

AESO 2024-25

Seconda prova di verifica

Si richiede l'implementazione in assembler ARMv7, rispettando tutte le convenzioni per il passaggio dei parametri e l'utilizzo dei registri, due funzioni:

- una funzione `int conta(char * s)`, che conta il numero di volte che nella stringa il cui indirizzo è passato come parametro (s) appaiono due caratteri ASCII consecutivi. In altri termini, conta le occorrenze di due caratteri che hanno codice ASCII i (primo carattere) e $i+1$ (secondo carattere).

Ad esempio, la stringa "ab23Abcd" ha le sequenze di caratteri "ab", "23", "bc", "cd" che soddisfano la condizione di ricerca. Dunque il risultato di `conta("ab23Abcd")` deve essere 4. Notare che la sequenza "Ab" non soddisfa la condizione in quanto il codice ASCII di 'A' corrisponde a 65 mentre quello di 'b' a 98. Si allega comunque la tabella ASCII come riferimento.

- una funzione `int conta_stringhe(char **v, int n)`, che restituisce la somma dei valori ottenuti applicando la `conta()` sugli elementi del vettore di stringhe `v[]` di `n` elementi.

Si ricorda anche le stringhe sono NULL terminated, ovvero che un puntatore `char *` è l'indirizzo in memoria dal primo carattere della stringa e che la fine dei caratteri della stringa stessa è indicata da un carattere NULL (codice 0).

Per testare il corretto funzionamento delle proprie implementazioni delle due funzioni, si possono utilizzare i due main (C o ASM) forniti di seguito. Uno serve a testare la `conta` e l'altro l'uso congiunto di `conta` e `conta_stringhe`.

Per testare il corretto funzionamento del codice avete due opportunità:

1. utilizzare la macchina di laboratorio2 (o docker o installazione locale su ubuntu/WLS2). In questo caso, i file si compilano con il comando `arm-linux-gnueabi-gcc -z noexecstack -static nomefile.s` e si eseguono con il comando `qemu-arm ./a.out ...`. In caso si vogliano testare con i main forniti il comando di compilazione deve includere tutti i file richiesti (quello del main, .c o -s e quello della funzione/i da testare, .s)
2. utilizzare cpulator (pagina web all'indirizzo <https://cpulator.01xz.net/?sys=arm>). In questo caso, le funzioni possono essere testate con del semplice codice preso dai main.s che trovate qui sotto.

Prova della sola funzione che conta le occorrenze in una singola stringa

```
/* conta occorrenze nella stringa passata come primo argomento della riga
di comando o in una stringa di default espressa direttamente nel codice
./a.out aa deve restituire 0
./a.out abc deve restituire 2
*/

#include <stdio.h>

extern int conta(char *);

int main(int argc, char ** argv) {

    if(argc != 1)
        printf("conta(%s)=%d\n",argv[1],conta(argv[1]));
    else
        printf("conta(abe2EfGHijjkkml)=%d\n",conta( "abe2EfGHijjkkml"));
    return(0);

}
```

Prova di entrambe le funzioni

```
/* conto le occorrenze nelle stringhe passate come parametri della
riga di comando
-/a.out aa ab abc Dedfde deve restituire 4
*/

#include <stdio.h>

extern int conta_stringhe(char **, int);

int main(int argc, char ** argv) {

    char ** v = &argv[1];
    int n = argc - 1;

    printf("contastringhe=%d\n", conta_stringhe(v,n));
    return(0);

}
```

Prova della conta

```
.data
s: "stringa di prova"@ modificare la stringa a piacimento

.text
.global main
.type main, %function

main: ldr r0, =s @ indirizzo della stringa
      bl conta @ chiamata della conta
      mov pc, lr @ qui r0 = #occorrenze delle coppie di caratteri
consec.
```

Prova della conta-stringhe

```
.text
.global main
.type main, %function

main: push {lr}
      bl prep @ prepara v
      ldr r0,=v
      ldr r1,=n
      ldr r1,[r1]
      bl conta
      mov r1, r0 @QUI SI PUÒ CONTROLLARE IL RISULTATO
      ldr r0, =fmt @ DA QUI IN POI FUNZIONA SOLO CON CROSS-TOOLCHAIN ...
      bl printf
      pop {pc}

.data
fmt: .string "Ris = %d\n"
s1: .string "abdefhu" @ deve dare 3 (ab, de, ef)
s2: .string "HJdfg" @ deve dare 1 (fg)
s3: .string "aerdswxz"@ deve dare 1 (wx)
n: .word 3
v: .fill 12

.text

prep: ldr r0, =v
      ldr r1, =s1
      str r1, [r0],#4
      ldr r1, =s2
      str r1, [r0],#4
      ldr r1, =s3
      str r1, [r0],#4
      mov pc, lr
```