CUVINT INAINTE

Lucrarea de faţă constituie prima parte a cursului "Limbaje şi compilatoare" ţinut la direcţia de aprofundare "Software" de la facultatea Automatică. Cu toate că volumul de ore acoperit de lucrare reprezintă mai puţin de o treime din totalul orelor de curs, importanţa capitolelor tratate în lucrare nu poate fi subestimată. Este vorba de o introducere în teoria limbajelor de programare, cu accent pus pe consecinţele diferitelor concepte şi noţiuni utilizate în teorie asupra implementării limbajelor de programare.

In tratarea subiectului am fost confruntat cu diversitatea limbajelor de programare, cu lipsa unei sistematizări a domeniului, cu dezvoltarea lui anarhică. Am încercat să surprind cele mai importante concepte ale domeniului şi să le înlănţui într-o viziune unificatoare. Fără să fi dorit a fi în primul rînd original, a trebuit:

- să introduc noţiuni noi,
- să accentuez asupra unor aspecte mai puţin frecventate de autori, dar edificatoare în perspectiva adoptată,
- să simplific acolo unde mi se părea a fi inutil de complicat,
- să elimin ceea ce nu este încă suficient decantat în domeniu.

Toate acestea în scopul unităţii de viziune, a desfăşurării logice a elementelor tratate. Din aceleaşi motive am precedat capitolul 3, principalul capitol al lucrării atît ca dimensiuni cît şi ca importanţă, de două capitole introductive care oarecum îl pregătesc, fără a fi într-o strînsă legătură cu el.

Primul capitol, "Sisteme informaţionale" introduce o anume perspectivă faţă de lumea înconjurătoare, în particular faţă de însuşi domeniul programării, perspectivă pe care apoi o adoptă tot cursul. Astfel, am plecat de la considerentul că un programator trebuie să-şi formeze prin instruire şi/sau experienţă un anumit gen de reguli în alcătuirea propriei replici informaţionale la lumea înconjurătoare. Regulile urmează să-l ajute în transcrierea "impresiilor" sale într-un limbaj de programare.

Mecanismul regulilor are la bază abstractizări şi detalieri dirijate şi duce inevitabil spre modele ce urmează a fi simulate pe calculator. Situaţia este frecventă în ingineria compilatoarelor.

Al doilea capitol intitulat "Comunicare şi limbaje" se referă la transmiterea informaţiilor reprezentate simbolic, ipostază tipică dialogului om-calculator în cadrul căruia programul constituie

cel mai complex mesaj. Rolul programului în comunicare,necesitatea unui limbaj precis definit şi a "înţelegerii" identice a acestui limbaj de către programator şi calculator ne vor conduce în mod natural la definiţia limbajului prin cele trei componente ale sale şi apoi la noţiunile de traducere,translator,compilator.

Avem deci o dublă perspectivă asupra oricărui limbaj de programare: mijloc de comunicare simbolică între programator şi calculator,dar în acelaşi timp mijlocitor între"lumea reprezentărilor programatorului" şi "lumea reprezentărilor calculatorului",deci între sistemele informaţionale şi sistemele de calcul. Ambele perspective sînt utile proiectantului de compilatoare şi discutate în prima parte a capitolului 3,"Limbaje de programare".

In acest capitol se face şi o trecere în revistă a celor mai importante facilităţi puse la dispoziţie de limbajele de programare procedurale (cele mai utilizate la ora actuală). Facilităţile,grupate în date,comenzi,structuri de nivel superior,sînt abordate atît conceptual cît şi din punct de vedere al implementării.

Lucrarea cuprinde şi un al patrulea capitol,"Limbaje de asamblare şi asambloare". Problematica capitolului este în mare măsură diferită de cea a capitolelor anterioare. Rostul lui în această secvenţă de capitole rezultă din considerarea limbajelor de asamblare în categoria limbajelor de programare. Dificultatea subiectului constă în dependenţa puternică a limbajelor de asamblare faţă de maşină, ceea ce împiedică o prezentare general valabilă a acestor limbaje. Am ales totuşi facilităţile mai des întîlnite şi anume doar cele transmise prin directive. Exemplificările au fost făcute pe baza limbajului MACRO 11. Restul capitolului se ocupă de implementarea limbajelor de asamblare,deci de structura asambloarelor. Problema nu mai ţine deloc de tema lucrării,dar rezultă din prezentarea limbajelor de asamblare şi poate fi rezolvată simplu,pe loc. Am preferat înserarea ei aici deoarece amînarea pînă în ultima parte a cursului nu ar fi justificată de învăţarea între timp a unor tehnici deosebite de cele cunoscute de studenţi,tehnici comune compilatoarelor; în afara tabelelor de simboluri,celelalte probleme de implementare sînt relativ distincte de problemele compilatoarelor şi cu rezolvări mai simple. In fine,ca o extindere a facilităţilor se prezintă macroasamblarea indicîndu-se şi soluţia de implementare a macroasambloarelor.

Am preferat în prezentarea noţiunilor stilul direct,limbajul natural,recurgînd la cel formalizat în cazuri speciale,acolo unde modelul formal,chiar simplificator mi s-a părut sugestiv. Consider că,

cel puţin în această parte a cursului,formalizarea apare ca un balast din lipsa unor instrumente matematice adecvate tratării clare a problemelor.

Partea a doua a cursului: "Bazele teoretice ale compilatoarelor" beneficiază de o tratare precisă,matematică. Aici sînt introduse principalele modele matematice folosite în teoria compilării. Referitor la această parte studenţii sînt îndemnaţi să consulte cele două manuale deja apărute pe plan local: "Teoria automatelor" (1975, 1979) şi "Teoria limbajelor formale" (1976,1979). Ele conţin mai multe informaţii decît se pot transmite în cadrul cursului de "Limbaje şi compilatoare" referitor la cele două domenii de mare interes teoretic.

Ultima parte a cursului,"Ingineria compilatoarelor",este cea mai amplă şi,prin forţa lucrurilor,cea mai inginerească avînd ca obiectiv învăţarea modului de proiectare şi implementare a unui compilator. Deocamdată nu este acoperită de un manual elaborat pe plan local. Acesta este în curs de redactare urmînd să apară în anul viitor.

Tot în categoria planurilor de viitor se înscriu şi două îndrumare,unul de laborator şi altul pentru proiecte. Ele vor suplini şi lipsa problemelor din lucrarea de faţă.

Sub forma în care apare,lucrarea trebuie considerată ca un material în lucru. Ea a fost actualizată de cîteva ori şi este departe de a fi în formă finală.

"Autorul îşi rezervă dreptul de a o mai îmbunătăţi ulterior".

Bucureşti,2o noiembrie 1981. Autorul.