



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σχεδιασμός Ενσωματωμένων Συστημάτων

9ο εξάμηνο

2η Εργαστηριακή Άσκηση

Βάρδια 1 Ομάδα 14

Πάυλος Καπούτσης 03110080
Παναγιώτης Μπουγουλιάς 03112025

Ερώτημα 1ο: Μετατροπή εισόδου από τερματικό.

Για το συγκεκριμένο ερώτημα δημιουργήθηκαν δύο αρχεία assembly ARM , το πρώτο (ask2.1-2017.s) που αποτελεί την συνάρτηση main απ' όπου αρχίζει και η εκτέλεση του προγράμματος, και το δεύτερο (ask2.1-2017-function.s) που περιλαμβάνει την συνάρτηση μετατροπής των γραμμών.

Το κυρίως πρόγραμμα είναι συνεχούς λειτουργίας και περιλαμβάνει ένα βασικό loop που διαβάζει από την είσοδο και εμφανίζει στην έξοδο, μέχρι να διαβαστεί 'q' ή 'Q' όπου και σταματάει η λειτουργία του προγράμματος.

Η λογική του δεύτερου προγράμματος που παίρνει ως είσοδο ένα string και το μετατρέπει σύμφωνα με τις απαιτήσεις του προβλήματος. Συγκεκριμένα, αν ο χαρακτήρας είναι:

- lowercase letter τότε το μετατρέπουμε σε uppercase letter αφαιρώντας του τον αριθμό 32 σύμφωνα με τον πίνακα ascii
- uppercase letter τότε το μετατρέπουμε σε lowercase letter προσθέτοντας τον 32
- αριθμός μεταξύ του 0 και του 4, προσθέτουμε τον αριθμό 5
- αριθμός μεταξύ του 5 και του 9, αφαιρούμε τον αριθμό 5

Για το διάβασμα της εισόδου και την εμφάνιση της εξόδου καθώς για το χειρισμό των strings χρησιμοποιήθηκαν οι συναρτήσεις της C: scanf, printf, strcpy, strlen. Ένα στιγμιότυπο της εκτέλεσης του προγράμματος φαίνεται παρακάτω:

```
root@debian-armhf:~/test/2.1# ./ask2.1
Input a string up to 32 chars long: kalispera 123 $^&*
KALISPERA 678 $^&*
Input a string up to 32 chars long: TiKanetePwsEisteKaiTaLoipaWRAIA
tIKANETEPWSeISTEKAIItALOIPAwraia
Input a string up to 32 chars long: kajflasdjflkasdjlkjkl123445930jvkljdkjll
KAJFLASDJFLKASDJLKJKL678990485JVK
Input a string up to 32 chars long: qQ#####()_+!
Qq#####()_+!
Input a string up to 32 chars long: q
Exiting.
root@debian-armhf:~/test/2.1#
```

Ιδιαίτερη έμφαση δόθηκε στον χειρισμό των whitespace characters με σκοπό η scanf να κρατάει όλο το line , χρησιμοποιώντας το κατάλληλο regex κατά την κλήση της, όπως φαίνεται και στο πρώτο παράδειγμα στην εικόνα.

Στο τρίτο παράδειγμα στην εικόνα φαίνεται ότι το πρόγραμμα αγνοεί στους περισσευούμενους χαρακτήρες και κάνει transform τους πρώτους 32.

Ερώτημα 2ο: Μετατροπή εισόδου από τερματικό σε αρχείο.

Σκοπός της άσκησης είναι η μέτρηση της συχνότητας εμφάνισης των χαρακτήρων σε μια συμβολοσειρά που εισάγει ο χρήστης στο τερματικό και η εκτύπωσή τους σε αρχείο με τη μορφή << χαρακτήρας -> συχνότητα<newline> >>, όπου ο κάθε χαρακτήρας εμφανίζεται στο αρχείο μια φορά ταξινομημένος με τη σειρά εμφάνισής του στη συμβολοσειρά.

Συγκεκριμένα χρησιμοποιήθηκε ένας πίνακας 94 bytes, από το 33_{10} - 126_{10} σε ascii κωδικούς, όπου κάθε φορά που εμφανιζόταν κάποιος η αντίστοιχη τιμή της θέσης (`ascii_code - 33`) του πίνακα να αυξάνεται κατά 1. Όταν κάποιος χαρακτήρας εκτυπωνόταν στο αρχείο, μηδενιζόταν η συχνότητα ώστε να μην εκτυπωθεί ξανά. Ο αλγόριθμος ήταν γραμμικός στο μήκος της εισόδου και συγκεκριμένα χρειαζόταν να διασχίσουμε μία φορά τη συμβολοσειρά ώστε να κατασκευάσουμε τον πίνακα συχνοτήτων και άλλη μια φορά για εκτυπώσουμε τη ζητούμενη έξοδο στο αρχείο.

Ακολουθεί στιγμιότυπο χρήσης του προγράμματος:

```
root@debian-armhf:~# ./ask2.2
Input a string up to 32 chars long: Hello there! Nice to meet you!
Input a string up to 32 chars long: HELLO BKJBKDJKBKJBK
Input a string up to 32 chars long: 2536827352 bajkkjw
Input a string up to 32 chars long: panagiotis 28396823023901271074
Input a string up to 32 chars long: QUIT
Input a string up to 32 chars long: quit
Input a string up to 32 chars long: q
```

Και η αντίστοιχη έξοδος(`cat count.txt`):

```
H -> 1
e -> 6
l -> 2
o -> 3
t -> 3
h -> 1
r -> 1
! -> 2
N -> 1
i -> 1
c -> 1
m -> 1
y -> 1
u -> 1

H -> 1
E -> 1
L -> 2
O -> 1
B -> 4
K -> 5
J -> 3
D -> 1
```

```
2 -> 3
5 -> 2
3 -> 2
6 -> 1
8 -> 1
7 -> 1
b -> 1
a -> 1
j -> 2
k -> 2
w -> 1

p -> 1
a -> 2
n -> 1
g -> 1
i -> 2
o -> 1
t -> 1
s -> 1
2 -> 4
8 -> 2
3 -> 3
9 -> 2
6 -> 1
0 -> 3
1 -> 2
7 -> 2
4 -> 1
```

```
Q -> 1
U -> 1
I -> 1
T -> 1

q -> 1
u -> 1
i -> 1
t -> 1
```

```
Exiting...
```

Ερώτημα 3ο: Σύνδεση κώδικα C με κώδικα assembly του επεξεργαστή ARM.

Σε αυτή την άσκηση έπρεπε να υλοποιήσουμε τι βασικές συναρτήσεις για χειρισμό strings της C. Συγκεκριμένα φτιάχτηκαν 4 αρχεία ARM assembly: *strlen.s* , *strcpy.s*, *strcat.s* και *strcmp.s*. Στον κώδικα *string_manipulation.c* απλά προσθέσαμε τις δηλώσεις :

```
extern int strlen(char *s);  
extern char *strcpy(char *dest, char *src);  
extern int strcmp(char *s1, char *s2);  
extern char *strcat(char *dest, char *src);
```

έτσι ώστε να μπορέσει ο linker να συνδέσει τα assembly αρχεία με τον κώδικα C.

Χαρακτηριστικό σημείο στην υλοποίηση είναι η σύμβαση για την 1-1 αντιστοιχία καταχωρητών και παραμέτρων της συνάρτησης. Για παράδειγμα στην *strcat.s*, θα βρούμε την τιμή του *dest* στον *r0* και την τιμή του *src* στον *r1*, και αντίστοιχα εμείς θα αποθηκεύουμε την τιμή που επιστρέφουμε στον *r0*.

Η λογική στην συνάρτηση *strlen* είναι: προσπέλαση της συμβολοσειράς μέχρι τον null terminating character '\0' , κρατώντας σε έναν counter το μήκος της.

Στην *strcpy*: προσπέλαση της συμβολοσειράς *src* χαρακτήρα-χαρακτήρα και αποθήκευση στην destination. Επιστρέφουμε στον *r0* την διεύθυνση του *dst*.

Στην *strcmp*: προσπέλαση και των δύο συμβολοσειρών. Αν σε κάποια θέση οι 2 συμβολοσειρές διαφέρουν τότε εξετάζουμε ποια από τις δύο είναι μεγαλύτερη με βάση την εξής συνάρτηση διάταξης:

Μια συμβολοσειρά *a* είναι μεγαλύτερη από μια συμβολοσειρά *b* αν στον πρώτο χαρακτήρα που διαφέρουν ο *ascii* κωδικός της *a* είναι μεγαλύτερος από τον *ascii* κωδικό της *b*.

Επιστρέφουμε 1 αν η πρώτη συμβολοσειρά είναι μεγαλύτερη από τη δεύτερη, -1 αν είναι μικρότερη , και 0 αν είναι ίσες.

Στην πραγματικότητα, η *strcmp* της C επιστρέφει αντί για 1 και -1 την θετική διαφορά και αντίστοιχα την αρνητική διαφορά στους *ascii* κωδικούς των πρώτων χαρακτήρων που διαφέρουν. Σπάνια χρησιμοποιείται αυτή η πληροφορία στην πράξη, κυρίως μας ενδιαφέρει το πρόσημο ή αν είναι 0 γι' αυτό και δεν κρίθηκε αναγκαίο να το συμπεριλάβουμε.

Κάποια screenshots για το build και την εκτέλεση φαίνονται παρακάτω:

```
root@debian-armhf:~/test/2.3# rm string_manipulation.out
root@debian-armhf:~/test/2.3# ls
Makefile random_input strcat.s strcmp.s strcpy.s string_manipulation.c strlen.s
root@debian-armhf:~/test/2.3# head string_manipulation.c
#include <stdlib.h>
#include <stdio.h>
// #include <string.h>

extern int strlen(char *s);
extern char *strcpy(char *dest, char *src);
extern int strcmp(char *s1, char *s2);
extern char *strcat(char *dest, char *src);

int main(int argc, char *argv[]) {
root@debian-armhf:~/test/2.3# make
gcc -Wall -g -c string_manipulation.c -o string_manipulation.o
gcc -Wall -g -c strcat.s -o strcat.o
gcc -Wall -g -c strcmp.s -o strcmp.o
gcc -Wall -g -c strcpy.s -o strcpy.o
gcc -Wall -g -c strlen.s -o strlen.o
gcc string_manipulation.o strcat.o strcmp.o strcpy.o strlen.o -o string_manipulation.out
root@debian-armhf:~/test/2.3# ./string_manipulation.out random_input
root@debian-armhf:~/test/2.3# ls
Makefile                strcat.s                string_manipulation.o
random_input            strcmp.o                string_manipulation.out
random_input_concat_out strcmp.s                strlen.o
random_input_len_out    strcpy.o                strlen.s
random_input_sorted_out strcpy.s
strcat.o                string_manipulation.c
root@debian-armhf:~/test/2.3# head random_input_concat_out
)o~"gWQ#.0EV4u\IEa.G6A\).DL!k$0':O{rdCV`D*_Kg`H?S@|b8p|+KeAc*@A]Xj2IGH~\ya2l?63!Y5^:&J.eQtivcw
AqqYm&.5,P|TyvuK
_mQhuD16>z16>1RvQX5t#y:&6mNFeZ%bnP[4#{Vcu1JnOY"*DX:_Mo,TxthA(60}>
eawfgiQQL^?e3xo(b-[|&k:P0>S)7~'oUw"qoyhMxNi4X9e5UF7Y*PHJ
gq^%r9E`^#\#xcam'U9d>KN=!U}F,+A[!;+p(EXQhYu|365cb+!4Y|q+m&ZF1COvT3t[IMV3F'!N0iKx0EtH39+
uylp<|.OV@2?(Wk*kRq'\p@n1T;j0_+&R]jWWw_XGFe{DY6mve:+ '6E1[tJ5_5[Zaq|XCSRHD&8it3[%4`F!On6z4--W0d
d[ ) [VeMetJH?HpXg0?ss
F2Qd#~5p;|9JI-h\DuL\A7RIW5UUC#|MDCn0E9`e44unt1MWDK|dNyq9L?AB({
gHn/#QB8V]?OX!xaf0?-urW=M66zW?I7Sm!5#1X3ZGMR|.}%$D
""}My-ml#CifU|e]S}*,,b%c}t3~D6QT@FMVxXMg)roBrA,"dOH1tVVASx/_w0e9?Hn$&ysL=,h94z'huNut|IAJu$Zl%y
RCb1
T(z4)2I13vYi=J_J@c'hKyw*'S]8`j^:F0]].W98
|XxLq#ja4"|"#^Qjne<}jIHU!j<|ox%EiAi-\2XT'd!nqG0h&?F$-0szKScMAM/A_L}JZ^/40{
root@debian-armhf:~/test/2.3#
```

Στον παραδοτέο κώδικα υπάρχουν όλα τα παραπάνω αρχεία με σχόλια καθώς και makefiles.