

**PROTOTIPAZIONE DI UN TRADUTTORE
DA SORGENTE PLC AD ASSEMBLY
DI UNA MACCHINA VIRTUALE**

Relatore: prof. Michele Moro

Laureando: Marco Beggio

Corso di laurea in Ingegneria Informatica

Anno Accademico
2006 - 2007

Indice

Sommario	3
1 Introduzione	7
2 Macchina virtuale	11
2.1 Struttura del linguaggio PLC	11
2.2 Definizione della macchina virtuale	19
2.2.1 Architettura	19
2.2.2 Istruzioni	20
2.2.3 Gestione dei costrutti	25
2.2.4 Mappatura della memoria	26
2.2.5 Strutture dati	30
2.2.6 Ingresso/Uscita	31
2.2.7 Debug	31
2.2.8 File di configurazione	31
3 Manuale utente	33
3.1 Avvio del traduttore	33
3.2 File tradotto	35
4 Manuale tecnico	37
4.1 Struttura del traduttore	37
4.1.1 Presentazione del parser	37
4.1.2 Struttura del modulo di traduzione	39

4.2	Scelte implementative	43
4.2.1	Scrittura del file di output e tecnica di backpatch	43
4.2.2	Codifica delle istruzioni	49
4.2.3	Gestione Label	54
4.2.4	Calcolo skip logico	58
4.2.5	Gestione Function	64
4.2.6	Gestione Job	68
4.3	Interventi di modifica o espansione	73
5	Conclusioni	75
	Ringraziamenti	77
	Bibliografia	79

Sommario

La tesi è dedicata allo sviluppo di un traduttore per un software PLC (*Programmable Logic Controller*). Questo traduttore è un modulo di un compilatore che permette di ottenere *assembly* per una macchina virtuale avendo come origine sorgente PLC.

Lo scopo di questo software è quello di interpretare gli output di un parser LALR per generare codice conforme alle specifiche della macchina virtuale. Il traduttore dovrà, nella produzione del codice, focalizzare l'attenzione sulla massimizzazione dell'efficienza in fase di esecuzione cercando di ridurre il carico di lavoro della macchina virtuale.

Il linguaggio di programmazione scelto è Java e il tool di sviluppo utilizzato è *BlueJ*. Questa scelta ha permesso di ottenere un software che potrà essere eseguito su qualsiasi piattaforma avente installata la macchina virtuale Java. Attualmente è stato testato in ambiente OS Linux e OS Microsoft Windows.

Capitolo 1

Introduzione

Lo scopo di questa tesi è la realizzazione di un software di traduzione da sorgente PLC a codice *assembly* per una macchina virtuale. Questo strumento è un modulo di un progetto più ampio che comprende la realizzazione di un parser *LALR* che effettui l'analisi lessicale, sintattica e semantica di un sorgente PLC e una macchina virtuale in grado di interpretare il codice tradotto ed offra la funzionalità di eseguire il debugging in tempo reale.

La progettazione di questo software è iniziata dall'analisi delle specifiche di funzionamento del linguaggio PLC dal manuale tecnico *PLC Compile v4.0* che ha portato allo sviluppo della definizione e delle specifiche della macchina virtuale. Nella fase di definizione della macchina virtuale è stato preso spunto dal libro "*Calcolatori elettronici*" [1]. Successivamente l'analisi si è focalizzata sulla struttura del parser precedentemente sviluppato in quanto la struttura del traduttore doveva essere conforme ad esso per la sua integrazione.

Il traduttore ha come requisito la capacità di generare codice che sia efficiente in fase di esecuzione nella macchina virtuale spostando il carico di lavoro, per quanto possibile, in fase di traduzione. A questo scopo sono stati risolti gli alias dei nomi, gestiti i salti ai label come offset relativi ed introdotti gli skip nelle espressioni logiche.

Il linguaggio di programmazione utilizzato è Java, il quale, oltre ad essere lo stesso linguaggio utilizzato per realizzare il parser, offre la possibilità di una programmazione modulare con tutti i benefici da essa derivati e permette

di ottenere un programma multiplatforma che può essere eseguito su diversi sistemi senza dover mettere mano al codice.

Il software di sviluppo utilizzato è *The Interactive Java Environment BlueJ* (<http://www.bluej.org>). Questo tool offre la possibilità di effettuare una veloce analisi del codice evidenziando gli errori sintattico - lessicali e genera uno schema UML delle classi che aiuta nella progettazione object oriented dell'architettura del software.

Un altro tool molto utilizzato per la visione dei file di uscita binari e il debug del software è l'editor di testi che viene distribuito gratuitamente *Notepad++* (<http://notepad-plus.sourceforge.net/it/site.htm>) il quale dispone di un *HexEditor* con molte funzionalità che sono state utili per agevolare il lavoro di verifica.

La tesi è suddivisa in tre parti principali:

- *definizione della macchina virtuale* che include le specifiche di funzionamento e la descrizione del linguaggio;
- *manuale utente* che guida all'utilizzo del software;
- *manuale tecnico* che illustra le scelte implementative e come intervenire per un successivo sviluppo.

Si richiede la conoscenza del linguaggio di programmazione Java per poter interpretare e comprendere l'implementazione del programma. Un altro requisito è la conoscenza della struttura di un semplice calcolatore e del linguaggio *assembly* con cui si programma, quali sono le componenti e quale funzione adempiono, che sta alla base della progettazione della macchina virtuale.

Il PLC, o *Programmable Logic Controller*, è un computer industriale specializzato nella gestione di processi industriali. Esso esegue un programma ed elabora i segnali digitali ed analogici provenienti da sensori e diretti agli attuatori presenti in un impianto industriale. La programmazione viene effettuata

normalmente con un PC sul quale un software specializzato permette di creare programmi da scaricare nella memoria della CPU del PLC. Il programma viene scritto su PC, quindi scaricato sul PLC, e salvato sul PC stesso, per ulteriori modifiche.

Il risultato che si vuole ottenere è quello di poter abbandonare l'hardware dedicato e poter eseguire tutti i software, senza la necessità di modifiche, in ambiente embedded con sistema operativo Windows CE. Questo strumento offre inoltre la possibilità di effettuare il debug più velocemente e con minore difficoltà.

