

Architettura degli elaboratori – A.A. 2020-2021 – Corso A e B

12 Gennaio 2022

Una coda di interi utilizza elementi definiti (in C) come:

```
typedef struct __elemento {
    int info;
    struct __elemento * next;
} ELEM;
```

Si implementino in assembler ARMv7 le tre funzioni:

- `int isEmpty(ELEM ** coda)`
che restituisce vero o falso (1 o 0) a seconda se coda è o non è vuota
- `void push(int x, ELEM ** coda)`
inserisce x in testa alla coda
- `int pop(ELEM ** coda)`
restituisce il campo info del primo elemento della coda e rimuove l'elemento stesso dalla coda.

(Si presti attenzione al tipo del parametro coda: è l'indirizzo di una cella di memoria che contiene l'indirizzo del primo elemento della coda oppure il valore NULL, ovvero 0, in caso di coda priva di elementi)

Se le funzioni sono state definite correttamente, il seguente programma C `check.c` fa vedere come vengono inseriti tutti i parametri passati da riga di comando e successivamente estratti dalla coda:

```
#include <stdio.h>
#include <stdlib.h>
#include "elem.h"

void stampa(ELEM * p) {
    if(p == NULL) printf("NULL\n");
    else { printf("%d -> ", p->info); p = p->next; stampa(p); }
    return;
}

int main(int argc, char ** argv) {
    ELEM * coda = NULL;
    int i;

    stampa(coda);
    for(i=1; i<argc; i++) {
        push(atoi(argv[i]), &coda); stampa(coda);
    }
    while(!isEmpty(&coda)) {
        printf("pop -> %d\n", pop(&coda)); stampa(coda);
    }
    return(0);
}
```

Il file `elem.h` contiene la definizione di `ELEM` oltre alle tre dichiarazioni relative alle funzioni esterne:

```
typedef struct __elemento {
    int info;
    struct __elemento * next;
} ELEM;

extern void push(int, ELEM**);
extern int isEmpty(ELEM**);
extern int pop(ELEM**)
```

Una esecuzione del programma (con le funzioni implementate correttamente) darebbe come risultato:

```
pi@raspberrypi:~/Gennaio $ gcc check.c ops.s
pi@raspberrypi:~/Gennaio $ ./a.out 1 2 3
1 -> NULL
2 -> 1 -> NULL
3 -> 2 -> 1 -> NULL
pop = 3
2 -> 1 -> NULL
pop = 2
1 -> NULL
pop = 1
NULL
pi@raspberrypi:~/Gennaio $
```