

CORSO DI ARCHITETTURA DEGLI ELABORATORI A-L M-Z

PROGETTO di PROGRAMMAZIONE ASSEMBLY

15 Luglio 2008

CONSEGNA Lunedì 15 Settembre 2008

Lo studente programmi nel linguaggio Assembly 8088 presentato a lezione un micro 8088 usato per controllare il funzionamento di un Portable Digital Assistance (PDA) (palmare).

Si usi la versione di assembly e l'assemblatore allegato al libro "Andrew S. Tanenbaum. *Architettura dei Calcolatori. Un Approccio Strutturato*. Pearson Prentice Hall, 5 ed., 2006.", e scaricabile dall'indirizzo internet <ftp://ftp.cs.vu.nl/pub/evert>.

1. Descrizione del sistema

In Figura 1 è rappresentato lo schema del dispositivo, denominato **PALM88**. Il sistema è composto da una consolle di dialogo con l'utente (tastiera), un display, una memoria principale (RAM), una memoria ROM, una periferica GPS di input ed un microprocessore. Il blocco 8088 incorpora anche la logica di controllo delle varie periferiche.

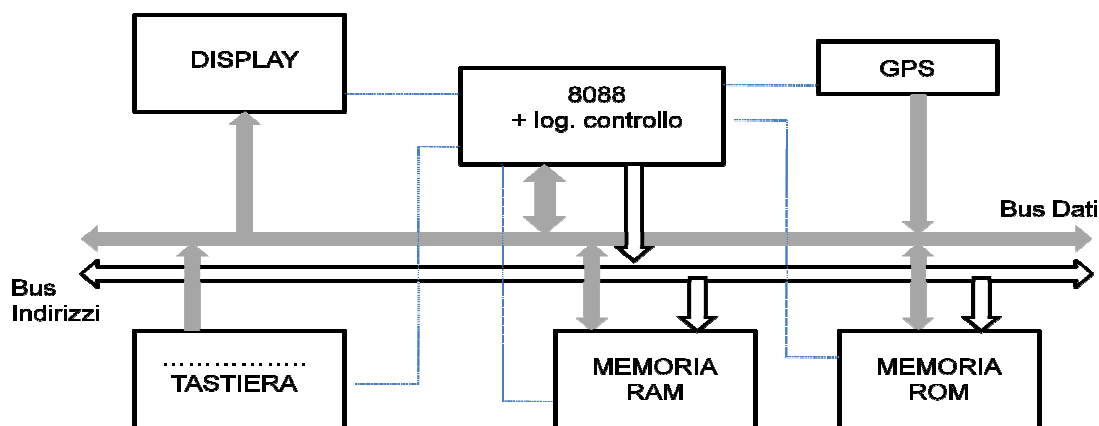


Figura 1: Schema del dispositivo PALM88

Il dispositivo **PALM88** mette a disposizione le seguenti funzionalità:

- 1) **Controllo degli accessi.** Appena acceso, il sistema rimane di attesa di leggere il codice PIN da tastiera. Il codice immesso da tastiera viene confrontato con il codice contenuto in memoria ROM. Se il codice PIN risulta corretto, il sistema visualizza il menù principale con l'elenco dei servizi

implementati dal dispositivo (vedi sotto). In caso contrario, il sistema visualizza un messaggio di errore e resta in attesa di leggere un nuovo codice PIN dall'utente. Se si digita per tre volte un codice errato, il sistema resta in attesa di leggere da tastiera il codice PUK (una sola volta). In caso di inserimento errato del codice PUK, il sistema termina di funzionare.

- 2) **Menù di selezione.** Il dispositivo visualizza sul display la lista delle funzionalità disponibili e resta in attesa di una selezione da parte dell'utente.
- 3) **Rubrica Elettronica.** Il dispositivo consente di inserire un nuovo contatto, di visualizzare i contatti esistenti e di ricercare contatti all'interno della rubrica.
- 4) **Servizio SMS e Gestione del credito.** Il dispositivo consente di inviare un messaggio di testo ad un contatto della rubrica. Il credito residuo viene decrementato in base all'operatore del mittente e del destinatario. Si assume esistano 3 tipi di operatori, denominati con le lettere **A,B,C**.
- 5) **Funzionalità GPS.** Il dispositivo riceve in input dal dispositivo GPS le attuali coordinate e visualizza sul display la posizione corrente all'interno della mappa.

2. Funzionalità del sistema

Il sistema funziona nel seguente modo:

- 1) Legge da input il codice PIN/PUK
- 2) Visualizza il menù di selezione
- 3) Legge da input la scelta dell'utente
- 4) Esegue la funzionalità richiesta
- 5) Torna al punto (2)

3. Emulazioni

Non avendo a disposizione un sistema reale, in particolare le periferiche di I/O, si introducono le seguenti emulazioni:

- 1) Tastiera e display sono emulati dalla consolle dell'elaboratore usato.
- 2) L'accesso alla memoria ROM è emulato dalla lettura del file *rom.txt*. Il file *rom.txt* contiene le seguenti informazioni: < Codice PIN, Codice PUK, Credito Attuale, Operatore >

Esempio di file: 1234 5678 200 A

- 3) L'accesso in lettura/scrittura alla rubrica è emulato dal file *rubrica.txt*. Ogni contatto della rubrica contiene le seguenti informazioni: <Nome Contatto, Numero Telefono, Identificativo_Operatore >

Esempio di file: Casa 23456 A

Ufficio 12345 B

- 4) La rilevazione delle coordinate GPS è emulata dalla lettura del file *gps.txt*. Il file *gps.txt* contiene la seguente informazione: <Coordinata_X; Coordinata_Y>.

Esempio di file: 10 10

SI SVOLGANO I SEGUENTI PUNTI

Punto 1

Si implementi la procedura

```
int GET_CODE(void)
```

che resta in attesa di leggere il codice dall'input e restituisce tramite stack il codice (un intero) letto.

Si implementino le procedure

```
int IS_PIN_CORRECT(void) int IS_PUK_CORRECT(void)
```

che confrontano il codice letto dall'input con i codici PIN/PUK contenuti nella ROM e restituiscono 0 in caso di codice corretto, -1 in caso di codice non corretto.

Utilizzando le funzioni `GET_CODE(void)`, `IS_PIN_CORRECT(void)` e `IS_PUK_CORRECT(void)` si implementi la procedura di controllo degli accessi definita nella sezione 1.

Punto 2

In caso di inserimento corretto del codice PIN/PUK d'accesso, si stampi il contenuto del menù con la lista delle funzionalità del dispositivo che sono state implementate (*). Nel caso in cui l'utente digiti una voce che non compare nel menù, si visualizzi un messaggio di errore.

Esempio di schermata:

```
+++++
                          PALM 88 Menu principale
1) Stampa rubrica
2) Inserisci contatto in rubrica
3) Visualizza credito
4) Ricerca in rubrica
5) Servizio SMS
6) Servizio GPS
7) Esci dal programma

Inserire scelta (1..7):

+++++
```

Punto 3

Fatto il punto 2, si implementi la funzionalità "Stampa Rubrica", che visualizza sul display la lista dei contatti contenuti nella rubrica (in pratica, il contenuto del file *rubrica.txt*).

(*) Le funzionalità di "Stampa Rubrica", "Inserisci Contatto" e "Visualizza Credito" sono obbligatorie.

Punto 4

Fatto il punto 3, si implementi la funzionalità “Inserisci contatto” che legge da tastiera le informazioni <Nome, Numero_Telefono> di un nuovo contatto e le aggiunge all’interno della rubrica scrivendole nel file *rubrica.txt*. La dimensione massima della rubrica è di 10 contatti. Nel caso in cui si cerchi di aggiungere un nuovo contatto con rubrica piena, si visualizzi un messaggio di errore.

Punto 5

Fatto il punto 4, si implementi la funzionalità “Visualizza credito” che legge il credito attuale dalla ROM (*rom.txt*) e lo visualizza sul display.

Punto 6

Fatto il punto 5, si implementi la funzionalità “Ricerca in rubrica” che legge da tastiera un carattere (es. “A”) e stampa a video tutti i contatti le cui iniziali contengono il carattere suddetto (es. ALDO, ANNA, ...).

Punto 7

Fatto il punto 5, si implementi la funzionalità “Servizio SMS” che legge da tastiera le seguenti informazioni: <Nome_destinatario, Testo_SMS>. Nel caso in cui il nome del destinatario non appaia in rubrica, viene visualizzato un messaggio di errore. Il credito residuo viene decrementato in base al gestore del mittente e del destinatario del SMS, secondo la seguente tariffa:

Operatore Mittente	Operatore Destinatario	Costo SMS
A	A	1
A	B	2
A	C	3
B	A	2
B	B	0
B	C	3
C	A	2
C	B	2
C	C	2

(Si ricorda che l’informazione relativa al gestore del mittente è contenuta nella ROM). Nel caso in cui il credito residuo non sia sufficiente per l’invio dell’SMS, un messaggio di errore deve essere visualizzato sul display. Le informazioni <Nome_destinatario, Testo_SMS> relative ad ogni SMS correttamente inviato devono essere scritte all’interno del file *sms.txt*.

(Potete assumere che il Testo_SMS abbia dimensione massima prefissata, per esempio 20 caratteri).

Punto 8

Fatto il punto 5, si implementi la funzionalità “Servizio GPS” che legge le coordinate GPS correnti (*gps.txt*) e visualizza la posizione all’interno della mappa indicandola con il carattere “*”. L’area della mappa è di 30x30 caratteri.

I punti 1-2-3-4-5 sono obbligatori per valutare positivamente l'elaborato. I punti 6-7-8 sono facoltativi e concorrono al raggiungimento del massimo dei voti.

Quanto non riportato nelle specifiche fa parte delle scelte del progettista, quindi dello studente. Si riportino le scelte e le assunzioni fatte, commentandole, nella relazione da allegare al progetto.

NOTE PER LA CONSEGNA

La valutazione del progetto dipende strettamente dai seguenti vincoli:

- Il progetto deve essere consegnato entro le 24.00 della data indicata nella prima pagina.
- Saranno attribuiti dei punti ai seguenti aspetti generali:
 - 1) **Presenza di commenti.** Il codice assembly deve essere commentato in modo da agevolare la comprensione. Ogni funzione deve essere preceduta da opportuni commenti che ne chiariscono la funzionalità e gli argomenti. **I progetti privi di commenti NON saranno valutati.**
 - 2) **Codice funzionante.** Il progetto deve essere assemblabile e funzionante. Indicare nella relazione quali sono i Punti di questo testo svolti e quale sistema operativo si è usato.
 - 3) **Codice ottimizzato.** Il codice disordinato è difficile da verificare. Saranno valutate positivamente scelte progettuali che ottimizzino il codice. L'ottimizzazione deve riguardare anche i commenti: chiari e concisi.
 - 4) **Codice strutturato.** Il codice deve essere opportunamente strutturato facendo uso di chiamate a funzioni.
- **Relazione**, di max. 4 pagine per facilitare la lettura del codice. **Riportare nella relazione le scelte progettuali.**
- **Lavoro di gruppo.** Il progetto deve essere svolto da gruppi di lavoro, composti da almeno due persone e al massimo tre, in maniera collaborativa da tutti i componenti del gruppo. **Ogni componente del gruppo deve conoscere le scelte progettuali e i dettagli implementativi.**
- **Confronto tra gruppi.** E' utile il confronto tra gruppi. Tuttavia, dovendo ogni gruppo fare le sue scelte progettuali, è fondamentale evitare scelte di massa. Saranno valutati negativamente progetti simili tra loro o con simili scelte progettuali.

Modalità di consegna

- Preparare una cartella chiamata "**cognome1_cognome2_cognome3**" (dove **cognome1,cognome2, ...** sono i cognomi dei componenti il gruppo di lavoro; evitare spazi nei nomi) contenente le seguenti cartelle/file:
 1. cartella "**progetto**": contiene i file SORGENTI del progetto
 2. file "**relazione.pdf**": relazione PDF
 3. file "**sorgente.pdf**": file PDF contenente i SORGENTI del progetto
 4. file .pdf col testo del progetto (questo file!)
- Comprimere la cartella "**cognome1_cognome2_cognome3**" in "**cognome1_cognome2_cognome3.zip**"
- Spedire il file .zip allegato ad un mail ai seguenti indirizzi (ad entrambi e con una sola mail) entro il termine di consegna:

1. difelice@cs.unibo.it

2. biafelice@arces.unibo.it

- Indicare come **soggetto** del mail: “[ARCHI]cognome1_cognome2_cognome3”
- Riportare eventuali commenti nella **relazione**, non nel testo del mail.

Problemi riscontrati durante il precedente appello per la consegna

- Mail inviati separatamente ai due indirizzi: genera confusione rendendo difficile capire chi ha consegnato.
- Mail (forse) spedite ma non arrivate. Alcuni account di posta possono avere malfunzionamenti o limiti nella dimensione della posta in uscita. Si consiglia di mettere in copia alla mail di consegna l’indirizzo di un componente del gruppo per verificarne l’invio.
- Mail spedite senza l’allegato.

Errori di questo tipo possono rendere il progetto non valido vanificando il lavoro fatto.