UNIUNEA EUROPEANĂ
GUVERNUL ROMÂNIEI
Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Programare în limbaj de asamblare

4. Etapele realizarii unui program în limbaj de asamblare (LA).

Programarea procesoarelor

Fiecare procesor (sau calculator bazat pe un procesor) are propriul său limbaj, denumit limbaj maşină, care este exprimat în alfabetul binar {0,1}. Acest limbaj se defineşte în etapa de proiectare a unității de comandă a procesorului (calculatorului). Unitatea de comandă este cea care citeşte instrucțiunile din memorie (fetch), le decodifică şi le execută. Aceste instrucțiuni reprezintă operații elementare: transferul datelor între memorie şi unitatea de prelucrare, operații aritmetice şi logice, modificarea fluxului de comandă etc. Un program în cod (limbaj) maşină mai este denumit şi cod obiect.

Limbajele de nivel înalt, precum C sau Pascal, au fost concepute pentru a nu ține cont de caracteristicile tehnice specifice oricărui calculator, în timp ce limbajul de asamblare este proiectat pentru o familie specifică de procesoare.

Limbajul de asamblare reprezintă o codificare simbolică a codului maşină. Aceste coduri simbolice sunt denumite mnemonici. Limbajul, care este foarte aproape de structura maşinii, are următoarele avantaje:

- oferă posibilitatea utilizării tuturor resurselor şi facilităților calculatorului;
- permite implementarea mai eficientă (cu mai puține resurse şi mai rapid) a anumitor funcții, dar şi de a realiza taskuri de mare tehnicitate, care ar fi dificile, dacă nu imposibile, într-un limbaj de nivel înalt;

dar are şi unele dezavantaje:

- programele nu sunt portabile; întrucât fiecare procesor are propriul său limbaj de asamblare, programele nu pot fi executate pe alt tip de procesor, din altă familie;
- solicită un efort mult mai mare din partea programatorului pentru elaborarea unor aplicații complexe.

De exemplu, un fişier sursă care conține un program sursă, în limbaj de asamblare, este mult mai mare decât un fişier ce conține un program echivalent într-un limbaj de nivel înalt (Pascal sau C). În schimb, programul obiect obținut pentru varianta în limbaj de asamblare este mult mai mic şi, în consecință, mai rapid. Programarea în limbaj de asamblare presupune cunoaşterea structurii şi arhitecturii calculatorului, cunoaşterea sistemului de operare şi a asamblorului.

Programele rezidente şi rutinele de tratare a întreruperilor sunt aproape întotdeauna dezvoltate în limbaj de asamblare. De asemenea, deşi cei mai mulți specialişti dezvoltă noile aplicații în limbaje de nivel înalt, care sunt mai uşor de scris şi de întreținut, o practică obişnuită este de a scrie în limbaj de asamblare acele rutine care sunt critice din punct de vedere al timpului de execuție (adică cele care produc „gâtuiri“ de prelucrare).

Asambloarele sunt programe care translatează programele sursă, scrise în limbaj de asamblare, în cod maşină. Asemănător limbajelor de nivel înalt, şi asambloarele au cunoscut o evoluție în timp. Astfel, primele asambloare utilizau pentru referirea datelor adrese fizice, după care s-au introdus nume simbolice pentru date şi adrese. Ulterior au apărut macroasambloare, care permit utilizarea de macroinstrucțiuni, care permit ca unor secvențe de instrucțiuni să li se asocieze un nume; aceste macroinstrucțiuni pot fi parametrizate şi pot fi utilizate pentru a defini un limbaj de tip limbaj de asamblare, dar mult mai puternic. De asemenea, se pot utiliza nu numai date simple, dar şi date structurate, asemănătoare celor din limbajele de nivel înalt.

Operația de asamblare se realizează în două etape:

1. La primul pas se generează o tabelă de simboluri, ce va conține toate numele simbolice din program, exceptând numele simbolice externe (definite în alte module), instrucțiuni şi directive de asamblare. În această etapă, asamblorul contorizează instrucțiunile şi datele şi asociază numelor simbolice o poziție relativă (deplasament) față de începutul programului, ca şi cum programul ar începe de la adresa zero. În realitate, programul nu se încarcă în memoria RAM de la adresa zero, care este o zonă folosită de sistemul de operare, ci de la o adresă furnizată de sistemul de operare, în funcție de spațiul de memorie disponibil; din acest motiv, programul furnizat de asamblor este relocabil.
2. La pasul al doilea se obține programul obiect, translatând programul instrucțiune cu instrucțiune şi înlocuind în instrucțiunile respective numele simbolice cu valorile numerice asociate în tabela de simboluri. Programul executabil se obține în urma etapei de editare de legături (linkeditare), care permite legarea mai multor module relocabile într-un fişier executabil, rezolvându-se toate referințele încrucişate care au fost folosite în alte module.

În concluzie, etapele care trebuie parcurse pentru realizarea şi execuția unui program în limbaj de asamblare sunt descrise schematic în figura următoare.

Prima etapă este cea de editare a programului sursă, ce are extensia *.ASM*, şi care poate fi realizată cu orice editor (WS- opțiunea NonDocument, NC, editoarele din TPascal sau TC etc.).

După aceasta urmează asamblarea, care furnizează programul obiect, cu extensia *.OBJ*; dacă apar erori în faza de asamblare, se reia editarea programului, corectând erorile respective.

La editarea de legături, pe lângă programul curent, pot fi utilizate şi alte programe obiect, eventual din bibliotecă.

Fişierul executabil astfel obținut, poate fi testat şi depanat utilizând una din utilitarele TurboDebugger (TD) sau Debug(ger).

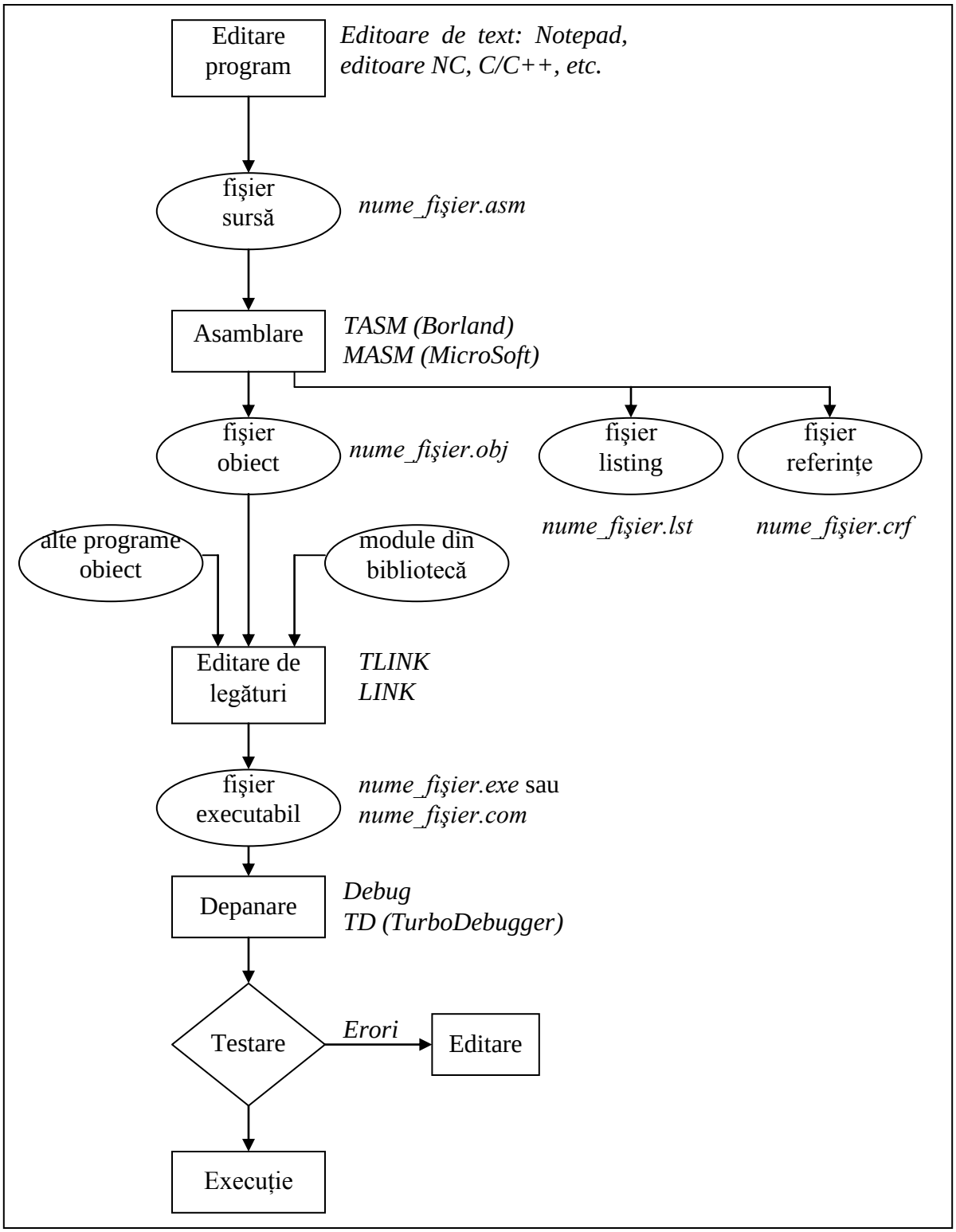
Pentru a nu relua de fiecare dată secvența de comenzi, se poate edita un fişier de comenzi (batch), ASM.BAT, pentru asamblarea, editarea de legături, şi eventual ştergerea fişierului neutilizat (.OBJ), de forma:

```
@echo off
tasm              % 1.asm
tlink       % 1.obj
del               % 1.obj
```

Comanda va fi utilizată astfel:

```
>ASM nume_fisier.asm
```

Dacă se doreşte şi obținerea fişierului listing, atunci comanda de asamblare trebuie utilizată cu opțiunea /l.



Etapele de realizare a unui program în limbaj de asamblare