverse y with the next character of x
61  j      L1                      #jump to L1 to continue
62 L2_next:
63  addi   $t2, $t2, 1             #j += 1 to traverse the next character of y
64  j      L2
65 L2_end:
66  addi   $t1, $t1, 1             #i += 1 to traverse the next character of x
67  addi   $t2, $0, 0              #j = 0 to traverse y with the next character of x
68  j      L1                      #jump to L1 to continue
69 end:
70  li    $v0, 4
71  la    $a0, mes_out
72  syscall
73  li    $v0, 1
74  la    $a0, 0($t0)
75  syscall
76 exit:

```

- Kết quả:

Input

?

Enter the first string s1:

aabcc

OK

Cancel

Input

?

Enter the second string s2:

adcaa

OK

Cancel

Mars Messages

Run I/O

Clear

```
commonCharacterCount(s1, s2) = 3
-- program is finished running (dropped off bottom) --
```

s1	0x10010000
s2	0x10010064

Address	Value (+0)	Value (+4)
0x10010000	c b a a	\0 \0 \n

Address	Value (+0)	Value (+4)	Value (+8)
0x10010060	\0 \0 \0 \0	d	\0 \0 \n a

- b) Code thực thi (thực hiện với 2 string literals cho trước)

- Lúc này, chuỗi kết thúc tại ký tự ‘\0’ = 0 = 0x0

```

1 #Mid-term Mini Project, Task 14
2 #Nguyen Trong Tue - 20194710
3 .data
4     s1:      .asciiz    "aabcc"
5     s2:      .asciiz    "adcaa"
6     mes_out: .asciiz    "commonCharacterCount(s1, s2) = "
7 .text
8 main:
9     la    $s0, s1
10    la    $s1, s2
11    j      commonCharacterCount
12 commonCharacterCount:
13    addi $t0, $0, 0           #$t0 = count = init 0
14    addi $t1, $0, 0           #$t1 = i = init -1
15    addi $t2, $0, 0           #$t2 = j = init -1
16 L1:
17    add $t3, $s0, $t1         #$t3 = $s0 + $t1 = x[0] + i = x[i]
18    lb  $t4, 0($t3)           #$t4 = value at $t3 = x[i]
19    beq $t4, 0x20, L2_end     #check whether x[i] == space or not, if equals => skip to the next character of x
20    beq $t4, 0x0, end         #check whether x[i] == '\0' or not, if equals => end of x <=> end of function
21    j      L2
22 L2:
23    add $t5, $s1, $t2         #$t5 = $s1 + $t2 = y[0] + j = y[j]
24    lb  $t6, 0($t5)           #$t6 = value at $t4 = y[j]
25    beq $t6, 0x0, L2_end     #check whether y[j] = '\0' or not => end of L2
26    bne $t4, $t6, L2_next    #if x[i] != y[j] => move to L2_next to continue the inner loop, else
27 common_character_found:
28    addi $t0, $t0, 1           #count += 1 <=> common character found
29    addi $t1, $t1, 1           #i += 1 to traverse the next character of x
30    li   $t7, 32              #$t7 = ' ' = 0x20 = 32
31    sb   $t7, 0($t5)          #y[j] = $t7 = 0x20 = 32 = ' ' = space character
32    addi $t2, $0, 0           #j = 0 to traverse y with the next character of x
33    j      L1                 #jump to L1 to continue
34 L2_next:
35    addi $t2, $t2, 1           #j += 1 to traverse the next character of y
36    j      L2
37 L2_end:
38    addi $t1, $t1, 1           #i += 1 to traverse the next character of x
39    addi $t2, $0, 0           #j = 0 to traverse y with the next character of x
40    j      L1                 #jump to L1 to continue
41 end:
42    li   $v0, 4
43    la   $a0, mes_out
44    syscall
45    li   $v0, 1
46    la   $a0, 0($t0)
47    syscall
48 exit:

```

- Kết quả:

s1	0x10010000
s2	0x10010006

Address	Value (+0)	Value (+4)	Value (+8)
0x10010000	c b a a	d \0 c	\0 a

Mars Messages

Run I/O

Clear

```
commonCharacterCount(s1, s2) = 3
-- program is finished running (dropped off bottom) --
```