

# CORSO DI ARCHITETTURA DEGLI ELABORATORI A-L e M-Z

## PROGETTO di PROGRAMMAZIONE ASSEMBLY

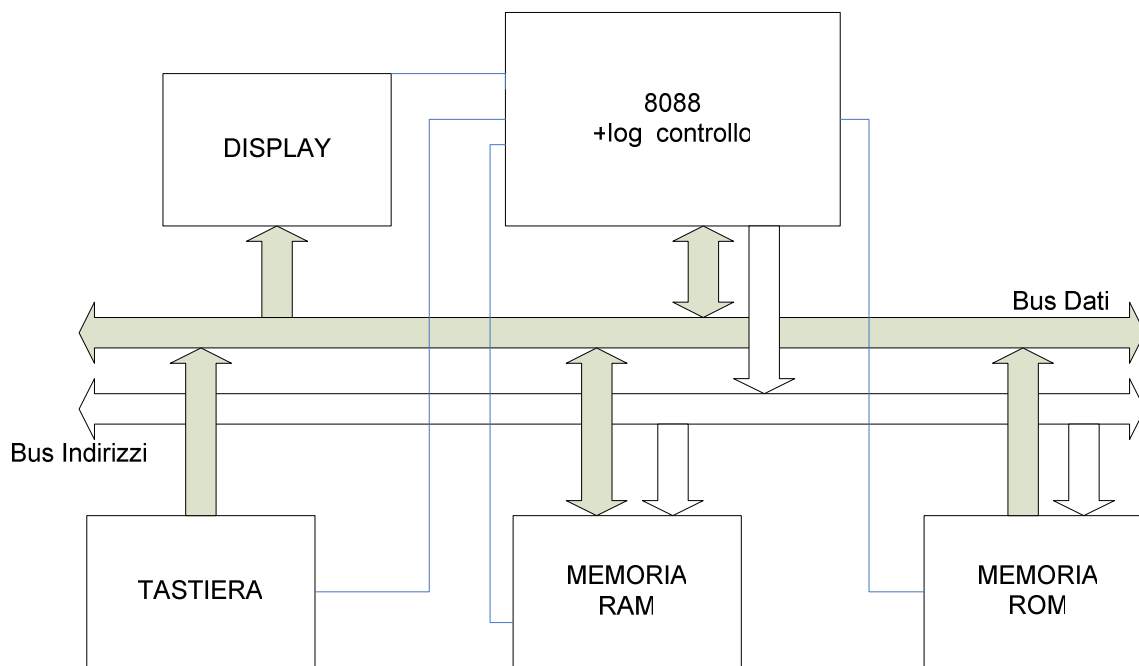
20 Dicembre 2008

**CONSEGNA Lunedì 2 Febbraio 2009**

Lo studente programmi nel linguaggio Assembly 8088 presentato a lezione un micro 8088 usato per controllare il funzionamento di una Automatic Teller Machine (ATM). Si utilizzino la versione di assembly e l'assemblatore allegato al libro di testo e scaricabile dall'indirizzo internet <ftp://ftp.cs.vu.nl/pub/evert>.

### Descrizione del Sistema

In Figura 1 è rappresentato lo schema del dispositivo, denominato **ATM88**. Il sistema è composto da una console di dialogo con l'utente (tastiera), un display, una memoria principale (RAM) e una memoria ROM ed un microprocessore.



**Figura 1:** Schema del dispositivo **ATM88**.

Il dispositivo **ATM88** mette a disposizione le seguenti funzionalità:

1. **Controllo degli accessi.** Appena acceso il sistema rimane in attesa di leggere il numero di conto e il PIN da tastiera, entrambi seguiti dal carattere #. Questa coppia di valori viene confrontata con i codici contenuti in memoria ROM. Se il codice PIN corrisponde al numero di conto, 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 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. **Gestione del Conto Corrente.** Il dispositivo permette di prelevare contante in Euro e in valuta straniera, di controllare l'ammontare di denaro presente sul proprio conto e di trasferire una somma di denaro su un altro conto.
4. **Pagamento Bollettini Postali.** E' possibile procedere al pagamento di bollettini postali, prelevando il relativo importo dal proprio conto.
5. **Ricariche Telefoniche.** Il dispositivo permette infine di ricaricare il proprio telefono cellulare, addebitando l'importo sul proprio conto corrente.

## Funzionalità del Sistema

Il sistema funziona nel seguente modo:

1. Legge da input il numero di conto e il codice PIN.
2. Visualizza il menù di selezione.
3. Legge da input la scelta dell'utente.
4. Esegue la funzionalità richiesta.
5. Torna al punto 2.

## 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 console 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: <Numero di Conto Corrente, PIN>. Ad esempio:

```
107593, 5632
927483, 3728
```

3. Il conto corrente è emulato tramite la RAM. L'accesso in lettura e scrittura al conto corrente è gestito tramite il file `conti.txt`. Il file `conti.txt` contiene le seguenti informazioni: <Numero di Conto Corrente, Ammontare in Euro>. Ad esempio:

```
107393, 2300
927483, 15200
```

4. Le comunicazioni tra il dispositivo e la compagnia telefonica sono emulate tramite un file `ricariche.txt` che risiede nella RAM. Il file `ricariche.txt` contiene le seguenti informazioni:  
<Numero telefonico, Ammontare in Euro>. Ad esempio:

```
3337561245, 100
3337561245, 20
3373728192, 25
```

Si osservi che lo stesso numero di telefono può essere ricaricato più volte e che in tal caso occorre comunicare più volte con la società telefonica.

5. Le comunicazioni tra il dispositivo e l'amministrazione postale sono emulate tramite un file `poste.txt`, che risiede nella RAM. Il file `poste.txt` contiene le seguenti informazioni: <Numero di conto corrente postale, Ammontare del versamento in Euro>. Ad

esempio:

```
1573727, 200
9467293, 100
```

6. I dati relativi ai rapporti di cambio tra alcune valute e l'euro sono memorizzate in un file `valute.txt`. Il file contiene le seguenti informazioni: <Valuta, Rapporto di Cambio>. Ad esempio:

```
DOLLARO, 1.3734
STERLINA, 0.856
FRANCO, 1.654
```

Il rapporto di cambio indica la quantità di valuta straniera corrispondente ad 1 Euro.

## Punti da svolgere

### Punto 1

Si implementino le procedure

```
int GET_NUMBER(void)
int GET_PIN(void)
```

che restano in attesa di leggere il numero di conto e il relativo PIN dall'input e restituiscono tramite stack gli interi letti. Si implementi poi la procedura

```
int IS_PIN_CORRECT(void)
```

che confrontato il numero di conto e il PIN letti con i dati contenuti in ROM e restituiscono 0 se i dati inseriti sono corretti e -1 se tali dati non sono corretti.

Utilizzando le funzioni di cui sopra si implementi la procedura di controllo degli accessi definita nella sezione 1.

### Punto 2

In caso di inserimento corretto del numero di conto e del PIN, 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:

```
*****
ATM88 - Menù principale

1) Visualizzazione Ammontare Residuo
2) Prelievo in Euro
3) Bonifico verso altro Conto
4) Ricarica Telefonica
5) Pagamento Bollettino Postale
6) Prelievo in Valuta

Inserire scelta (1 -> 6):
*****
```

### **Punto 3**

Fatto il punto 2, si implementi la funzionalità “Visualizzazione Ammontare Residuo” che visualizza sul display l'ammontare residuo del conto in Euro (che va reperito nel file conti.txt)

### **Punto 4**

Fatto il punto 3, si implementi la funzionalità “Prelievo in Euro” che legge da tastiera l'ammontare che l'utente vuole prelevare dal conto, controlla se tale somma è disponibile e in caso affermativo sottrae dal conto l'ammontare richiesto. Se la somma, invece, non è disponibile, si visualizzi un messaggio di errore.

### **Punto 5**

Fatto il punto 4, si implementi la funzionalità “Bonifico verso altro Conto” che legge da tastiera l'ammontare che l'utente vuole trasferire verso altro conto e il numero del conto beneficiario. Occorre prima di tutto controllare se la somma è disponibile (come per il Punto 4). Poi, occorre anche controllare che il numero del conto beneficiario effettivamente esiste. In caso negativo, occorre visualizzare un messaggio di errore. In caso affermativo, la somma va sottratta da un conto e aggiunta ad un altro conto.

### **Punto 6**

Fatto il punto 5, si implementi la funzionalità “Ricarica Telefonica” che legge da tastiera l'ammontare che l'utente vuole trasferire sul proprio telefono cellulare e il numero di telefono. Occorre, come al solito, controllare che la somma sia disponibile. Poi, occorre dare comunicazione alla società telefonica dell'avvenuta ricarica, tramite il file ricariche.txt

### **Punto 7**

Fatto il punto 6, si implementi la funzionalità “Pagamento Bollettino Postale” che legge da tastiera l'ammontare che l'utente vuole trasferire verso il conto corrente postale e il numero di conto corrente postale beneficiario. Dopo aver controllato che la somma è disponibile, si deve dare comunicazioni all'autorità dell'avvenuto versamento, tramite il file poste.txt

### **Punto 8**

Fatto il punto 7, si implementi la funzionalità “ Prelievo in Valuta”. Viene innanzitutto presentata all'utente la tabella delle valute disponibili, da reperire nel file valute.txt. L'utente sceglie la valuta tra tutte quelle disponibili, digitando il nome oppure il suo numero progressivo. L'utente, infine, inserisce la quantità di valuta estera richiesta. A questo punto, occorre controllare che il corrispondente ammontare in Euro sia disponibile e, eventualmente, segnalare all'utente l'impossibilità di effettuare il prelievo. Se l'ammontare è disponibile, la somma in Euro va ovviamente sottratta dal conto.

**I primi 7 punti sono obbligatori per valutare positivamente l'elaborato. I punto 8 è facoltativo e concorre 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:
  - cartella **“progetto”**: contiene i file SORGENTI del progetto
  - file **“relazione.pdf”**: relazione PDF
  - file **“sorgente.pdf”**: file PDF contenente i SORGENTI del progetto
  - 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:
  - [zeynep@cs.unibo.it](mailto:zeynep@cs.unibo.it)
  - [dallago@cs.unibo.it](mailto:dallago@cs.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.**