

Università degli Studi di Verona

Relazione di Elaborato Assembly

Corso di laurea in Informatica – Architettura degli
Elaboratori - Laboratorio

Mezzacasa Cristian VR489474

Zeni Davide VR486320

Anno accademico 2022/2023

Descrizione

realizzazione di un programma ASM per l'utilizzo del **menù cruscotto di un'auto**.

Il programma permette di accedere a due modalità diverse del menù, rispettivamente utente (*user*) e supervisore (*supervisor*), quest'ultimo attraverso l'inserimento in linea di comando del codice '2244'. Il programma permette, attraverso le frecce direzionali (freccia-su/giù+invio), lo scorrimento delle voci del menù:

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

L'accesso al menù da supervisore permette la visualizzazione delle voci aggiuntive "Frecce direzionali" e "Reset pressione gomme".

7. Frecce direzionali
8. Reset pressione gomme

Per le voci 4, 5, 6, 7 (le ultime due solo in modalità supervisor) sono permesse le modifiche attraverso l'inserimento da tastiera di freccia-destra+invio: per le prime due voci con freccia-su/giù+invio è possibile cambiare da ON a OFF e viceversa. La voce 7 visualizza il numero di lampeggi modalità autostrada (default 3) modificabili tramite tastiera con un numero compreso tra un minimo di 2 lampeggi ed un massimo di 5, qualsiasi altro valore fuori dal range corrisponderà al setting del valore minimo o massimo. Infine la voce 7 visualizza il messaggio "Pressione gomme resettata" per poi ritornare al menù principale.

Mentre ci si trova nei sopra citati sottomenù con l'inserimento da tastiera di "invio" è possibile ritornare al menù principale.

Variabili utilizzate

file getArrow.s:

- **Input:** variabile ascii lunga 4 byte, ognuno dei quali rappresenta un carattere della codifica di freccia+invio, rispettivamente i primi tre per la freccia e l'ultimo per l'invio.

file BlinksManager.s:

- **Input:** variabile ascii di 2 byte, serve ad acquisire numero+invio durante la modifica del numero di lampeggi.

File isSupervisor.s:

- **supervisorCode:** variabile ascii definita come "2244\0", identifica il codice necessario a riga di comando per poter accedere al menù supervisor
- **supervisor:** variabile globale che identifica se il programma è stato eseguito in modalità supervisor o meno

file menuList.s:

- **menu1:** variabile ascii definita come "Setting automobile\n"
- **menu2:** variabile ascii definita come "Data 26/05/2023\n"
- **menu3:** variabile ascii definita come "Ora 14:32\n"
- **menu4:** variabile ascii definita come "Blocco automatico porte"
- **menu5:** variabile ascii definita come "Back-home"
- **menu6:** variabile ascii definita come "Check olio\n"
- **menu7:** variabile ascii definita come "Frecce direzione"
- **menu8:** variabile ascii definita come "Reset pressione gomme"
- **supervisorTag:** " (supervisor)\n"

File menu.s:

- **user:** variabile ascii "menu normale". La variabile **LenUser** ne indica la lunghezza
- **freccia:** variabile ascii "0", indica quale freccia e' stata premuta
- **porte:** valore del menu Blocco automatico porte (ON/OFF -> 1/0)
- **lampeggio:** per il valore del lampeggio inserito dall'utente
- **moreOption:** se il menu e' nella modalita moreOption o meno
- **backHome:** analogo della variabile porta
- **limitofMenu:** limiter del contatore 5/7
- **menuCounter:** contatore del menù

File newLine.s:

- **newLineChar:** \n
- **lenNewllne:**

File printMoreOptionFreccia.s:

- **lampeggio**

File printMoreOptionONOFF.s:

- **ON:** variabile ascii "ON\n"
- **OFF:** variabile ascii "OFF\n"

File printMoreOptionFreccia.s:

- **pressione:** "Pressione gomme resettata"
- **lenPressione**

Questa funzione ha lo scopo di controllare del programma, esegue le funzioni principali come (getArrow,blinksmanager,menulist)

Modalità di passaggio/restituzione dei valori delle funzioni create

i parametri vengono passati esclusivamente attraverso lo stack.

Funzione **menu.s**: Funzione cuore del programma perché gestisce tutte le call a funzione e la gestione dei valori da loro restituiti. come parametri riceve gli argomenti dell'esecuzione del programma tra cui il codice supervisor

Il file **isSupervisor.s** è il programma richiamato dal file menu.s che riceve come argomento se presente il codice 2244 dalla funzione menu ed imposta la variabile globale isSupervisor

Funzione getArrow.s: prende da tastiera la codifica delle frecce per potersi muovere all'interno del menù e dei sottomenù delle voci predisposte alla modifica. Se ci troviamo all'interno di uno dei sottomenù sopracitati, il ricevimento da terminale del solo invio ci permette di ritornare al menù principale. Per questo deve avere come parametro la variabile moreOption.

L'input viene validato attraverso una serie di controlli sui singoli caratteri così da non permettere l'inserimento di valori non permessi

Caratteri permessi:

- Freccia su
- Freccia giù
- Freccia destra
- Invio

Funzione **menuList.s**: permette lo scorrimento da una voce all'altra del menù e l'entrata a sottomenù per le voci che possiedono quest'ultimo. A seconda del contatore ricevuto da menu.s, menuList visualizzerà il menù utente o la variante supervisor.

I parametri che richiede sono:

- menuCounter: variabile contatore che può assumere valori da 0 a 4 oppure da 0 a 6 a seconda se si è in modalità supervisor o meno
- moreOptions: indica se si è nella modalità moreOption accessibile attraverso la freccia destra nei menu 4,5,7,8
- Value: valore della variabile da stampare nel sottomenu(porte,backHome,lampeggio)

esempio

```
if(menuCounter == 4 && value){  
    printf("ON");  
}  
else{  
    printf("OFF");  
}
```

Funzione blinkManager.s: è la funzione che permette l'impostazione del numero di lampeggi modalità autostrada. La funzione riceve da tastiera 'numero+invio' e successivamente modifica il valore numerico della rispettiva voce del menù. Inoltre la funzione si occupa di riconoscere valori fuori range e impostare la modifica restituendo l'estremo dell'intervallo rispettivo: vuol dire che se viene inserito un numero minore di 2 (valore minimo di lampeggi disponibile) allora il numero di lampeggi verrà portato a 2, mentre se si

inserisce un numero maggiore di 5 il numero di lampeggi restituito sarà proprio 5 (valore massimo di lampeggi disponibile).

Funzioni di utility:

- **printMoreOptionPressione**
- **printMoreOptionFreccia**
- **printMoreOptionONOFF**
- **newLine**

Diagramma di flusso del programma

1. In menu.s:

- chiamata di isSupervisor.s

- in isSupervisor:

- 1. Definizione della costante superVisorCode come "2244\n"
- Definizione delle variabili globali isSupervisor, cont e limitOfMenu
- Definizione delle variabili freccia, porte, backHome, lampeggio, pressione, parameterMenuList, moreOption
- Viene letto il parametro utente da riga di comando e controllata la corrispondenza al superVisorCode. isSupervisor viene impostato a 1 se il codice corrisponde, altrimenti a 0

- inizializzazione le variabili locali e impostazione limitOfMenu a 7 se isSupervisor è 1, altrimenti a 5

- ciclo while infinito:

- viene chiamata la funzione menuList con i parametri cont, parameterMenuList e moreOption. Se moreOption è vero e cont è uguale a 6:

- chiama il metodo blinksManager per gestire le opzioni aggiuntive
- se il valore restituito è 10, imposta a 0 moreOption , altrimenti imposta lampeggio e parameterMenuList con il valore restituito oppure chiama il metodo getArrow per ottenere l'input dell'utente

- si utilizza una struttura switch per gestire le varie opzioni in base all'input dell'utente:

- se l'input corrisponde a una freccia su, si decrementa cont a meno che moreOption non sia vero e cont non sia già 0, altrimenti si modifica il valore di porte o backHome in base alla situazione
- se l'input corrisponde a una freccia giù, viene incrementato cont a meno che moreOption non sia vero e cont non sia già uguale a limitOfMenu, altrimenti il valore di porte o backHome viene modificato in base alla situazione
- se l'input corrisponde a una freccia destra, si modificano le variabili moreOption e parameterMenuList in base a cont
- se l'input corrisponde a un invio, moreOption viene impostato a 0

- fine della struttura switch

- fine del ciclo while

2. Nella funzione menuList:

- si utilizza una struttura switch per stampare il menù in base al valore di cont e in base a quest'ultimo si stampa il menu corrispondente
- se moreOption è vero, si stampa il valore corrispondente a value
- fine della struttura switch

3. Nella funzione getArrow:

- ciclo while finché non viene inserito un carattere valido
- si legge un carattere dall'input. Se il carattere è '\e', leggi ulteriori caratteri per determinare l'input corrispondente a una freccia. Restituisci il valore corrispondente all'input dell'utente

4. Nel metodo blinksManager:

- ciclo do-while finché non viene inserito un valore valido
- viene letto un carattere dall'input che viene convertito in ASCII se il valore non è un numero valido compreso tra 0 e 9 o l'invio non viene premuto correttamente, isValid si imposta a falso e scarta il resto della riga
- se il valore è minore di 2, si imposta il valore a 2 se il valore è maggiore di 5, imposta il valore a 5 - Restituisci il valore corretto

5. Fine del programma