

# *Università degli Studi di Verona*

## **Dipartimento di Informatica**

### **Corso di Laurea in Informatica**

#### **ASM A.A 2022/2023**

##### **Descrizione:**

Realizzare un programma Assembly per la gestione del menù cruscotto di un'automobile. Il menù dovrà visualizzare e permettere l'impostazione delle seguenti informazioni: data, ora, impostazione blocco automatico porte, back-home, check olio. Se acceduto in modalità supervisor, il menù dovrà permettere anche l'impostazione di lampeggi frecce modalità autostrada e reset pressione gomme.

##### **Specifiche:**

Il programma potrà essere avviato in due modalità:

- utente (lanciando solo il nome dell'eseguibile da riga di comando) •  
supervisor (lanciando il nome dell'eseguibile seguito dal codice **2244**).

Se avviato in modalità utente, il programma dovrà visualizzare, una riga alla volta, il seguente menù, partendo dalla prima riga e scorrendo sulle altre premendo il tasto freccia-giù/freccia-su+invio

(inserire freccia-giù+invio da messaggio di riga 6 corrisponde alla visualizzazione del messaggio riga 1):

1. Setting automobile:
2. Data: 15/06/2014
3. Ora: 15:32
4. Blocco automatico porte: ON
5. Back-home: ON
6. Check olio

Se avviato in modalità supervisor, il programma dovrà visualizzare la riga 1 nel seguente modo:

“Setting automobile (supervisor)”, e poter visualizzare anche le seguenti righe:

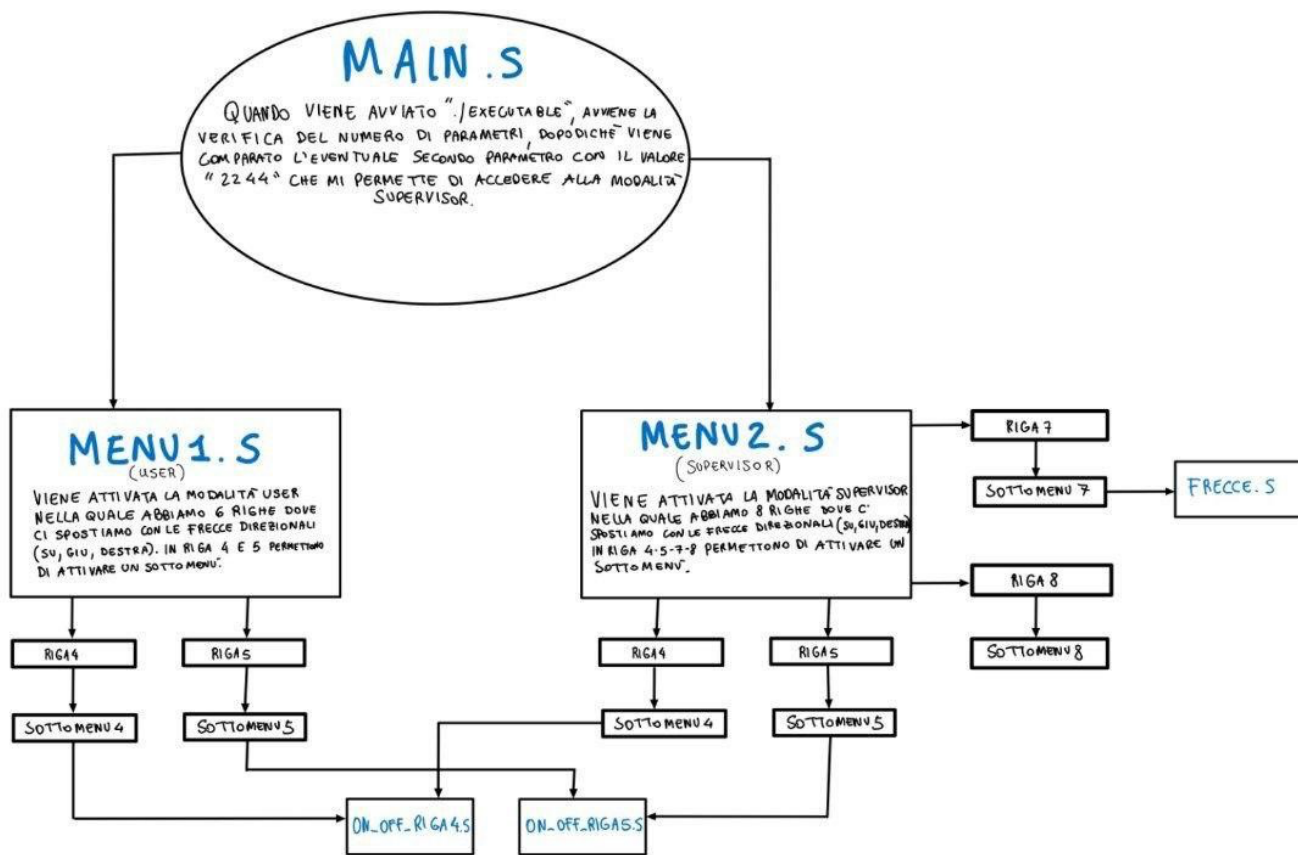
7. Frecce direzione
8. Reset pressione gomme

Ad ogni voce visualizzata, premendo i tasti freccia-destra+invio si potrà entrare nel sottomenù corrispondente, nel quale verrà visualizzato lo stato attuale del setting e data la possibilità di impostazione.

Ad esempio, una volta visualizzato il menù “Blocco automatico porte: ON”, premendo il tasto freccia-destra+invio, si dovrà permettere il cambiamento del setting ON, tramite i tasti freccia-giù o freccia-su+invio. In particolare, si dovrà permettere il setting dei sottomenù 4 e 5, con possibilità di impostazione ON/OFF come nell’esempio sopra. I sottomenù 2, 3, 6, non dovranno essere implementati in questo elaborato. All’interno di un sottomenù, il solo inserimento di invio da tastiera corrisponde al ritorno al menù principale.

Se avviato in modalità supervisor, il sottomenù “Frecce direzione” dovrà visualizzare il numero dei lampeggi modalità autostrada (3 per default) con possibilità di variazione (valore minimo 2, valore massimo 5) tramite tastiera (valori inseriti fuori range corrispondono al setting del max/min valore). Nel sottomenù “Reset pressione gomme”, inserendo il carattere freccia-destra seguito da invio, il menù dovrà restituire il messaggio “Pressione gomme resettata” e tornare al menù principale

## DIAGRAMMA DI FLUSSO:



## FILE:

- **MAKEFILE:** dove attraverso il comando make costruiamo in simultanea i file eseguibili.
- **MAIN:** dove parte il tutto e dove si capisce se andare in modalità supervisor o no.
- **2 MENU:** **menu1.s** dove si avvia la modalità utente, **menu2.s** dove si avvia la modalità supervisor.
- **FRECCES:** dove gestiamo il numero di frecce.
- **ON\_OFF\_SUPERVISOR** e **ON\_OFF\_USER:** che con la stessa funzione, uno gestisce il supervisor l'altro utente, cambiano lo stato da ON a OFF.

- **LETTURA:** dove mi legge il comando inserito dall'utente.
- **atoi e itoa:** le due funzioni che mi convertono **atoi** converte una stringa numerica in un valore intero. **itoa** il contrario.

## VARIABILI:

Nel menu denominato "**menu1.s**", abbiamo definito diverse righe di codice. Ogni riga contiene una stringa da stampare, che viene salvata utilizzando ".ascii", e la sua lunghezza relativa, che viene salvata utilizzando ".long". Inoltre, nel nostro menu, abbiamo anche il codice per la stampa dei valori "**ON**" e "**OFF**". Utilizziamo le variabili **r4t** e **r5t** per la formattazione a tabulazione e le variabili **r4\_state** e **r5\_state** per tenere traccia se siamo nello stato "**ON**" o "**OFF**".

Nel menu denominato "**menu2.s**", abbiamo delle righe di codice aggiuntive rispetto a menu1.s, ovvero la riga 7 e la riga 8. Per la riga 7 aggiungiamo il salvataggio del numero di lampeggi utilizzando la variabile "**lampeggi**".

Nel file denominato "**frecce.s**", abbiamo una variabile chiamata "**str**" che contiene la stringa della riga 7, la quale viene formattata con un carattere di tabulazione e la sua lunghezza viene salvata utilizzando l'istruzione ".long". Inoltre, abbiamo anche una variabile chiamata "**lampeggi**" che viene utilizzata per salvare il numero di lampeggi.

Nel codice denominato "**lettura.s**", abbiamo una variabile chiamata "**var**" che viene utilizzata per salvare quale comando è stato utilizzato.

Nei file denominati "**on\_off\_supervisor.s**" e "**on\_off\_user.s**", abbiamo due variabili comuni: "**riga**" e "**stato**".

La variabile "**riga**" viene utilizzata per indicare in quale riga ci troviamo all'interno del codice. La variabile "**stato**" viene utilizzata per determinare se siamo nello stato "**OFF**" (spento) o "**ON**" (acceso). Questa variabile tiene traccia dello stato attuale del sistema.

## FLUSSO DI PROGRAMMA:

Il programma inizia da main e solamente con l'inserimento di **2244** si potrà accedere alla modalità supervisor altrimenti si passa alla modalità user. In entrambi i casi si parte alla rispettiva riga 1 e con le frecce direzionali "**SU**" e "**GIU**" ci si muoverà nel menù. Nelle righe 4 e 5 sia in Super che non, attraverso la freccia direzionale destra si potrà accedere

al relativo sottomenu e richiamare **on\_off\_supervisor.s** o **on\_off\_user.s** e con questo cambiare stato. In supervisor in riga 7 con la freccia destra si entra nel relativo sottomenu 7 e con la call alla funzione **frecce.s** si potrà cambiare il numero di lampeggi. In riga 8 sempre con destra verrà **r8t** che contiene la stringa riguardante il reset. Se sono in cima al menu e vado in alto andrò in fondo al menu e viceversa.

Per gestire le frecce abbiamo fatto la somma dei caratteri che la componevano, ad esempio freccia SU “ ^[, [ , A “ corrispondeva a “27, 91, 65”, sommandoli si ottiene 183. Comparando questo numero al comando digitato dall’utente si salta all’etichetta corrispondente. Abbiamo agito alla stessa maniera per i caratteri "GIÙ, DESTRA" con i rispettivi numeri “184, 185”. Per il tasto ENTER è più semplice perché corrisponde al numero “0”.

Per testare il tutto basterà entrare da riga di comando nella cartella contenente il **makefile** e generare tutti i file oggetto tutti in una volta sola utilizzando il comando “**make**”. Dopodichè avviare l’e eseguibile con “**./executable**” per la modalità user e “**./executable 2244**” per la modalità supervisor. Ogni altra combinazione sbagliata permetterà di entrare esclusivamente nella modalità user.

Per quel che riguarda l’implementazione in linguaggio C del cruscotto è evidente che sia molto piu’ compatto, soprattutto grazie a delle librerie che ci vengono in aiuto. Ad esempio la libreria <nurses.h> ci permette di scorrere direttamente nel menu senza dover premere invio ogni volta. Per testarlo basterà utilizzare da riga di comando “gcc -o cruscotto cruscotto.c -Inurses” , una volta generato il file oggetto, avviarlo con “./cruscotto”. Bisogna anche verificare che tutte le librerie siano state installate specialmente con linux, in questo caso la libreria “libnurses-dev”.